Alocação de Blocos de Recursos em Redes Sem Fio Considerando Fatiamento de Rede e Comunicação D2D Utilizando Aprendizado por Reforço Profundo

Hudson H. de Souza Lopes[1], **Flávio H. Teles Vieira** [1]

[1] Centro de Excelência em Redes Inteligentes Sem Fio e Serviços Avançados (CERISE)
Escola de Engenharia Elétrica, Mecânica e de Computação (EMC)
Universidade Federal de Goiás (UFG), Goiânia-GO-Brasil.

hudson_lopes@ufg.br, flavio_vieira@ufg.br

Abstract. *In this article, we consider a mobile network scenario composed of a single Base Station (BS), where Network Slice Instances (NSI) and Device-to-Device (D2D) Communication occur. In order to solve the complex problem of allocation Resource Blocks (RBs) considering Service Level Agreements (SLAs) in each NSI, we propose an approach named DDPG-KRP, which is based on the Deep Deterministic Policy Gradient (DDPG) algorithm, the K-Nearest Neighbors (KNN) algorithm, and the Reward Penalization (RP) method. The simulation results show that the proposed algorithm significantly outperforms other Deep Reinforcement Learning (DRL) based methods.*

Resumo. *Neste artigo, consideramos um cenário de redes sem fio composto por uma única Estação Base (BS - Base Station) onde ocorrem Instâncias de Fatia de Rede (NSI - Network Slice Instances) e Comunicação Dispositivo a Dispositivo (D2D - Device-to-Device). A fim de resolver o problema complexo de alocação de Blocos de Recursos (RBs - Resource Blocks) levando em consideração os Acordos de Nível de Serviço (SLAs - Service Level Agreements) em cada NSI, propomos uma abordagem denominada DDPG-KRP, baseada no algoritmo de Gradiente de Política Determinística Profunda (DDPG - Deep Deterministic Policy Gradient), com o algoritmo K-vizinhos mais próximos (KNN - K-Nearest Neighbors) e no método de Penalização de Recompensa (RP - Reward Penalization). Os resultados de simulações mostram que o algoritmo proposto supera significativamente outros métodos baseados em Aprendizagem por Reforço Profundo (DRL - Deep Reinforcement Learning).*

1. Introdução

Redes de comunicação sem fio cada vez mais densas tendem a ser projetadas como redes multisserviço, a fim de atender às demandas de aplicações exigentes, como cidades inteligentes, Internet das Coisas (IoT) e Indústria 4.0. Nesse contexto, o Fatiamento de Rede (NS - Network Slicing) permite a criação de múltiplas redes virtuais independentes sobre uma mesma infraestrutura física, cada uma adaptada a requisitos específicos de serviço As fatias de rede ou Instâncias de Fatia de Rede (NSI - *Network Slice Instances*) podem diferir em termos de recursos suportados e otimizações de funções de rede; nesse caso, tais NSI podem atender, por exemplo, diferentes tipos de aplicações [Liu et al. 2020, 3GPP 2026]. Em paralelo, a comunicação Dispositivo a Dispositivo

(D2D - Device-to-Device), introduzida nas especificações do 3GPP, possibilita conexões diretas entre dispositivos próximos, reduzindo a carga na estação base e aumentando a eficiência espectral [3GPP 2015, Nadeem et al. 2021].

A integração entre NS e D2D tem se destacado por proporcionar ganhos em vazão, eficiência espectral e flexibilidade no suporte a aplicações heterogêneas [Moubayed et al. 2015, Sun et al. 2020, Nadeem et al. 2021]. No entanto, a alocação eficiente de recursos de rádio nesse cenário configura um problema altamente complexo, principalmente devido ao elevado espaço de ações. Nesse sentido, técnicas de Aprendizagem por Reforço Profundo (DRL - *Deep Reinforcement Learning*) têm sido amplamente exploradas por sua capacidade de aprender políticas de decisão diretamente da interação com o ambiente, sem depender de modelagem matemática explícita, sendo particularmente adequadas para redes sem fio de próxima geração [Liu et al. 2020, Souza Lopes et al. 2023, Engin et al. 2025].

Neste trabalho, propõe-se o algoritmo DDPG-KRP, que combina o método *Deep Deterministic Policy Gradient* (DDPG) com *clustering* K-vizinhos mais próximos (KNN - K-*Nearest Neighbor*) e Penalização de Recompensa (RP - *Reward Penalization*), com o objetivo de melhorar a aprendizagem em espaços de ação de alta dimensionalidade, mitigando o problema conhecido como *curse of dimensionality* e garantindo o atendimento aos requisitos de Acordo de Nível de Serviço (SLA - *Service-Level Agreement*) de cada aplicação. A abordagem modela o problema de alocação de recursos como um processo de decisão sequencial, no qual o agente aprende diretamente a partir do tráfego das NSIs, sem necessidade de formulação matemática fechada.

Adicionalmente, é proposta uma função de recompensa que equilibra o atendimento aos SLAs com a maximização da vazão e da Taxa de Entrega de Pacotes (TEP), promovendo também o uso eficiente da potência por meio de uma estratégia de penalização que atribui recompensa nula a ações que violam restrições. Para lidar com a alta dimensionalidade do espaço de ação, é incorporada uma estratégia de eliminação de ações inviáveis durante o treinamento, removendo combinações que excedem os recursos disponíveis ou não satisfazem os SLAs, o que melhora a eficiência da exploração e acelera a convergência do aprendizado. Por fim, o desempenho do algoritmo proposto é validado por meio de extensas simulações, demonstrando ganhos significativos em relação a métodos DRL convencionais, especialmente em cenários com grande espaço de ação.

O restante deste artigo está estruturado da seguinte forma: Seção 2 apresenta trabalhos relacionados a este estudo. Seção 3 fornece o modelo do sistema. Seção 4 formula o problema de alocação de recursos de rádio através de transição de estados. Seção 5 introduz o algoritmo DDPG-KRP proposto para resolver o problema de alocação de recursos, considerando o cenário de rede móvel com NS e D2D . Seção 6 discute os resultados obtidos dos agentes DRL. Seção 7 resume as conclusões obtidas.

2. Trabalhos Relacionados

O gerenciamento de recursos para melhorar o desempenho de redes sem fio modernas é um problema complexo para ser resolvido utilizando métodos tradicionais. Portanto, considerando os desafios relacionados, em [Moubayed et al. 2015], os autores formularam o problema de virtualização de recursos em uma rede sem fio com comunicação D2D *underlay*. Essa combinação gerou um problema de programação inteira não linear

dividido em dois problemas inteiros lineares menores. As duas soluções possíveis foram combinadas e o problema foi resolvido usando dois escalonadores diferentes. Em [Saravanan and Ganeshkumar 2020], é utilizado um modelo DRL para monitorar, analisar e prever o comportamento do caminho de encaminhamento relativamente à capacidade de transmissão e à disponibilidade dos veículos numa rede *ad hoc* veicular. Em [Nadeem et al. 2021], os autores apresentaram um problema de otimização de eficiência espectral em uma rede celular 5G heterogênea baseada em D2D com NS para maximizar a eficiência espectral média e a vazão da rede sem degradar o desempenho do sistema. Em [Moubayed et al. 2015, Xu 2017, Nadeem et al. 2021], os autores formularam o problema de alocação de recursos através de um problema de otimização complexo. A alocação de recursos foi baseada em métodos matemáticos e utilizou algoritmos tradicionais para resolver o problema de otimização. Neste trabalho, o DRL foi usada como uma técnica para alocação de recursos num cenário de rede sem fio estocástica. Em [Sun et al. 2020], os autores propuseram um esquema de alocação de recursos entre fatias em uma rede de comunicação virtualizada baseada em D2D. Um agente DRL foi usado para o ajuste de recursos entre as fatias. Os autores formularam o problema de alocação de recursos como um problema de otimização convexa e resolveram-no com um método de direção alternada. O objetivo era equilibrar a utilização de recursos e a QoS para várias fatias.

Em [Suh et al. 2022], os autores propuseram uma técnica de NS baseada em DRL. O algoritmo DQN foi usado mais especificamente para encontrar a política de alocação de recursos que maximizar a vazão e satisfaz os requisitos de QoS em sistemas *Beyond* 5G (B5G). Os autores utilizaram uma técnica para eliminar ações indesejáveis que não podem satisfazer os requisitos de QoS. Assim, a exploração do agente DRL foi direcionada para ações desejáveis, melhorando as hipóteses de tomar a decisão ótima na alocação de recursos e aumentando a velocidade de convergência no treino DRL. Este trabalho apresenta um avanço ao propor alocação de recursos de rádio em um cenário complexo de rede sem fio composto por fatiamento de rede e comunicação D2D, uma vez que poucos trabalhos na literatura consideram essas duas tecnologias de rede sem fio de próxima geração. Neste trabalho, propomos uma melhoria ao algoritmo DDPG usando KNN e penalização por recompensa para tornar a sua aprendizagem mais eficiente, de modo que a alocação de recursos seja mais adaptável a este novo cenário de comunicação móvel e tentar cumprir os requisitos de SLA de cada fatia de rede. Conforme ilustrado na Tabela 1, nossa pesquisa oferece uma análise mais aprofundada e abrangente das técnicas de DRL destinadas a enfrentar os desafios da gestão de recursos de rádio, ultrapassando o escopo dos trabalhos existentes.

Tabela 1. Revisões das pesquisas sobre gerenciamento de recursos de rádio.

Referência	Alocação de potência	NS	D2D	Método
[Moubayed et al. 2015]	x	✓	✓	Tradicional
[Xu 2017]	x	✓	x	Tradicional
[Sun et al. 2020]	x	✓	✓	DRL
[Saravanan and Ganeshkumar 2020]	x	x	✓	DRL
[Nadeem et al. 2021]	✓	✓	✓	Analítico
Esta pesquisa	✓	✓	✓	DRL

3. Modelo do Sistema

Consideramos o cenário onde uma única BS fornece o recurso de rádio para três tipos de NSI que executam aplicações diferentes: Comunicações Ultra-Confiáveis e de Baixa Latência (URLLC - *Ultra-Reliable and Low-Latency Communications*), Vídeo e Voz sobre IP (VoIP - *Voice-over IP*). Em cada NSI ocorre comunicações dos UEs com os pares D2D como mostra a Fig. 1. O modelo do sitema de comunicação móvel consiste de uma área de cobertura de uma pequena célula (*Small Cell*) com a BS localizada no centro. De forma semelhante ao trabalho em [Li et al. 2018], a demanda de tráfego dos UEs foi gerada utilizando distribuições probabilísticas tanto para chegada quanto para o tamanho do pacote. Para as comunicações dos pares D2D foi considerado a categoria *inband* e a subcategoria *underlay* para reutilizar o espectro licenciado, tendo em conta o efeito de interferência na alocação de recursos a qualquer UE ligado ao par D2D. No entanto, assumimos que os UEs não reutilizam o espectro de outros UEs.

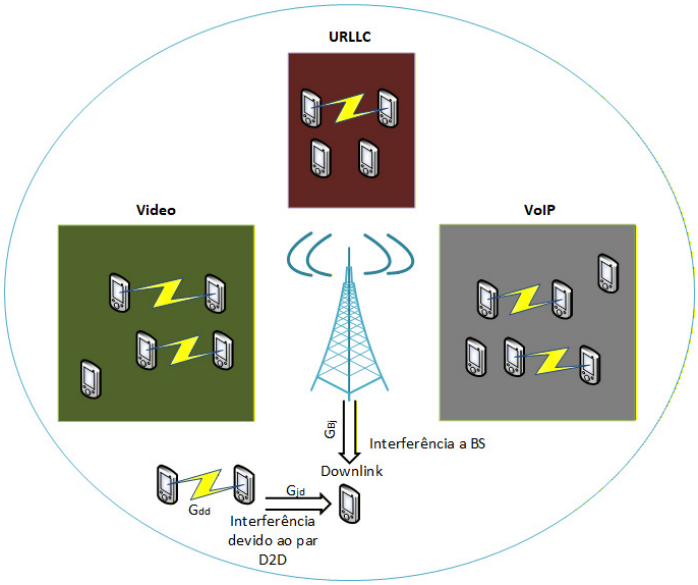


Figura 1. Modelo de sistema de comunicação móvel com uma BS e três NSI com comunicação D2D.

O recurso de largura de banda de rádio pode ser visto como bidimensional, ou seja, inclui os domínios do tempo e frequência. Em nossa implementação, consideramos uma largura de banda de 10 MHz, ou seja, temos 50 Blocos de Recursos (RBs - *Resource Blocks*) por *slot*. Essa configuração foi adotada com o objetivo de reduzir o esforço computacional das simulações; no entanto, a metodologia proposta é diretamente extensível às configurações de largura de banda e numerologias atualmente empregadas em sistemas 5G comerciais. Sendo assim, neste trabalho a alocação de recursos é realizada por *slot* com uma duração de um Intervalo de Tempo de Transmissão (TTI - *Transmission Time Interval*) de 1 ms. A nossa implementação do sistema 5G leva em consideração a infraestrutura 4G LTE em modo não autônomo (NSA - *non-standalone*). Nesse sistema, Os RBs são sempre agendados em pares, compondo um Bloco de Escalonamento (SB - *Scheduling Block*), com um TTI de 1 ms.

3.1. Modelagem do Canal de Comunicação

Neste trabalho, os RBs alocados ao par D2D são compartilhados com os UEs que estão ligados ao par, ou seja, o espectro entre os UEs e os pares D2D conectados é reutilizado, assim, a vazão total da rede é aumentada. Quando a transmissão de pacotes ocorre na direção *downlink*, os UEs ficam expostos as interferências quando qualquer par D2D transmite utilizando os mesmos RBs alocados. Neste trabalho, consideramos que não possui interferência entre células distintas pois consideramos o cenário de uma única célula. A interferência da comunicação D2D com os UEs na rede celular deve ser restringida para manter um nível mínimo de desempenho da rede. Se a distância entre os dispositivos D2D for longa, o D2D necessitará de mais energia. Neste caso, a rede celular perde desempenho devido à interferência [Zulhasnine et al. 2010]. O nível de interferência não dependerá apenas da potência de transmissão do par D2D, mas também do ganho de canal entre o par D2D e os UEs. A relação sinal-interferência e ruído (SINR - *Signal-to-Interference and Noise Ratio*) do UE j é proposta da seguinte forma:

$$\Gamma_j = \frac{p_j \times G_{Bj}}{\sigma^2 + \sum_{d=1}^{D} p_d \times G_{jd}}, \tag{1}$$

onde G_{jd} representa o ganho de canal entre o UE j e o par D2D d, G_{Bj} representa o ganho de canal entre a BS e o UE j, p_j e p_d são as potências dos sinais de transmissão do UE j e do par D2D d, respectivamente. σ_z^2 é a potência de ruído. A SINR do par D2D d é proposta como:

$$\Gamma_d = \frac{p_d \times G_{dd}}{\sigma^2 + \sum_{j=1}^{Z} p_j \times G_{Bj}}, \tag{2}$$

onde G_{dd} representa o ganho de canal entre os dispositivos do par D2D d. De acordo com [Zulhasnine et al. 2010], o ganho entre dois dispositivos é dado por $G_{dd} = \Phi_{dd} \times d_{dd}^{-\delta_{pl}}$, onde d_{dd} é a distância entre o transmissor d e o receptor d; δ_{pl} é o expoente de perda de percurso constante; e Φ_{dd} é uma constante de normalização que depende das propriedades de propagação do ambiente. Neste trabalho, assumimos que δ_{pl} = 2 e $\Phi = 0,001$.

O ganho linear entre a BS e o UE G_{Bj} e entre o UE e o par D2D G_{jd} é dada por $10^{-PL/10}$, onde PL é a perda de caminho recebida pela *Small Cell*. Consideramos o modelo de Okumura-Hata para predição da perda de percurso (*path loss*) [Nadeem et al. 2021] dado por $PL = 140,70 + 36,70 \times log(dist[km])dBm$, onde $dist$ é a distância entre o UE e a BS. A capacidade Shannon da rede em cada NSI i é calculada da seguinte forma $C_i^s = \sum_{j=1}^{Z} l_j \times (log_2(1 + \Gamma_j) + log_2(1 + \Gamma_d))$, onde Z é o número total de UEs e l_j são os RBs alocados aos UE.

4. Formulação do Problema

Neste artigo, propomos modelar o problema de alocação de RBs para cada NSI através de transição de estados, onde o agente de DRL decide qual ação tomar, explorando o ambiente e maximizando a recompensa de longo prazo [Chakraborty and Sivalingam 2023]. O ambiente de RL é representado por três componentes: estado, ação e recompensa, que são descritos abaixo:

O estado do sistema $s(t)$ é representado pelo número de pacotes que chegam em cada NSI i. O novo estado é representada pela equação de fila primeiro a entrar primeiro a sair (FIFO - *First-In-First-Out*). A ação é dada pelos RBs w alocados a cada NSI i.

A função de recompensa $r(s(t), a(t))$ proposta na Eq (3) é dada pela soma ponderada da capacidade do canal de Shannon C e do índice de satisfação da taxa de entrega de pacotes TEP para cada NSI. Propomos também aplicar o método RP que consiste em penalizar a função de recompensa, ou seja, durante a fase de treino, se o agente DRL tomar uma ação que não satisfaça as restrições de SLA ou execeda a quantidade de recurso disponível a recompensa será um valor de 0.

$$r(s(t), a(t)) = \begin{cases} 0, \text{se a ação escolhida é indesejada} \\ \sum_{i=1}^{F} (\iota \times C_i^s + \varsigma \times TEP_i), \end{cases} \tag{3}$$

onde F é à quantidade total de NSI, ι e ς são as constantes utilizadas para adicionar peso aos valores e garantir que eles tenham o mesmo número de dígitos. A TEP é obtida dividindo o número de pacotes transmitidos com sucesso pelo número total de pacotes que estão no *buffer*. Os SLAs impõem requisitos rigorosos de vazão e latência para a TEP [Li et al. 2018].

4.1. Redução do Espaço de Ação e Eliminação de Ações Indesejadas

O tamanho do espaço de ação na alocação de recursos em rede móveis considerando o paradigma NS é muito elevado, pois é proporcional ao número de NSI e as possíveis combinações de ações. O número das possíveis decisões aumentam exponencialmente com o número de RBs e fatias. Devido a este enorme espaço de ações, a BS irá provavelmente explorar ações indesejáveis durante a fase de treino, tais como, a alocação de recursos que não podem satisfazer as restrições de vazão, os requisitos de atraso ou que excedem a quantidade total de recursos disponíveis, o que irá diminuir a velocidade de convergência do treino e impedir a DRL de maximizar a recompensa. Neste trabalho, consideramos a alocação de recursos num tempo de simulação de 1 segundo, ou seja, 2000 *slots* a serem alocados entre as NSI, de modo que o número de combinações possíveis para as três fatias serão de $2000 \times 2000 \times 2000 = 8 \times 10^9$. Como a saída da rede neural DQN e DDQN é composta pelo número de combinações possíveis de ações, o treino da rede neural em simulações torna-se impraticável, sem eliminar as ações indesejáveis para reduzir o espaço de ação.

Assim como em [Suh et al. 2022], para os algoritmos DQN e DDQN, utilizamos a eliminação de ações para resolver o problema da alta dimensão do espaço de ações. A eliminação de ações é a técnica que consiste em excluir combinações de ações indesejáveis do espaço de ações, a fim de aumentar a velocidade e a qualidade da política de treino. Após um número razoável de episódios de treino, a eliminação de ações indesejáveis permite que a rede RL treinada (DQN e DDQN) na BS gere boas estratégias (ou seja, decisões de alocação de recursos). A eliminação de ações satisfazem as restrições de SLA para cada aplicação sem exceder a quantidade total de recursos disponíveis. No nosso caso, vazão de 10 Mbps e latência de 5 ms para a aplicação URLLC, vazão de 5 Mbps e latência de 5 ms para a aplicação de vídeo e vazão de 51 kbps e latência de 5 ms para a aplicação de VoIP [Li et al. 2018].

5. DDPG -KRP

O algoritmo de Gradiente de Política Determinística Profunda (DDPG - *Deep Deterministic Policy Gradient*) é composto pela integração do DQN e do método ator-crítico para resolver problemas com espaço de ações contínuo e de alta dimensão. O DDPG é um dos agentes que realiza a aprendizagem de políticas utilizando o gradiente da função objetivo (4) que representa a recompensa esperada a longo prazo. O agente DDPG aprende uma política que é contínua e determinística. Para garantir a exploração do ambiente, as trajetórias geradas pelo agente provêm de uma outra política, que é estocástica e faz com que o DDPG seja um método fora da política (*off-policy*). O espaço de observações pode ser discreto ou contínuo, mas o espaço das ações tem de ser contínuo. O algoritmo DDPG é livre de modelo (*model-free*), ou seja, não necessita do modelo do ambiente e, para efetuar a aprendizagem, aprende a política determinística $\mu(s|\theta^{\mu})$, parametrizada pelos pesos da rede neural θ^{μ}, e a função de valor do par ação-estado para esta política $Q(s, a|\theta^{Q})$, parametrizada pelos pesos da rede neural θ^{Q}. Além disso, utiliza a arquitetura ator-crítico [Lillicrap et al. 2016].

$$\nabla_{\theta^{\mu}} J \approx E[\nabla_{\theta^{\mu}} Q(s(t), a(t)|\theta^{Q})|_{s=s(t),a=\mu(s(t)|\theta^{\mu})}]. \tag{4}$$

Os autores em [Silver et al. 2014] mostram que o gradiente desta função objetivo em relação aos parâmetros θ^{μ} é dada por:

$$\nabla_{\theta^{\mu}} J \approx E[\nabla_{a} Q(s(t), a(t)|\theta^{Q})|_{s=s(t),a=\mu(s(t))}.\nabla_{\theta^{\mu}} \mu(s(t)|\theta^{\mu})|_{s=s(t)}]. \tag{5}$$

Note-se que a equação acima está no formato da arquitetura ator-crítico, em que o ator está relacionado com $\nabla_{\theta^{\mu}} \mu(s(t)|\theta^{\mu})$ e o crítico está relacionado com $\nabla_{a} Q(s(t), a(t)|\theta^{Q})$. O método ator-crítico é constituído por uma rede neural chamada de ator e outra de crítico. A rede de ator toma o estado como sua entrada e produz uma ação determinística exata. A rede do crítico é uma rede de valor Q que toma tanto o estado como a ação como entradas, e produz o valor Q como uma única saída. Assim como no DQN o DDPG também utiliza redes de destino que são cópias atrasadas no tempo das suas redes originais que seguem lentamente as redes aprendidas. A utilização destas redes de valores-destino melhora consideravelmente a estabilidade na aprendizagem, outra característica do DQN utilizada no DDPG é o *buffer* de repetição (RR) para a amostra de experiência para atualizar os parâmetros da rede neural [Alwarafy et al. 2021]. No algoritmo DDPG, os parâmetros das cópias são atualizados periodicamente em cada passo τ de uma forma suavizada em relação aos originais através de um filtro de primeira ordem com um fator de suavização (ρ_s) de 10^{-5}. Estas cópias são designadas por redes de destino. A utilização destas redes de destino melhoram consideravelmente a estabilidade da aprendizagem, como mostra a Fig. 2.

O **Algoritmo 1** mantém uma função de ator parametrizada, $\mu(s|\theta^{\mu})$, que especifica a política atual, mapeando deterministicamente os estados para uma ação específica. A rede crítico $Q(s, a|\theta^{Q})$ aprende minimizando o erro utilizando a Equação (6). O ator é atualizado seguindo a regra da cadeia aplicada ao retorno esperado desde o início da

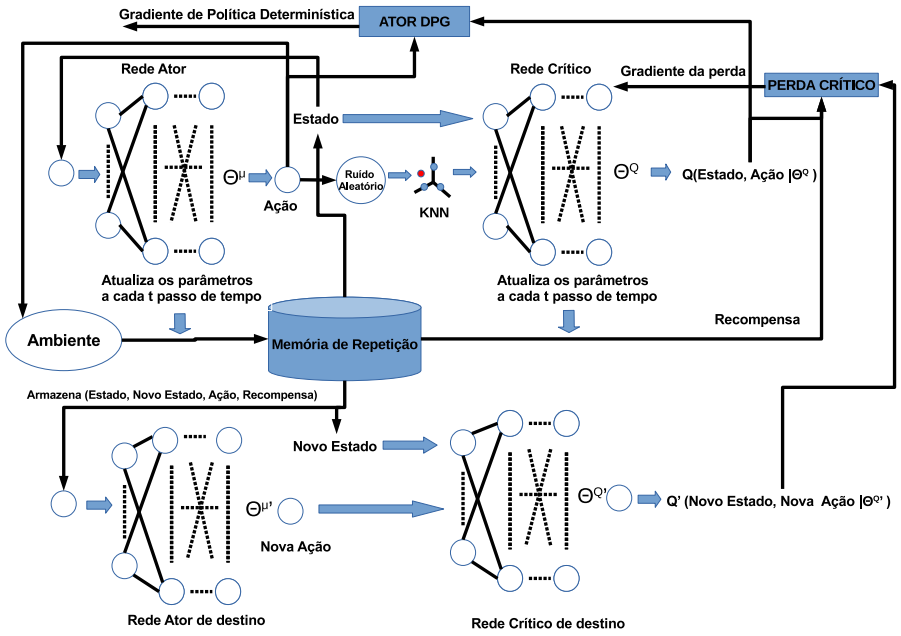


Figura 2. Arquitetura DDPG-KRP.

distribuição J dada pela Equação (5) [Lillicrap et al. 2016]. Uma vez que o DDPG gera uma política determinística, para garantir a exploração do ambiente, acrescentamos um ruído (κ) na saída do ator. Isto é necessário porque nem todos os ambientes são naturalmente estocásticos. Neste caso, a saída da rede de atores é dada por $a(t) = \mu(s(t)|\theta^{\mu}) + \kappa_t$.

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(y(i) - Q(s(i), a(i)|\theta^{Q}))^2, \tag{6}$$

onde $y(i)$ para $i = 1, ..., N$ são os valores de saída desejados para as amostras de entrada de treino dadas pela função de perda de Bellman, ou seja, $y(i) = r(i) + \gamma Q^{'}(s(i+1), \mu^{'}(s(i+1)|\theta^{\mu^{'}})|\theta^{Q'})$ em que $\mu^{'}(s(i+1)|\theta^{\mu^{'}})$ e $Q^{'}(s(i+1), \mu^{'}(s(i+1)|\theta^{\mu^{'}})|\theta^{Q'})$ são as redes de destino do ator e do crítico, respectivamente.

5.1. K-Vizinhos mais Próximos

O DDPG foi originalmente concebido para lidar com espaço de ação com valor contínuo. No entanto, a alocação de recurso é um problema de decisão com valor discreto em que as combinações da alocação de recursos não podem exceder o total de recurso disponível. Inspirados na abordagem em [An et al. 2023], utilizamos o algoritmo k vizinhos mais próximos (KNN - *K-Nearest Neighbors*) para discretizar o DDPG e adaptá-lo ao espaços de ações discretos. A ideia básica é utilizar um algoritmo baseado no espaço contínuo para gerar primeiro uma ação inicial contínua. Depois, as K ações discretas mais próximas são encontradas utilizando o algoritmo KNN [Muja and Lowe 2014]. Escolhemos a ação vizinha mais próxima.

Em termos de sobrecarga computacional, o algoritmo DDPG tem um melhor desempenho de treino porque a ação gerada na saída da sua rede neural de atores está

Algoritmo 1: DDPG-KRP.

Inicialize aleatoriamente a rede crítico Q(s,a|θ^Q) e a rede ator μ(s|θ^μ) com pesos θ^Q e θ^μ.
Inicialize a rede de destino $Q^{'}$ e $\mu^{'}$ com pesos $\theta^{Q^{'}}$, $\theta^{\mu^{'}}$.
Inicialize o *buffer* de repetição RR.
para $episode$ *de 1 até M* **faça**
    Recebe o estado de observação inicial s(1);
    para t *de 1 até T* **faça**
        Selecionar a ação de acordo com a política atual e o ruído de exploração;
        a(t) = knnsearch(), ou seja, escolhe a ação vizinha mais próxima [Muja and Lowe 2014];
        if *ação indesejada* **then**
            r(t) = 0;
        end
        Executar a ação a(t) e observar a recompensa r(t) e observar o novo estado s(t+1);
        Armazena a transição (s(t), a(t), r(t), s(t+1)) em RR;
        Escolha uma amostra de um mini lote aleatório de N transições (s(i), a(i), r(i), s(i+1)) de RR;
        Seja y(i) = r(i) + γ $Q^{'}$ (s(i+1), $\mu^{'}$(s(i+1) |$\theta^{\mu^{'}}$)|$\theta^{Q'}$);
        Atualiza a rede crítico minimizando a perda;
        Loss = $\frac{1}{N}\sum_{i=1}^{N}$ (y(i) - Q(s(i), a(i)|θ^Q))^2;
        Atualiza a política de ator utilizando uma amostra do gradiente de política:
        $\nabla_{\theta^\mu} J \approx \sum_{i=1}^{N} \nabla_a \mathrm{Q(s(i),a(i)}|\theta^Q)|_{s=s(i),a=\mu(s(i))} . \nabla_{\theta^\mu}\mu(s(i)|\theta^\mu)|_{s=s(i)}$;
        if *mod(t,*τ*) == 0* **then**
            Atualiza os pesos das redes de destino:
            $\theta^{Q^{'}} = \rho_s \times \theta^Q + (1-\rho_s) \times \theta^{Q^{'}}$;
            $\theta^{\mu^{'}} = \rho_s \times \theta^\mu + (1-\rho_s) \times \theta^{\mu^{'}}$;
        end
    fim
fim

relacionada com apenas um neurônio, enquanto os outros métodos têm um neurônio na saída da sua rede neural para cada combinação possível de ações, o que no nosso cenário equivale a aproximadamente 500,000 combinações. Portanto, a abordagem proposta, que usa o algoritmo KNN, melhora o desempenho do algoritmo DDPG-KRP, bem como a eficiência do DDPG considerando redes de grande porte. Vários trabalhos recentes tentaram resolver o problema de algumas abordagens de DRL que apresentam um grande espaço de ação discreto, discretizando o espaço de ação contínuo [Dulac-Arnold et al. 2015, An et al. 2023]. Neste sentido, como mostra o Algoritmo 1 proposto, uma aproximação KNN é usada devido à sua busca ágil em tempo logarítmico e mapeia um espaço de ação contínuo para um espaço de ação discreto

[Muja and Lowe 2014, An et al. 2023].

6. Resultados e Discussões

Nesta seção, avaliamos o desempenho do algoritmo DDPG-KRP. As simulações foram realizadas com as seguintes configurações de *software* e *hardware*: *Software* MATLAB versão R2023b com a biblioteca (*Reinforcement Learning Toolbox*), processador Intel Core i5-1035G1 de 1,00 GHz e 16 GB de RAM, sem placa de vídeo dedicada. A Tabela 2 apresenta o conjunto completo dos parâmetros para as simulações.

Tabela 2. Parâmetros das simulações

Parâmetros	Valores
Potência de transmissão da BS (P_{bs})	10 W ou 40 dBm
Potência de transmissão do par D2D	0.2 W ou 23 dBm
Quantidade de UEs por NSI	URLLC = 4, Video = 46, VoIP = 46
Quantidade de pares D2D por NSI	Distribuição de Poisson [média = 5]
Potência de ruído	-114 dBm
Raio da Small Cell	300 m
Distância máxima entre o par D2D	30 m
Intervalo de tempo de transmissão (TTI)	1 ms
Modelo de mobilidade	Os UEs e os pares D2D são uniformemente distribuídos na Small Cell a cada TTI
Tempo de simulação	1 s
Peso sobre a capacidade do canal (ι)	0,1
Peso sobre a TEP(ς)	5000

Foi implementado o algoritmo DDPG-KRP para resolver o problema de alocação de recursos em cada NSI. A rede neural do ator foi implementada com quatro camadas ocultas totalmente ligadas, com 256 neurônios cada, e a função de ativação *leaky* ReLU, a entrada da rede neural do ator são três neurônios representando a quantidade de pacotes que chegam nas três NSI, a saída também são três neurônios representando os RBs alocados a cada fatia. Além disso, a rede neural crítico foi utilizada considerando três camadas ocultas totalmente ligadas com 128 neurônios cada e a função de ativação *leaky* ReLU. Foi efetuada uma afinação empírica dos hiperparâmetros. As taxas de aprendizagem para as redes de atores e de críticos foram de 0,01. O tamanho do lote foi de 256. O tamanho do *buffer* de repetição (RR) foi de 10000. O fator de desconto para a recompensa acumulada foi de 0,90.

O desempenho do algoritmo proposto DDPG-KRP foi comparado com outros dois métodos DRL baseados em redes neurais profundas (i.e., DQN e DDQN), considerando a mesma função de recompensa proposta na equação (3) e a discretização das ações utilizando porcentagens e eliminando as ações indesejáveis na saída da rede neural. A rede neural dos dois métodos foi implementada com quatro camadas ocultas totalmente ligadas com 64 neurônios em cada uma e a função de ativação *leaky* ReLU. Os hiperparâmetros foram definidos com uma taxa de aprendizagem de 0,01, uma probabilidade de escolha de ação aleatória de 0,1, um fator de desconto de 0,9, um tamanho de lote de 256 e um tamanho de *buffer* de repetição de 10000. O desempenho de convergência dos algoritmos de DRL foi analisado para mais de 3000 episódios. Para uma quantidade maior de

episódios, não observamos alterações significativas nos resultados obtidos com os agentes considerados.

A Fig. 3 apresenta uma comparação detalhada do desempenho das técnicas candidatas, na qual os resultados para os agentes DRL são obtidos após a aprendizagem. As Figuras (a)-(c) ilustram a porcentagem de RBs total alocados a cada NSI utilizando o gráficos de pizza e destacam a TEP pelo texto circundante. A Fig. 3 mostra que a alocação de recursos utilizando o algoritmo DQN não consome todo o recurso disponível, gerando uma TEP baixa. O algoritmo DDQN consegue alocar quase todo o recurso disponível, mas não satisfaz a TEP de todas as NSI. Em comparação com os algoritmos DQN e DDQN, o algoritmo DDPG-KRP proposto permite que o agente atue gerando uma ação determinística no espaço contínuo. A Fig. 3(c), confirma a nossa convição de que o DDPG-KRP apresenta uma melhor aprendizagem na complexa relação entre a demanda e a TEP do que os outros agentes, uma vez que todos os recursos disponíveis são utilizados e é obtido um melhor desempenho em termos de TEP em comparação com as outras abordagens consideradas.

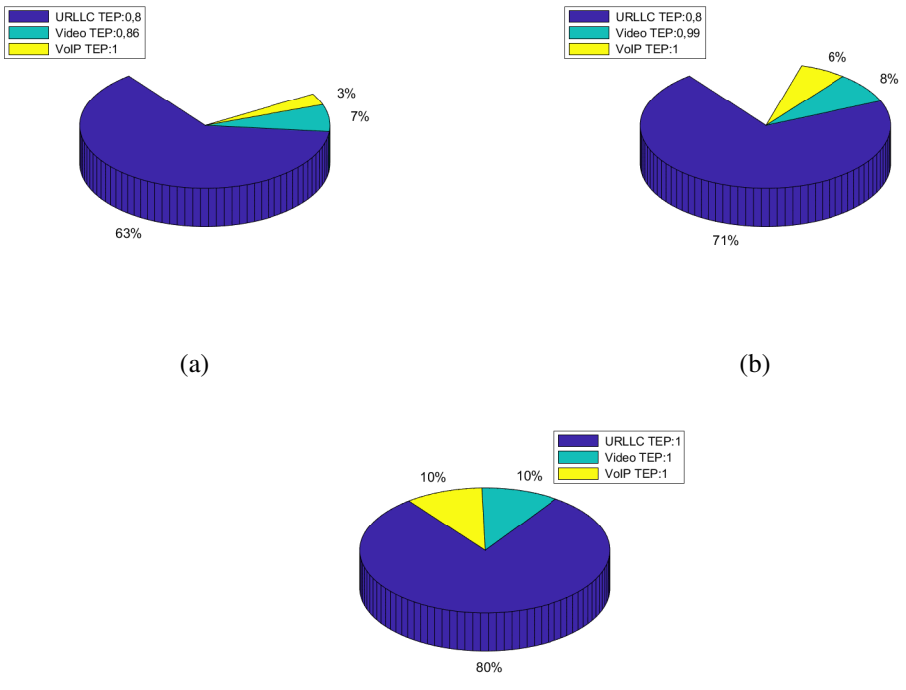
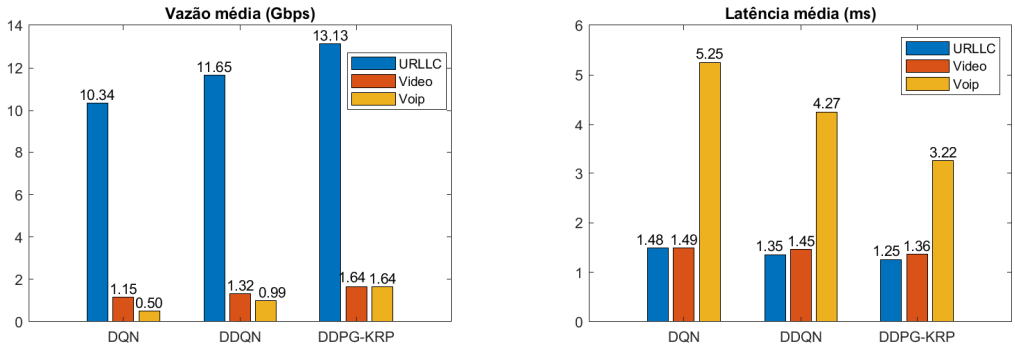


Figura 3. Comparação de desempenho entre os diferentes agentes DRL. (a) DQN. (b) DDQN. (c) DDPG-KRP.

A Fig. 4 demonstra uma comparação da vazão e latência em cada NSI por cada agente DRL. As exigências de tráfego da aplicação URLLC é alta devido ao tamanho do pacote. Dado o maior volume de transmissão e o requisito de latência estritamente inferior, o agente DRL aprende a alocar uma maior quantidade de recurso para essa aplicação, consequentemente é a aplicação que apresenta uma maior vazão e menor latência. Em se-

guida, a aplicação de video que domina a transmissão devido ao seu grande número de pacotes. O agente DDPG-KRP apresenta os maiores valores de vazão média e menores valores de latência média para cada NSI devido à sua capacidade de aprendizagem em alocar todo o recurso disponível.



(a) (b)

Figura 4. Comparação de vazão e latência entre os diferentes agentes DRL após alocação de RBs em cada NSI. (a) Vazão. (b) Latência.

A Fig. 5 mostra a latência média a medida que aumentam-se a quantidade de pares D2D e maximiza a vazão na rede. Comparamos o valor da latência média entre os três agentes DRL considerando somente a aplicação URLLC com requisitos mais rigorsos. Uma vez que a formulação matemática da latência média não é trivial, assume-se a existência de uma fila FIFO mantida na BS. A latência média é então estimada ao longo de 1000 intervalos de tempo de transmissão (TTIs), considerando que o tráfego não atendido no slot corrente é enfileirado para transmissão nos slots subsequentes. Conforme ilustrado na Fig. 5, o agente DDPG-KRP apresenta desempenho superior a partir de 10 pares D2D na rede quando comparado aos algoritmos DQN e DDQN.

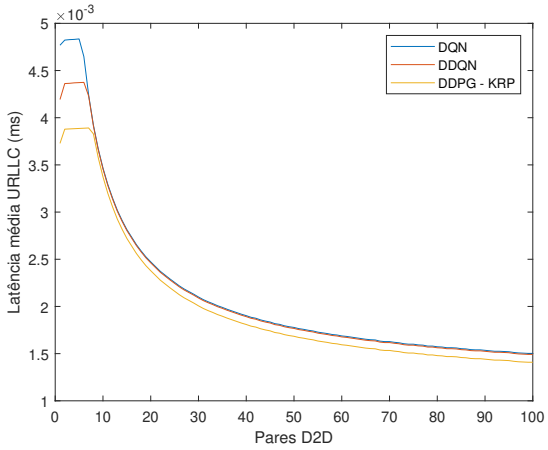


Figura 5. Comparação da latência média da aplicação URLLC entre os agentes DRL com o aumento de pares D2D na rede

7. Conclusão

Este trabalho propõe um algoritmo baseado no agente DDPG, incorporando o algoritmo KNN e penalização de recompensa para ações indesejáveis, para abordar o problema

de alocação de recursos envolvendo NS e pares D2D, levando em conta os SLAs de cada fatia. Considerando os efeitos de interferência na alocação de recursos para UEs e pares D2D a rede móvel reutilizou o espectro maximizando a vazão e minimizando a latência na transmissão de dados. A saída da rede neural dos métodos (DQN e o DDQN) baseia-se na quantidade discreta de ações possíveis, considerando o elevado número de combinações possíveis para a alocação de recursos em cada NSI, foi empregada uma técnica de eliminação de ações indesejadas com o objetivo de reduzir o espaço de ações. Como consequência, aproximadamente 85% das ações inviáveis foram removidas, resultando em uma redução significativa da complexidade do processo de decisão. As extensas simulações mostraram que o DDPG-KRP pode alcançar melhores resultados, uma vez que as abordagens comparativa baseadas em DRL que utilizam o espaço de ação discreto, dificulta a aprendizagem do agente devido a grande quantidade de neurônios na saída da rede neural profunda.

Em trabalhos futuros pretende-se a implantação do cenário de rede em ambiente real em conformidade com o padrão (O-RAN – *Open Radio Access Network*) com suporte a pilha do projeto (srsRAN - *software radio systems Radio Access Network*) com o projeto de código aberto Open5GS, e a plataforma de rádio de alto desempenho *Ettus Research* (USRP - *Universal Software Radio Peripheral*) X310 para avaliar uma rede 5G considerando outras NSI com requisitos distintos.

8. Agradecimentos

Os autores gostariam de agradecer ao Centro de Excelência em Redes Inteligentes Sem Fio e Serviços Avançados (CERISE), à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), e à Fundação de Amparo à Pesquisa no Estado de Goiás (FAPEG) pelo apoio e financiamento no desenvolvimento da pesquisa.

Referências

3GPP (2015). Proximity-based services (ProSe). Technical Specification (TS) 23.303, 3rd Generation Partnership Project (3GPP). TS 23.303 v13.0.0.

3GPP (2026). System architecture for the 5G System (5GS). Technical Specification (TS) 23.501, 3rd Generation Partnership Project (3GPP). TS 23.501 v19.7.0.

Alwarafy, A., Abdallah, M. M., Ciftler, B. S., Al-Fuqaha, A. I., and Hamdi, M. (2021). Deep reinforcement learning for radio resource allocation and management in next generation heterogeneous wireless networks: A survey. *CoRR*, abs/2106.00574.

An, Q., Segarra, S., Dick, C., Sabharwal, A., and Doost-Mohammady, R. (2023). A deep reinforcement learning-based resource scheduler for massive MIMO networks. *CoRR*, abs/2303.00958.

Chakraborty, S. and Sivalingam, K. (2023). Drl-based admission control and resource allocation for 5g network slicing. *Sādhanā*, 48.

Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., and Coppin, B. (2015). Deep reinforcement learning in large discrete action spaces.

Engin, E., Hokelek, I., Gorcin, A., and Cirpan, H. A. (2025). A pre-emptive scheduling mechanism for service assurance of network slicing in next generation cellular networks. *IEEE Access*, 13:23297–23311.

Li, R., Zhao, Z., Sun, Q., I, C.-L., Yang, C., Chen, X., Zhao, M., and Zhang, H. (2018). Deep reinforcement learning for resource management in network slicing. *IEEE Access*, 6:74429–74441.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Liu, Y., Ding, J., and Liu, X. (2020). A constrained reinforcement learning based approach for network slicing. *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pages 1–6.

Moubayed, A., Shami, A., and Lutfiyya, H. (2015). Wireless resource virtualization with device-to-device communication underlaying lte network. *IEEE Transactions on Broadcasting*, 61(4):734–740.

Muja, M. and Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240.

Nadeem, L., Amin, Y., Loo, J., Azam, M. A., and Chai, K. K. (2021). Efficient resource allocation using distributed edge computing in d2d based 5g-hcn with network slicing. *IEEE Access*, 9:134148–134162.

Saravanan, M. and Ganeshkumar, P. (2020). Routing using reinforcement learning in vehicular ad hoc networks. *Computational Intelligence*, 36(2):682–697.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, page I–387–I–395.

Souza Lopes, H., Rocha, F., and Vieira, F. (2023). Deep reinforcement learning based resource allocation approach for wireless networks considering network slicing paradigm. *Journal of Communication and Information Systems*, 38(1):21–33.

Suh, K., Kim, S., Ahn, Y., Kim, S., Ju, H., and Shim, B. (2022). Deep reinforcement learning-based network slicing for beyond 5g. *IEEE Access*, 10:7384–7395.

Sun, G., Boateng, G. O., Ayepah-Mensah, D., Liu, G., and Wei, J. (2020). Autonomous resource slicing for virtualized vehicular networks with d2d communications based on deep reinforcement learning. *IEEE Systems Journal*, 14(4):4694–4705.

Xu, Y. (2017). Energy-efficient power control scheme for device-to-device communications. *Wireless Personal Communications*, 94(3):481–495.

Zulhasnine, M., Huang, C., and Srinivasan, A. (2010). Efficient resource allocation for device-to-device communication underlaying lte network. *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 368–375.