DEEP REINFORCEMENT LEARNING E HIPER-HEURÍSTICA APLICADOS À ALOCAÇÃO DE RECURSOS EM SISTEMAS DE COMUNICAÇÕES 6G COM COMUNICAÇÕES D2D E SENSOREAMENTO

GABRIEL PIMENTA DE FREITAS CARDOSO

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DEEP REINFORCEMENT LEARNING AND HYPER-HEURISTIC APPLIED TO RESOURCE ALLOCATION IN 6G COMMUNICATIONS SYSTEMS WITH D2D COMMUNICATIONS AND SENSING

DEEP REINFORCEMENT LEARNING E HIPER-HEURÍSTICA APLICADOS À ALOCAÇÃO DE RECURSOS EM SISTEMAS DE COMUNICAÇÕES 6G COM COMUNICAÇÕES D2D E SENSOREAMENTO

GABRIEL PIMENTA DE FREITAS CARDOSO

ORIENTADOR: PROF. PAULO ROBERTO DE LIRA GONDIM

DISSERTAÇÃO DE MESTRADO EM
ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGEE 814/24

BRASÍLIA/DF: MAIO - 2024

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DEEP REINFORCEMENT LEARNING E HIPER-HEURÍSTICA APLICADOS À ALOCAÇÃO DE RECURSOS EM SISTEMAS DE COMUNICAÇÕES 6G COM COMUNICAÇÕES D2D E SENSOREAMENTO

GABRIEL PIMENTA DE FREITAS CARDOSO

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

_____

Prof. Paulo Roberto de Lira Gondim – ENE/FT - Universidade de Brasília
Orientador

_____

Prof. José Marcos Câmara Brito – INATEL
Membro Externo

_____

Prof. Hugerles Sales Silva – ENE/FT - Universidade de Brasília
Membro Interno

_____

Prof. Leonardo Rodrigues Araújo Xavier de Menezes – ENE/FT - Universidade de Brasília
Suplente

BRASÍLIA, 28 DE MAIO DE 2024.

FICHA CATALOGRÁFICA

PIMENTA, GABRIEL
Deep Reinforcement Learning e Hiper-Heurística aplicados à alocação de recursos em sistemas de comunicações 6G com comunicações D2D e sensoreamento [Distrito Federal] 2024.
xvi, 104p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2024).
Dissertação de mestrado – Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Deep Reinforcement Learning	2. PPO
3. Hiper-heurística	4. Comunicações e Sensoreamento
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

PIMENTA DE FREITAS CARDOSO, G. (2024). Deep Reinforcement Learning e Hiper-Heurística aplicados à alocação de recursos em sistemas de comunicações 6G com comunicações D2D e sensoreamento . Dissertação de mestrado em Engenharia Elétrica, Publicação PPGEE 814/24, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 104p.

CESSÃO DE DIREITOS

Gabriel Pimenta de Freitas Cardoso

Departamento de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

Dedico esse trabalho à minha avó Maria José Cardoso por me ensinar o valor do trabalho intelectual, à minha avó Francisca Maria de Jesus por me ensinar o valor do trabalho prático e ao meu tio Djalma Cardoso, por me mostrar o valor da união de ambos.

ACKNOWLEDGMENTS

Agradeço, primeiramente, a Deus por me proporcionar essa oportunidade tão valiosa para o meu aprendizado em vida.
Em seguida, agradeço à minha mãe Jane Margarete Pimenta e ao meu pai Luiz Fernando de Freitas Cardoso por todo apoio e dedicação dispendidos antes e durante a minha formação acadêmica, além de toda força e amor recebidos diariamente.
Sou profundamente grato aos meus irmãos Matheus Pimenta de Freitas Cardoso e Ludmila Pimenta de Freitas Cardoso por sempre me servirem de suporte e inspiração desde os meus primeiros dias de vida.
Agradeço à minha namorada Stefani da Mota Ribeiro por sempre me apoiar nos momentos de alegria e de dificuldade, renovando minhas forças para continuar a cada dia e estando presente para me ajudar e para celebrar as nossas conquistas.
Agradeço, também, ao Professor Dr. Paulo Henrique Portela de Carvalho, por todo empenho dedicado a mim, me ensinando e acompanhando de perto desde a graduação. Além disso, de forma mais geral, agradeço pela dedicação que o Professor Paulo Portela tem com a pesquisa acadêmica e com a educação e desenvolvimento de seus alunos.
Por fim, agradeço ao Professor Dr. Paulo Roberto de Lira Gondim, orientador deste trabalho, por toda dedicação e cuidado com o trabalho desenvolvido, além de agradecer por sua disponibilidade para me acompanhar durante esta importante etapa da minha vida.

RESUMO

Título: Deep Reinforcement Learning e Hiper-Heurística aplicados à alocação de recursos em sistemas de comunicações 6G com comunicações D2D e sensoreamento
Autor: Gabriel Pimenta de Freitas Cardoso
Orientador: Prof. Paulo Roberto de Lira Gondim
Programa de Pós-Graduação em Engenharia Elétrica
Brasília, 28 de maio de 2024

Este trabalho propõe uma estratégia para a realização conjunta da alocação de espectro e do controle de potências em sistemas de comunicações móveis de 5G e gerações futuras com sensoreamento integrado. A aplicação tratada neste trabalho se situa em um contexto relacionado à Indústria 4.0, abarcando um cenário industrial com comunicações primárias, comunicações D2D e sensores. A solução proposta para realizar a alocação de recursos no uplink desse sistema é composta pela conjunção de duas técnicas no estado da arte: algoritmos de *Deep Reinforcement Learning* (DRL) e Hiper-Heurísticas (HH). O primeiro algoritmo que forma a estratégia conjunta proposta neste trabalho foi desenvolvido utilizando-se redes neurais treinadas por meio de técnicas de DRL para controle de potências. O segundo algoritmo, que completa a estratégia proposta, foi desenvolvido através de técnicas relacionadas à aplicação de HHs em conjunção com algoritmos de DRL, para realização da alocação do espectro disponível. A estratégia conjunta teve como objetivos principais: proteger as comunicações primárias, almejando-se reduzir a taxa de *outage* para garantia de uma comunicação de qualidade, além de proteger os sensores do sistema, objetivando-se reduzir a taxa de *outage* dos sensores para garantir que a probabilidade de detecção estivesse acima de um limiar pré definido. Como objetivo secundário, o algoritmo proposto buscou maximizar a taxa de transmissão das comunicações D2D.

Os resultados mostraram que o algoritmo de controle de potências que obteve o melhor desempenho, em comparação com outros algoritmos da área no estado da arte, foi o *Proximal Policy Optimization* (PPO). Esse algoritmo proposto, separadamente ao de alocação do espectro, foi capaz, em um *Resource Block* (RB), de reduzir a taxa de *outage* das comunicações primárias de 64.35% para 11.75%, reduzir a taxa de *outage* dos sensores de 38.5% para 4.4% e aumentar a SNIR das comunicações D2Ds de -25.6 dB para -7.5 dB, se comparado com os resultados obtidos por um algoritmo aleatório. Para a estratégia completa, isto é, com algoritmos de DRL e HH realizando tanto o controle de potências quanto a alocação do espectro, os resultados indicaram que, em comparação com uma alocação de recursos baseada em escolhas aleatórias, a estratégia conjunta foi capaz de reduzir a taxa de *outage* das comunicações primárias de 65.8% para 13.3%, reduzir a taxa de *outage* dos sensores de 48.1% para 3.3% e aumentar a SNIR das comunicações D2Ds de -24.3 dB para -11.2 dB em sistemas com múltiplos RBs. Além disso, o algoritmo se mostrou escalável para sistemas com diferentes quantidades de comunicações, sensores e RBs, sendo aplicável em diferentes configurações do sistema.

Palavras-chave: Deep Reinforcement Learning, PPO, Hiper-heurística, Comunicações e Sensoreamento.

ABSTRACT

Title: Deep Reinforcement Learning and Hyper-Heuristic applied to resource allocation in 6G communications systems with D2D communications and sensing
Author: Gabriel Pimenta de Freitas Cardoso
Supervisor: Prof. Paulo Roberto de Lira Gondim
Graduate Program in
Brasília, May 28, 2024

This work proposes a strategy for the joint execution of spectrum allocation and power control in 5G mobile communication systems and future generations with integrated sensing. The application addressed in this work is situated in a context related to Industry 4.0, encompassing an industrial scenario with primary communications, D2D communications, and sensors. The proposed solution for resource allocation in this system consists of the conjunction of two state-of-the-art techniques: Deep Reinforcement Learning (DRL) algorithms and Hyper-Heuristics (HH). The first algorithm that forms the proposed joint strategy in this work was developed using neural networks trained through DRL techniques for power control. The second algorithm, which completes the proposed strategy, was developed using techniques related to the application of HHs in conjunction with DRL algorithms for the allocation of available spectrum. The main objectives of the joint strategy were: to protect primary communications, aiming to reduce the outage rate to ensure quality communication; and to protect the system's sensors, aiming to reduce the sensor outage rate to ensure that the detection probability was above a predefined threshold. As a secondary objective, the proposed algorithm sought to maximize the transmission rate of D2D communications.

The results showed that the power control algorithm that performed best, compared to other state-of-the-art algorithms in the area, was Proximal Policy Optimization (PPO). This proposed algorithm, separately from the spectrum allocation algorithm, was able, in a Resource Block (RB), to reduce the primary communications outage rate from 64.35% to 11.75%, reduce the sensor outage rate from 38.5% to 4.4%, and increase the SNIR of D2D communications from -25.6 dB to -7.5 dB, compared with the results obtained by a random algorithm. For the complete strategy, that is, with DRL and HH algorithms performing both power control and spectrum allocation, the results showed that, compared to a resource allocation based on random choices, the joint strategy was able to reduce the primary communications outage rate from 65.8% to 13.3%, reduce the sensor outage rate from 48.1% to 3.3%, and increase the SNIR of D2D communications from -24.3 dB to -11.2 dB in systems with multiple RBs. Additionally, the algorithm proved scalable for systems with varying amounts of communications, sensors, and RBs, being applicable in different system configurations.

Keywords: Deep Reinforcement Learning, PPO, Hyper-Heuristic, Communications and Sensing.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

Conjuntos

A	Conjunto de comunicações primárias
C	Conjunto de veículos alvo do sensoreamento
J	Conjunto de comunicações móveis
K	Conjunto de *resource blocks*
L	Conjunto de comunicações D2D
M	Conjunto de equipamentos do usuário
N	Conjunto de equipamentos de sensoreamento
Q	Conjunto de sensores

Variáveis

$\eta_{[\cdot],k}$	Ruído recebido pela comunicação/sensor $[\cdot]$ no k-ésimo RB
$\phi_{\mathbf{min}}$	Limiar mínimo da probabilidade de detecção dos sensores
ψ	Eficiência espectral
$\psi_{\mathbf{min}}$	Eficiencia espectral mínima para as comunicações primárias
$\zeta_{[\cdot],k}$	SNIR do canal da comunicação/sensor $[\cdot]$ no k-ésimo RB
$b_{[\cdot],k}$	Variável indicadora se a comunicação/sensor $[\cdot]$ será alocada no k-ésimo RB
$h^{d}_{[\cdot],[\star],k}$	Ganho de propagação do canal direto entre o transmissor da comunicação/sensor $[\cdot]$ e o receptor da comunicação/sensor $[\star]$ no k-ésimo RB
$h^{e}_{[\cdot],[\star],k}$	Ganho de propagação do canal com reflexão no alvo entre o transmissor da comunicação/sensor $[\cdot]$ e o receptor da comunicação/sensor $[\star]$ no k-ésimo RB
$p^{t}_{[\cdot],k}$	Potência de transmissão da comunicação/sensor $[\cdot]$ no k-ésimo RB
$x_{[\cdot],k}$	Símbolo transmitido pela comunicação/sensor $[\cdot]$ no k-ésimo RB
$y_{[\cdot],k}$	Sinal recebido pela comunicação/sensor $[\cdot]$ no k-ésimo RB

LISTA DE ACRÔNIMOS E ABREVIA CÕES

3GPP *3rd Generation Partnership Project*. 1, 16, 94

5G 5ª geração dos sistemas de comunicação móvel. 1, 2, 16, 94

6G 6ª geração dos sistemas de comunicação móvel. 1, 2, 94

D2D *device-to-device*. xii, xiii, 1–3, 13, 15, 18, 20, 21, 23, 41, 43–48, 51–53, 56–64, 72, 73, 79, 80, 84, 85, 88, 90–96

DRL *Deep Reinforcement Learning*. 1, 5, 6, 9, 12, 25, 31, 41, 43, 51, 52, 57, 62, 63, 70, 73, 74, 83, 85, 92, 94, 95

ERB Estação Rádio Base. 1, 3, 13, 14, 18, 94

GLRT *Generalized Likelihood Ratio Test*. 1, 17, 94

HH Hiper-Heurística. 1, 6, 12, 66–74, 78–81, 94

LLH *Low Level Heuristic*. xiii, 1, 67–74, 78, 80, 81, 83, 94

LOS *Line-of-Sight*. 1, 16, 94

LTE-A *Long Term Evolution - Advanced*. 1, 94

mmWave *Millimeter wave*. 1, 15, 16, 94

OFDM *Orthogonal Frequency Division Multiplexing*. 1, 7, 8, 14, 94

RB *Resource Block*. xii, xiii, xv, 1, 7, 14–17, 20, 21, 24, 41, 43, 44, 61–64, 71–73, 78–81, 89–91, 93–96

RCS *Radar cross section*. 1, 17, 94

SNIR *Signal-to-Noise-plus-Interference Ratio*. xii, xiii, 1, 8, 16, 18, 45, 51–53, 56–58, 61, 63, 84, 85, 88, 91–96

UE *User Equipment*. 1, 3, 13–15, 47, 72, 94

V2V *vehicle-to-vehicle*. 1, 2, 94

1 INTRODUÇÃO

1.1 5G E 6G NA INDÚSTRIA 4.0

A Indústria 4.0, também referida como a quarta revolução industrial, é caracterizada pela tendência atual de automação e intercâmbio de dados na tecnologia de manufatura. Este conceito incorpora uma série de inovações tecnológicas nos domínios da automação, controle e tecnologia da informação, todas aplicadas aos processos de manufatura. Através da integração de sistemas ciber-físicos, da Internet das Coisas (IoT) e da Internet dos Serviços, a Indústria 4.0 promove a implementação de um modelo de produção altamente digitalizado e interconectado [1, 2].

Dentro do contexto da Indústria 4.0, as tecnologias de 5ª geração dos sistemas de comunicação móvel (5G) e, posteriormente, as de 6ª geração (6G) têm e terão um papel crucial. O 5G vem permitindo maior capacidade de tráfego de dados e latência mais baixa em comparação com as gerações anteriores. Essas características tornam o 5G uma tecnologia importante para suportar a grande quantidade de dispositivos conectados na Indústria 4.0, permitindo a comunicação em tempo real entre máquinas e sistemas [2].

A sexta geração dos sistemas de comunicação móvel, o 6G, ainda está em fase de pesquisa e desenvolvimento, mas promete trazer ainda mais avanços. Espera-se que o 6G ofereça taxas de transmissão de dados ainda mais altas do que o 5G, latência ultra-baixa e novas funcionalidades, como a integração avançada com a inteligência artificial [3, 4, 5]. Isso permitirá a criação de sistemas de manufatura ainda mais eficientes, fornecendo novas possibilidades para a Indústria 4.0 e para a Indústria 5.0 [6].

1.2 COMUNICAÇÕES D2D

As comunicações *device-to-device* (D2D) são comunicações que permitem a troca direta de informações entre dispositivos sem a necessidade de um intermediário, como uma estação base. A comunicação D2D é uma tecnologia emergente que pode melhorar a eficiência da rede, reduzir a latência e aumentar a capacidade do sistema [7].

Esse tipo de comunicação foi integrado à arquitetura *Long Term Evolution - Advanced* (LTE-A) do 4G pelo *3rd Generation Partnership Project* (3GPP) na *Release* 12, inicialmente com foco em aplicações de segurança pública. As *Releases* subsequentes trouxeram melho-

rias e expansões para a funcionalidade D2D, incluindo suporte para comunicação veicular *vehicle-to-vehicle* (V2V) na *Release* 14 e a base para a aplicação dessas comunicações no 5G na *Release* 15. Apesar de ter sido incorporada na quarta geração dos sistemas de comunicação móvel, espera-se um aumento no uso de tais comunicações no 5G e no 6G [7].

A comunicação D2D é especialmente relevante no contexto da Indústria 4.0, pois pode permitir uma comunicação mais eficiente entre máquinas, entre operadores e entre operadores e máquinas em um ambiente de produção. Isso pode levar a melhorias significativas na eficiência operacional e na flexibilidade do sistema de produção. Um exemplo de aplicação seria os operadores se comunicarem para coordenar ações necessárias em uma linha de produção.

Considerando cenários *inband*, em que tanto as comunicações primárias quanto as com D2D utilizam o mesmo espectro licenciado, existem dois principais tipos de comunicações D2D: *underlay* e *overlay*.

Nas comunicações D2D *underlay*, os dispositivos D2D compartilham o mesmo espectro de frequência com as comunicações celulares tradicionais. Isso pode melhorar a eficiência espectral, mas também pode levar a interferências entre as comunicações D2D e celulares. Por outro lado, nas comunicações D2D *overlay*, os dispositivos D2D usam um espectro de frequências separado, o que evita a interferência entre as comunicações, mas também leva a um uso menos eficiente do espectro [7].

Assim, a implementação de comunicações D2D *underlay* se apresenta como boa alternativa para melhoria da eficiência espectral de um sistema, mas também apresenta desafios, como o controle da interferência entre dispositivos e a garantia de que as comunicações primárias, entre um celular e a Estação Rádio-Base, continuarão sendo realizadas de maneira satisfatória. Para superar esses desafios, é fundamental adotar estratégias inteligentes para a alocação dos recursos disponíveis, buscando garantir que os usuários possam aproveitá-los de maneira eficiente.

1.3 SENSOREAMENTO

Outras tecnologias que possuem papel importante na Indústria 4.0 são as de sensoreamento. Os sensores são dispositivos que coletam dados do ambiente, permitindo que sistemas automatizados respondam de maneira inteligente às condições em tempo real [8]. Na Indústria 4.0, os sensores podem ser usados para uma variedade de propósitos, desde o monitoramento da condição das máquinas até a otimização dos processos de produção [8].

Um exemplo específico de aplicação do sensoreamento na Indústria 4.0 é o sensoreamento espacial de máquinas e veículos. Neste contexto, os sensores podem ser usados para

rastrear a localização e o movimento de máquinas e veículos em um ambiente de produção. Isso permite um controle mais preciso e eficiente das operações, bem como a detecção precoce de problemas potenciais.

Historicamente, os sistemas de sensoreamento e comunicação móvel operam de forma independente, cada um com seu próprio conjunto de recursos e espectro. No entanto, com o advento das tecnologias 5G e 6G e a crescente demanda por recursos espectrais, surgiu uma tendência de unificação desses dois sistemas. A ideia é que, ao integrar a comunicação e o sensoreamento em um único sistema, seja possível otimizar o uso do espectro disponível, aumentar a eficiência na gestão dos recursos do sistema e permitir a implementação de novas funcionalidades [9].

As estratégias conjuntas de comunicação e sensoreamento, como JCAS (*Joint Communication and Sensing)* e RadCom (*Radar and Communication*), são abordagens que buscam integrar a comunicação e o sensoreamento em um único sistema [9].

Além da possibilidade de melhoria do uso do espectro, sistemas conjuntos permitem uma maior integração entre as funções de comunicação e sensoreamento [9]. Usando o exemplo supracitado, um mesmo sistema pode ser usado para comunicar informações entre máquinas e para monitorar as condições do ambiente de produção.

A Figura 1.1 ilustra uma célula de um sistema de comunicações móveis com convivência simultânea entre comunicações primárias, entre *User Equipment* (UE) e Estação Rádio Base (ERB), comunicações D2Ds e sensoreamento.

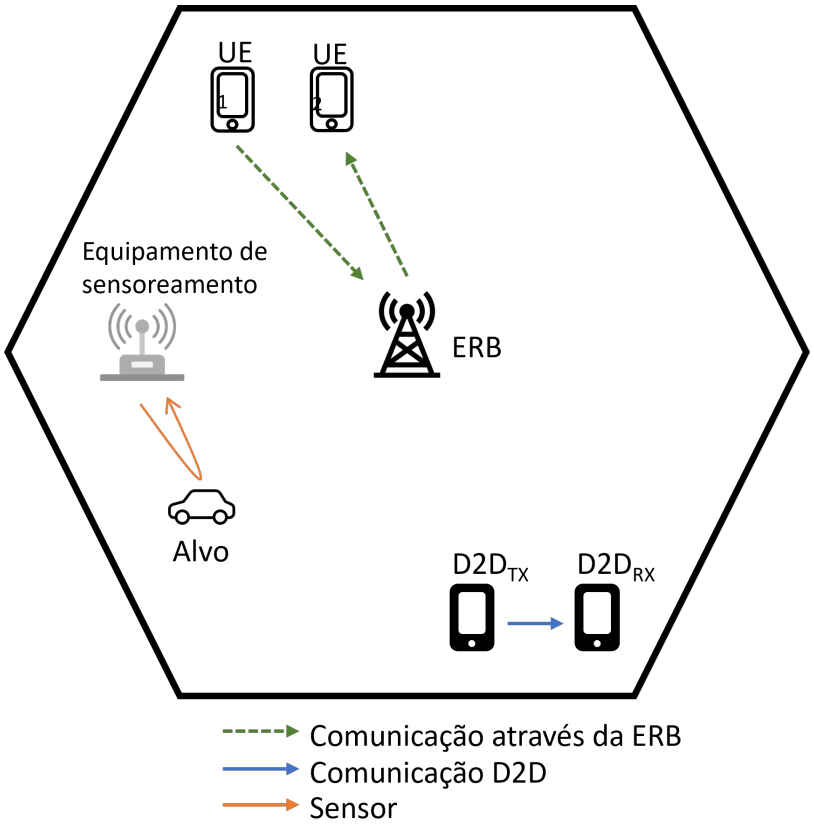


Figura 1.1 – Ilustração de uma célula com diferentes tipos de comunicação e sensores. Fonte: autoria própria.

Essas estratégias conjuntas de comunicação e sensoreamento abrem novas possibilidades para a Indústria 4.0, que permitem a criação de sistemas de produção mais inteligentes e eficientes, mas que apresentam desafios também. Um desafio é a necessidade de gerenciar a interferência entre as funções de comunicação e sensoreamento. Isso requer técnicas sofisticadas de alocação dos recursos disponíveis para garantir que ambas as funções possam operar efetivamente [9].

1.4 ALOCAÇÃO DE ESPECTRO E CONTROLE DE POTÊNCIAS

A alocação de recursos limitados do sistema, como espectro e energia, é um aspecto crítico para um bom funcionamento de um sistema JCAS, assim como de sistemas com comunicações D2D. Com um grande número de dispositivos conectados e uma variedade de funções a serem suportadas, é essencial gerenciar os recursos de forma eficiente [9, 10].

Em sistemas conjuntos que compartilham o espectro, a alocação de recursos é fundamental para garantir um controle eficiente da interferência gerada entre as comunicações e o sensoreamento [9, 10]. Entre os recursos disponíveis estão a largura de banda de frequências

ocupada e a energia de transmissão utilizada por cada comunicação/sensor.

Para alcançar o aumento da eficiência espectral que o JCAS possibilita, é necessário que se consiga garantir que ambos os sistemas funcionem adequadamente e que não se desperdicem recursos.

Em sistemas que contenham comunicações D2D *underlay*, a alocação se torna ainda mais desafiadora, uma vez que tais comunicações utilizam esses recursos de forma oportunística. Neste caso, o espectro estaria dividido para 3 usos diferentes, para as comunicações primárias, para as comunicações D2D e para o sensoreamento.

Nesse contexto, a estratégia utilizada precisa ser capaz de, em um ambiente em que existe a interferência entre todas as comunicações e sensoreamentos, atender às necessidades das comunicações primárias clássicas, isto é, entre um Equipamento do Usuário e a Estação Rádio Base, e do sensoreamento, além de buscar maximizar a taxa de transmissão das comunicações D2D.

Diante dos desafios apresentados para uma alocação eficiente de recursos em sistemas JCAS com D2D, as técnicas de Inteligência Artificial emergem como soluções promissoras [11]. O avanço contínuo nesta área tem possibilitado o desenvolvimento de algoritmos capazes de realizar tarefas de alta complexidade com notável capacidade de generalização [11].

Entre essas técnicas, destacam-se os algoritmos de Aprendizado por Reforço Profundo - *Deep Reinforcement Learning* (DRL). Estes modelos têm se mostrado eficazes para lidar com problemas que apresentam fortes não linearidades e não convexidade, características comuns em problemas de alocação de recursos [12].

Os algoritmos de DRL aprendem a tomar decisões através do recebimento de recompensas, adaptando-se e otimizando suas ações ao longo do tempo [13, 12]. Em contraste com as técnicas tradicionais de aprendizado supervisionado, que podem ser desafiadoras de aplicar quando a rotulação das entradas para treinamento não é trivial, os algoritmos de DRL não dependem da existência dessas variáveis resposta.

No contexto da alocação de recursos em sistemas JCAS, a dinamicidade do sistema é um desafio. Em paralelo, por se basear na solução de Problemas de Decisão de Markov (*Markov Decision Problems* - MDP), os algoritmos de DRL são capazes de se adaptarem de maneira dinâmica às alterações nas condições do problema. Tal característica desses algoritmos pode ser explorada para a tarefa em questão, a partir de sua implementação adequada.

No entanto, é importante ressaltar que as técnicas de DRL são sensíveis à dimensionalidade do espaço de estados do ambiente (que servem como entrada para o algoritmo) e à dimensionalidade do espaço de ações. Em cenários de múltiplas ações, onde o agente decide mais de uma ação simultaneamente, a complexidade de aprender a combinar tais ações

através de um único valor de recompensa aumenta conforme o número de ações a serem definidas também aumenta [14].

Os problemas de alocação de recursos em sistemas JCAS são problemas combinatoriais dinâmicos que podem ter alta dimensionalidade, a depender do número de dispositivos de comunicação e sensoreamento presentes no sistema. Essa característica requer a aplicação de outras técnicas em conjunto com os algoritmos de DRL para simplificação do problema [14, 12].

Dada a alta dimensionalidade, a dinamicidade e a natureza combinatória do problema de alocação de recursos, as Hiper-Heurísticas (HHs) emergem como uma solução promissora neste contexto [15, 16].

As HHs são métodos de busca heurística que visam automatizar o processo de seleção, combinação, geração ou adaptação de várias heurísticas mais simples (conhecidas como Heurísticas de Baixo Nível (*Low Level Heuristic* - LLH)) para resolver eficientemente problemas de busca computacional [16]. Este processo de seleção das LLHs é frequentemente realizado com a incorporação de técnicas de Aprendizado de Máquina (*Machine Learning* - ML) [17, 18].

As Hiper-Heurísticas foram desenvolvidas a partir da observação de que a combinação de diferentes heurísticas de baixo nível pode produzir soluções de melhor qualidade do que se fossem aplicadas isoladamente [19].

Com o desenvolvimento das técnicas de ML, as HHs ganharam ainda mais destaque, pois esses algoritmos podem ser usados para selecionar, combinar, gerar ou adaptar as LLHs, tornando a combinação de ambas as técnicas uma alternativa promissora para a solução de problemas complexos.

Neste trabalho, propõe-se uma solução conjunta composta por um algoritmo de DRL para controlar a potência das comunicações e sensores e por uma hiper-heurística para realizar a alocação do espectro disponível. O objetivo desta combinação é utilizar a hiper-heurística para lidar inicialmente com o problema, enfrentando sua alta dimensionalidade, e após a alocação do espectro, o uso de cada bloco de recurso é tratado pelo algoritmo de DRL. Isso reduz a dimensionalidade dos dados de entrada recebidos por esses algoritmos e possibilita um melhor desempenho em ambas as subtarefas.

1.5 TRABALHOS RELACIONADOS

Vários trabalhos foram realizados com propostas de algoritmos para solucionar os problemas abordados em ambientes com comunicações e sensoreamento conjuntos.

Em [20], os autores propõem um algoritmo para alocação do espectro e controle de potências em um sistema *Orthogonal Frequency Division Multiplexing* (OFDM) com presença simultânea de comunicação e sensoreamento. O objetivo do algoritmo proposto é maximizar a informação mútua (*Mutual Information* - MI) dos sensores e, oportunisticamente, das comunicações. No sistema modelado, não há sobreposição de sensoreamento e comunicação em um mesmo *Resource Block* (RB), isto é, não há compartilhamento de espectro.

Apesar de modelar um sistema sem compartilhamento simultâneo do espectro [20], o trabalho mostra que a alocação de recursos conjunta, levando-se em consideração a otimização tanto dos sensores quanto das comunicações, é capaz de alcançar maiores valores de MI para o sistema de comunicações como um todo, se comparado com sistemas desenvolvidos unicamente voltados para comunicações ou para sensores. Entretanto, o trabalho apresentado em [20] não explora contextos mais complexos, com comunicações de diferentes tipos, necessidades diferentes para comunicações e sensores, além de não explorar o compartilhamento do espectro entre comunicações e sensoreamento.

Já os autores de [21] e [22] propõem um algoritmo que faz tanto a alocação do espectro quanto o controle de potência em um sistema JCAS. Em [21], a proposta é usar otimização linear inteira mista (MILP) para designar RBs para as comunicações e sensores, bem como para regular a potência de cada um. O objetivo primário da otimização é maximizar a MI dos sensores e, de forma secundária, maximizar a taxa de transmissão das comunicações.

O trabalho apresentado em [21] reforça que existe ganho de MI para o sistema de comunicações ao se realizar uma alocação de recursos conjunta entre comunicações e sensores. Porém, o trabalho também não explora contextos com diferentes tipos de comunicações, não modela as restrições de requisição de taxa de transmissão mínima para as comunicações, além de apresentar seus resultados apenas em termos de MI, sem explorar outras métricas importantes para um sistema deste tipo, como a probabilidade de detecção dos sensores ou a SNIR das comunicações.

Por outro lado, em [22], a proposta é usar um algoritmo baseado em otimização convexa, visando minimizar a potência consumida pelo sistema, respeitando os requisitos definidos para a MI dos sensores e para a taxa de transmissão das comunicações. Em ambos os trabalhos, também não há sobreposição de sensores e comunicações em um mesmo RB.

O trabalho mostra que a alocação de recursos conjunta realizada pelo algoritmo proposto reduz o consumo de potência do sistema, mantendo os requisitos mínimos para as comunicações e sensores. Dessa forma, os autores de [22] demonstram que a divisão do problema de alocação de recursos em subproblemas consegue ter desempenho superior a alguns *benchmarks* da literatura. Entretanto, a proposta apresentada não foi testada em cenários mais complexos, em que os canais de comunicação possuem incertezas, além de não modelar comunicações oportunísticas no problema, como as comunicações D2D.

Em [23] e [24], o foco é desenvolver algoritmos capazes de fazer a alocação de recursos em sistemas com compartilhamento de espectro entre comunicações e sensores.

Em [23], o algoritmo proposto é baseado em Programação Sequencial Convexa e faz o controle de potência tanto das comunicações quanto dos sensores do sistema com o objetivo de maximizar a SNIR dos sensores, respeitando os requisitos de taxa de transmissão mínima das comunicações.

Já em [24], os autores propõem a solução tanto para o controle de potências quanto para a alocação do espectro por meio de uma decomposição do problema com posterior aplicação de algoritmos de Programação Sequencial Convexa. O objetivo do algoritmo proposto é maximizar a taxa de transmissão do sistema.

O trabalho de [24] mostra que é possível desenvolver estratégias que visam maximizar a taxa de transmissão das comunicações em um sistema de comunicações com interferência cruzada entre comunicações e sensores, garantindo os requisitos mínimos dos sensores. Apesar disso, o sistema modelado é muito simples, já que não simula um sistema com modulação OFDM e possui poucas subportadoras para alocação, além de não modelar outros tipos de comunicação além das tradicionais no sistema, como as comunicações D2D, com suas peculiaridades e necessidades específicas.

Em [25] e [26], os autores propõem um algoritmo para controle de potências em sistemas com canal MIMO, com compartilhamento de espectro entre sensoreamento e comunicação.

Os autores de [25] propõem um algoritmo para controle de potências baseado em programação fracionária, cujo objetivo é maximizar a SNIR dos sensores, respeitando os requisitos de taxa de transmissão mínima das comunicações.

Já em [26], os autores propõem um algoritmo baseado em teoria de jogos, cujo objetivo é minimizar a potência consumida, desde que respeitados os requisitos de probabilidade de detecção dos sensores e de taxa de transmissão das comunicações.

Em [26], a modelagem do problema é mais completa, com incerteza nos canais das comunicações e sensores, com requisitos para os sensores relacionados à probabilidade de detecção dos alvos sensoreados, além de modelar um sistema de comunicações MIMO que precisa ser protegido da interferência gerada pelos equipamentos de sensoreamento. Os resultados alcançados mostram que o algoritmo converge, mas não explora bem os ganhos da estratégia proposta em comparação com outros algoritmos. Além disso, o trabalho não modela comunicações D2D e suas especificidades relacionadas à localização variante tanto dos dispositivos transmissores quanto dos receptores, além de não explorar a alocação do espectro como estratégia para minimizar a interferência entre comunicações e sensores.

Neste atual trabalho, propõe-se, de forma pioneira, até onde o conhecimento do autor alcança, um algoritmo para alocação do espectro e controle de potências em um sistema com

sensores compartilhando o espectro com 2 tipos distintos de comunicação, comunicações primárias e D2Ds.

Cada comunicação será modelada segundo as suas especificidades, explorando-se, assim, diferentes possibilidades de otimização e levando-se em consideração os requisitos de cada uma delas, bem como os dos sensores. Nesse contexto, os algoritmos apresentados nos trabalhos anteriores não são capazes de solucionar diretamente o problema em questão, devido à especificidade do problema ao se adicionar mais um tipo de comunicação, assim como pela não linearidade e não convexidade presente no problema modelado de forma mais completa.

Em adição, o sistema foi modelado em um contexto relacionado a aplicações da Indústria 4.0, utilizando-se diferentes tipos de comunicação e sensores para simular cenários mais próximos das aplicações reais, contextualizando-se o sistema criado em aplicações esperadas para serem suportadas pelos sistemas do 5G e do 6G.

Além disso, o trabalho propõe uma solução para um problema não linear e não convexo a partir da implementação de dois tipos de algoritmos que não foram explorados na literatura para alocação de espectro e potência em sistemas JCAS: algoritmos de DRL e as Hiper-heurísticas. A solução proposta foi desenvolvida para ser flexível aos diferentes cenários que o sistema pode se encontrar, assim como escalável para ser utilizada em contextos com necessidades em tempo real.

A Tabela 1.1 reúne os principais fatores de comparação entre os trabalhos relacionados, com propostas para alocação de recursos em sistemas com comunicações móveis e sensoreamento no *uplink*, e a proposta feita neste trabalho.

Tabela 1.1 – Tabela comparativa entre os trabalhos relacionados e a proposta deste trabalho

Trabalho	Algoritmo Proposto	Alocação de espectro	Controle de potência	Presença de D2Ds	Métricas Avaliadas
[20]	Derivação analítica	✓	✓	✗	MI
[21]	MILP	✓	✓	✗	MI
[22]	Otimização convexa	✓	✓	✗	Potência consumida e MI
[23]	Programação Sequencial Convexa	✗	✓	✗	SNIR e taxa de transmissão
[24]	Programação Sequencial Convexa	✓	✓	✗	Taxa de transmissão
[25]	Programação fracionária	✗	✓	✗	SNIR e taxa de transmissão
[26]	Teoria de jogos	✗	✓	✗	Potência consumida, probabilidade de detecção e taxa de transmissão
Proposta	DRL e Hiper-Heurísticas	✓	✓	✓	Taxa de *outage*, probabilidade de detecção e taxa de transmissão

1.6 CONTRIBUIÇÕES

De forma resumida, pode ser considerado que entre as principais contribuições deste trabalho estão:

- Revisão da literatura sobre alocação de recursos em redes de comunicação móvel com sensoreamento e com comunicações D2D;
- Discussão e avaliação da integração inovadora de comunicações primárias, sensoreamento e comunicações D2D em contextos relacionados à Indústria 4.0;
- Proposta e avaliação de algoritmo para controle de potências baseado em técnicas de DRL, comparando diferentes algoritmos no estado da arte;
- Proposta e avaliação de algoritmo para alocação de espectro, baseado na integração entre técnicas de DRL e Hiper-Heurísticas.

Dentre os trabalhos que foram realizados no âmbito do esforço de pesquisa, inclui-se o trabalho publicado no periódico *International Journal of Communication Systems*, conforme se apresenta no Apêndice A.

1.7 ORGANIZAÇÃO DA DISSERTAÇÃO

O texto desta dissertação está organizado da seguinte forma:

- O Capítulo 2 explica o problema abordado, com foco no detalhamento matemático para realizar a alocação do espectro e o controle de potências, além de explicar a modelagem do sistema de comunicações utilizada para a realização das simulações;
- O Capítulo 3 detalha o desenvolvimento de um algoritmo para controle de potências das comunicações e sensores do sistema, discorrendo sobre o algoritmo treinado e os resultados obtidos;
- O Capítulo 4 expõe a forma como o algoritmo de alocação do espectro foi desenvolvido, desde a configuração da hiper-heurística implementada, até a análise do desempenho obtido.
- Por fim, o Capítulo 5 reúne as principais conclusões obtidas a partir do trabalho realizado, assim como elenca pontos que podem ser abordados para evolução em trabalhos futuros.

1.8 CONCLUSÃO

A alocação de recursos em sistemas de comunicações de quinta geração é uma tarefa fundamental para o seu bom funcionamento. Com a inserção de comunicações D2D no sistema, tal tarefa se tornou mais desafiadora, mas também mais necessária. Espera-se que para o 6G e para sistemas JCAS tal necessidade aumente ainda mais, pelo aumento da quantidade de dispositivos comunicantes, assim como pela inserção dos sensores junto com as comunicações.

Nesse cenário, será apresentado o desenvolvimento de técnicas analíticas que melhoram o desempenho de tais sistemas. Com a evolução das técnicas de Inteligência Artificial, tais modelos se tornam opções interessantes para serem aplicadas em tais contextos.

Este trabalho, então, propõe uma estratégia conjunta, composta por dois algoritmos de naturezas diferentes, que realize a alocação de recursos de um sistema JCAS, atuando tanto no controle de potências quando na alocação do espectro. Esta estratégia utiliza algoritmos de DRL para o controle de potências e algoritmos baseados em HHs em conjunto com outros de DRL para alocação de espectro. O objetivo de tal estratégia é controlar as interferências geradas entre os diferentes tipos de comunicação e sensores para garantir o funcionamento adequado do sistema.

2 PROBLEMA

2.1 APLICAÇÃO

Com o avanço do 5G e do 6G, espera-se que tais tecnologias sejam utilizadas para implementação de várias inovações esperadas para a chamada Indústria 4.0. Nesse âmbito, espera-se um aumento massivo das comunicações em ambientes fabris, seja entre usuários, seja entre máquinas [27, 28]. Além disso, é esperado um aumento significativo de atividades de sensoreamento em tais ambientes, seja para a automatização de processos, seja para gerenciamento dos mesmos [27, 28].

Nesse cenário, a aplicação do estudo foi uma modelagem de um ambiente fabril, em que existem máquinas e funcionários trabalhando em um pátio central que possui formato quadrado com lados de 600 metros, que pode ser acessado por quatro vias que cruzam o pátio diagonalmente. Tais vias são utilizadas por veículos que transportam carga e funcionários do ambiente externo para o pátio. Apesar de estarem concentrados principalmente nos pátios, é possível que existam máquinas e/ou funcionários no entorno do pátio.

Para acompanhar o tráfego de veículos, foi instalado um sistema de sensoreamento rodoviário às margens das vias de acesso ao pátio. O objetivo é identificar um veículo que esteja em trânsito na via, contabilizando a entrada e saída dos mesmos.

Esse ambiente está situado no centro de uma célula do sistema de comunicações móveis, que fornece a estrutura básica para estabelecimento conjunto das comunicações móveis e do sensoreamento no ambiente. Tal sistema será detalhado a seguir.

2.2 SISTEMA DE COMUNICAÇÕES

O sistema de comunicações modelado consiste em uma única célula que envolve diferentes tipos de comunicações. Ele realiza até J comunicações, formando um conjunto $\mathbf{J} = \{1, 2, ..., J\}$ de comunicações, entre as quais A são comunicações primárias no uplink, formando o conjunto $\mathbf{A} = \{1, 2, ..., A\}$, que ocorrem entre um UE e uma ERB, e L são comunicações D2D *in-band underlay*, formando o conjunto $\mathbf{L} = \{1, 2, ..., L\}$, que são comunicações diretas entre dispositivos dentro da mesma faixa de frequência. O sistema também inclui Q sensores responsáveis pelo sensoreamento de veículos, formando o conjunto de sensores $\mathbf{Q} = \{1, 2, ..., Q\}$. Nesse sistema, $J = A + L$ e não há restrições quanto ao valor de Q.

No sistema, estão presentes M UEs, formando o conjunto $\mathbf{M} = \{1, 2, ..., M\}$), N equipamentos de sensoreamento com transmissores e receptores embutidos, formando o conjunto $\mathbf{N} = \{1, 2, ..., N\}$ e C veículos alvo a serem monitorados, formando o conjunto $\mathbf{C} = \{1, 2, ..., C\}$. Todos esses dispositivos estão envolvidos nas comunicações e sensoreamento mencionados anteriormente.

O sistema é baseado na tecnologia OFDM e utiliza K RBs, formando o conjunto $\mathbf{K} = \{1, 2, ..., K\}$, em que $K \leq A \leq J$. O OFDM é uma técnica de multiplexação/modulação que divide o sinal em múltiplas subportadoras ortogonais para aumentar a eficiência espectral e melhorar a robustez contra os efeitos do canal com desvanecimento de pequena escala seletivo em frequência (denominado na literatura de canal banda larga)

A modelagem da célula de comunicações foi realizada considerando um cenário onde a célula possui um formato quadrado com lados de 1 km, estando a ERB localizada no centro, e sendo o quadrado concêntrico ao pátio fabril.

Nesse ambiente, existem quatro ($N = 4$) equipamentos de sensoreamento posicionados nas margens das vias, próximos aos vértices da célula. Esses sensores desempenham o papel de detectar e monitorar o tráfego dos veículos. A Figura 2.1 ilustra de forma simplificada a estrutura da célula modelada.

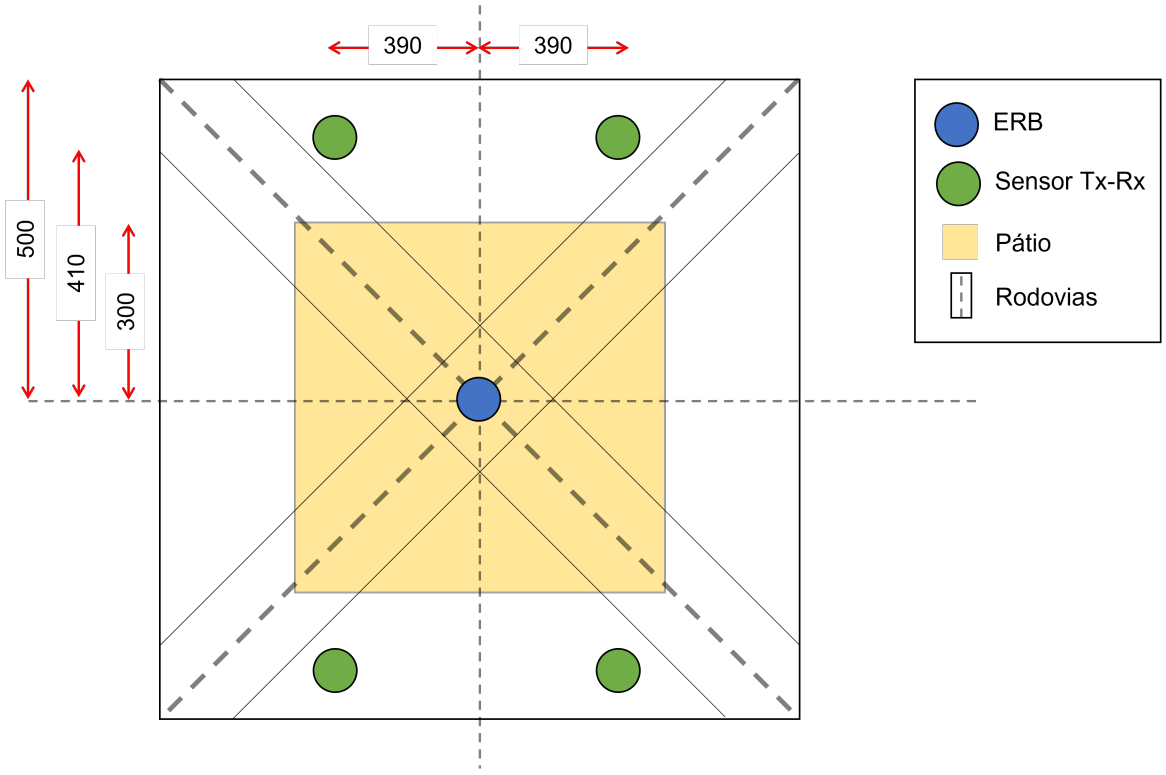


Figura 2.1 – Representação da célula modelada. Fonte: autoria própria.

Apesar de estarem concentrados principalmente no pátio, os UEs têm a liberdade de se moverem para qualquer posição dentro da célula, enquanto os sensores permanecem estáti-

cos e os veículos alvos do monitoramento se deslocam apenas ao longo das vias.

O modelo de mobilidade adotado para os UEs foi o movimento browniano [29]. Por outro lado, os veículos alvos do sensoreamento seguem trajetórias retilíneas com velocidade uniforme ao longo das vias.

Caso algum dispositivo ou veículo alcance a borda da célula, a trajetória é refletida de forma que os ângulos de reflexão e incidência possuam o mesmo valor. Essa reflexão segue o princípio da Primeira Lei da Reflexão, que estabelece que o ângulo de incidência é igual ao ângulo de reflexão.

2.2.1 Canal

No modelo implementado, foi considerado que cada comunicação possui uma fila infinita de pacotes a serem transmitidos. Essa abordagem permite simular o fluxo contínuo de dados entre os dispositivos, onde os pacotes são enfileirados para transmissão e enviados sequencialmente. Quanto aos sensores, estes só são ativados a partir do momento em que um alvo entra na célula, mantendo-se ligados continuamente até que o alvo saia da região da aplicação.

O sistema utiliza a banda mmWave, mais especificamente a banda de 28 GHz, em que tanto as comunicações primárias quanto as comunicações D2D e os sensores utilizam o mesmo canal de comunicação.

No sistema modelado, os dispositivos de comunicação e os sensores podem compartilhar um mesmo RB simultaneamente, mas só podem ser alocados em somente um RB ao mesmo tempo. Dessa forma, o sinal recebido no k-ésimo RB pelo receptor da j-ésima comunicação é definido pela Equação (2.1), enquanto o sinal recebido no k-ésimo RB pelo receptor do q-ésimo sensor é definido pela Equação (2.2).

$$y_{j,k} = \sqrt{p^t_{j,k}} h^d_{j,j,k} x_{j,k} + \sum_{i=1, i\neq j}^{J} \sqrt{p^t_{i,k}} h^d_{i,j,k} x_{i,k} + \sum_{q=1}^{Q} \sqrt{p^t_{q,k}} h^d_{q,j,k} x_{q,k} + \eta_{j,k} \tag{2.1}$$

$$y_{q,k} = \sqrt{p^t_{q,k}} h^e_{q,q,k} x_{q,k} + \sum_{j=1}^{J} \sqrt{p^t_{j,k}} h^d_{j,q,k} x_{j,k} + \sum_{i=1, i\neq q}^{Q} \sqrt{p^t_{i,k}} h^d_{i,q,k} x_{i,k} + \eta_{q,k} \tag{2.2}$$

em que $y_{[\cdot],k}$ é o sinal recebido pela comunicação/sensor $[\cdot]$ no k-ésimo RB, $p^t_{[\cdot],k}$ é a potência de transmissão da comunicação/sensor $[\cdot]$ no k-ésimo RB, $h^d_{[\cdot],[\star],k}$ é o ganho de propagação do canal direto entre o transmissor da comunicação/sensor $[\cdot]$ e o receptor da comunicação/-

sensor $[\star]$ no k-ésimo RB, $h^{e}_{[\cdot],[\star],k}$ é o ganho de propagação do canal com reflexão no alvo entre o transmissor da comunicação/sensor $[\cdot]$ e o receptor da comunicação/sensor $[\star]$ no k-ésimo RB, $x_{[\cdot],k}$ é o símbolo transmitido pela comunicação/sensor $[\cdot]$ no k-ésimo RB e $\eta_{[\cdot],k}$ é o ruído recebido pela comunicação/sensor $[\cdot]$ no k-ésimo RB.

Em cada equação mostrada acima, o 1º termo do lado direito da equação representa o sinal que o transmissor enviou para o receptor da comunicação/sensor, o 2º termo representa a interferência gerada por outras comunicações que estejam compartilhando o RB, o 3º termo, a interferência gerada pelos sensores que estejam no RB e o 4º termo é o ruído AWGN presente no canal.

A interferência gerada pela reflexão dos alvos dos sensores para outras comunicações e sensores foi desconsiderada, assumindo-se que a intensidade dessa fonte de interferência tende a ser menor que as demais e que foram empregadas técnicas de mitigação dessa interferência para cancelá-la [26, 30].

Assim, a Relação Sinal Ruído mais Interferência (*Signal-to-Noise-plus-Interference Ratio* (SNIR)) da j-ésima comunicação do sistema é definida pela Equação (2.3), ao passo que a SNIR do q-ésimo sensor do sistema no k-ésimo RB é definido pela Equação (2.4).

$$\zeta_{j,k} = \frac{p^{t}_{j,k}|h^{d}_{j,j,k}|^2}{\sum_{i=1,i\neq j}^{J} p^{t}_{i,k}|h^{d}_{i,j,k}|^2 + \sum_{q=1}^{Q} p^{t}_{q,k}|h^{d}_{q,j,k}|^2 + \eta^2_{j,k}} \tag{2.3}$$

$$\zeta_{q,k} = \frac{p^{t}_{q,k}|h^{e}_{q,q,k}|^2}{\sum_{j=1}^{J} p^{t}_{j,k}|h^{d}_{j,q,k}|^2 + \sum_{i=1,i\neq q}^{Q} p^{t}_{i,k}|h^{d}_{i,q,k}|^2 + \eta^2_{q,k}} \tag{2.4}$$

Assim, $\zeta_{[\cdot],k}$ é a SNIR do canal da comunicação/sensor $[\cdot]$ no k-ésimo RB.

A modelagem do *path loss* utilizado foi baseada na especificação 3GPP TR 38.901. Essa especificação foi desenvolvida pelo 3GPP e define modelos de propagação adequados para cenários do 5G, especialmente na faixa de frequência de ondas milimétricas (mmWave), como a banda de 28 GHz.

O modelo de propagação escolhido foi especialmente projetado para cenários com características de quadrado aberto com linha de visada (*Line-of-Sight* (LOS)), adequado para o problema. A Equação (2.5) é utilizada para o cálculo do *path loss* [31].

$$PL[\text{dB}] = 32.4 + 18.5\log_{10}(d) + 20\log_{10}(f_c) \tag{2.5}$$

em que d é a distância entre os dispositivos transmissor e receptor em metros e f_c é a frequência da portadora em GHz.

Assim, de maneira simplificada, o ganho de propagação do canal direto em dB (H^d) e o

ganho de propagação do canal com reflexão em um alvo (H^e), também em dB, são definidos pela Equação (2.6) e pela Equação (2.7), respectivamente.

$$H^d[\text{dB}] = G_t + G_r - (PL + SF), \tag{2.6}$$

$$H^e[\text{dB}] = G_t + G_r + \sigma_{RCS} - (PL_{T-A} + PL_{A-R} + SF), \tag{2.7}$$

em que G_t é o ganho da antena de transmissão do dispositivo transmissor, G_r é o ganho da antena de recepção do dispositivo receptor, σ_{RCS} é o *Radar cross section* (RCS) dos alvos do sensoreamento, PL_{T-A} é o *path loss* entre o transmissor de sensoriamento e o alvo sensoreado, PL_{A-R} é o *path loss* entre o alvo do sensoriamento e o sensor receptor [26] e SF (*Shadow Fading Loss*) é a perda ou ganho devido a fatores de sombreamento, que segue uma distribuição log-normal $\mathcal{N}(0, 4.2)$ [31, 32].

2.2.2 Requisitos

Um sistema de comunicações móveis possui requisitos que devem ser atendidos. Um deles é garantir que as comunicações primárias tenham um canal com uma eficiência espectral maior ou igual à eficiência espectral mínima para que o sistema consiga garantir a taxa de transmissão requisitada pelo serviço que está sendo entregue [33, 34], isto é:

$$\psi_a = \sum_{k}^{K} \log_2(1 + \zeta_{a,k}) \geq \psi_{\text{min}} \tag{2.8}$$

em que ψ_a é a eficiência espectral da a-ésima comunicação primária, $\zeta_{[\cdot],k}$ é a SNIR do canal da comunicação/sensor $[\cdot]$ no k-ésimo RB (Equação (2.3)) e ψ_{min} é a eficiencia espectral mínima para as comunicações primárias.

Por outro lado, para o sensoreamento, é fundamental garantir que a probabilidade de detecção dos veículos esteja acima de um determinado limiar ϕ_{min} definido pelo projetista do sistema [26, 35].

Na modelagem proposta, cada sensor identifica a presença de um veículo a partir do *Generalized Likelihood Ratio Test* (GLRT) [36]. Assim, a probabilidade de detecção de um alvo pode ser calculada pela Equação (2.9).

$$P_q^d = (1 + \frac{\lambda}{1-\lambda}\frac{1}{1+w\zeta_q})^{1-w} \tag{2.9}$$

em que P_q^d é a probabilidade de detecção de um alvo existente, λ é o limiar de detecção, ζ_q

é a SNIR do canal vivenciado pelo q-ésimo sensor e w é o número de pulsos recebido por cada sensor durante o *dwell time*, que é o período de tempo em que o alvo sensoreado está sendo iluminado pela onda emitida pelo dispositivo transmissor do sensor.

Já as comunicações D2D foram modeladas sem requisitos mínimos no sistema, de forma que o objetivo seja sempre maximizar a eficiência espectral de tais comunicações, mas de forma oportunística, isto é, explorando recursos não utilizados para atender às requisições mínimas das comunicações primárias e dos sensores.

2.2.3 Alocação de recursos

A alocação de recursos é um componente crucial na implementação eficiente das tecnologias 5G e 6G, especialmente no contexto da Indústria 4.0 [3]. Entre os recursos disponíveis para alocação em tais sistemas de comunicação estão o espectro e a potência para transmissão.

O espectro é um recurso essencial que determina a quantidade de dados que podem ser transmitidos simultaneamente, além de possibilitar a mitigação de interferência em sistemas OFDM. A potência de transmissão, por outro lado, afeta a qualidade do sinal e, consequentemente, a qualidade da comunicação.

O acesso a esses parâmetros de otimização é feito dinamicamente, permitindo que o sistema se adapte às mudanças nas condições da rede e às demandas dos usuários. Por exemplo, o espectro pode ser realocado para mitigação de interferência entre comunicações, ou a potência de transmissão pode ser ajustada para manter a qualidade do sinal em face de interferências.

Essa alocação do espectro e da potência é feita dinamicamente a partir do estado do canal de comunicações. Neste trabalho, considerou-se que a Informação do Estado do Canal (*Channel State Information* - CSI) é totalmente conhecida pela ERB.

Dessa forma, através do envio de símbolos piloto, a cada intervalo de tempo pré definido (*timestep*), o sistema amostra o ganho de propagação do canal direto (H^d) entre o dispositivo transmissor e receptor de cada comunicação e o ganho de propagação do canal com reflexão (H^e) de cada sensor. Além disso, a partir dessa técnica, obtém-se o ganho de propagação dos canais interferentes, tanto das comunicações quanto dos sensores. Assim, o algoritmo de alocação recebe uma matriz como a mostrada abaixo.

$$\begin{bmatrix} H^d_{1,1} & \cdots & H^d_{1,j} & \cdots & H^d_{1,J} & H^d_{1,J+1} & \cdots & H^d_{1,J+q} & \cdots & H^d_{1,J+Q} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ H^d_{j,1} & \cdots & H^d_{j,j} & \cdots & H^d_{j,J} & H^d_{j,J+1} & \cdots & H^d_{j,J+q} & \cdots & H^d_{j,J+Q} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ H^d_{J,1} & \cdots & H^d_{J,j} & \cdots & H^d_{J,J} & H^d_{J,J+1} & \cdots & H^d_{J,J+q} & \cdots & H^d_{J,J+Q} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ H^d_{J+1,1} & \cdots & H^d_{J+1,j} & \cdots & H^d_{J+1,J} & H^e_{J+1,J+1} & \cdots & H^d_{J+1,J+q} & \cdots & H^d_{J+1J+,Q} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ H^d_{J+q,1} & \cdots & H^d_{J+q,j} & \cdots & H^d_{J+q,J} & H^d_{J+q,J+1} & \cdots & H^e_{J+q,J+q} & \cdots & H^d_{J+q,J+Q} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ H^d_{J+Q,1} & \cdots & H^d_{J+Q,j} & \cdots & H^d_{J+Q,J} & H^d_{J+Q,J+1} & \cdots & H^d_{J+Q,J+q} & \cdots & H^e_{J+Q,J+Q} \end{bmatrix}$$

em que $H_{x,y}$ é o ganho de propagação do canal entre o transmissor da comunicação/sensor x e o receptor da comunicação/sensor y.

Essas informações são fundamentais para entender quais comunicações geram mais interferência entre si e quais possuem um canal com maior ganho. Com base nesses dados, o sistema pode alocar os recursos disponíveis de forma mais eficiente. O objetivo é assegurar os requisitos do sistema, como a qualidade da comunicação e a minimização da interferência, ao mesmo tempo que maximiza a eficiência espectral.

O ganho de propagação dos canais do sistema variam com o tempo, já que os dispositivos estão se movendo pelo ambiente e por conta de outros efeitos aleatórios como o *shadowing* e o desvanescimento de pequena escala. Assim, este valor amostrado varia a cada *timestep*, o que faz com que seja necessária uma realocação dos recursos para otimização do sistema considerando o novo estado do sistema.

Portanto, a alocação de recursos é um processo dinâmico e adaptativo que leva em consideração as condições atuais do canal de comunicação. Isso permite que o sistema responda efetivamente às mudanças nas condições da rede e às demandas dos usuários, garantindo a operação eficiente e confiável do sistema de comunicação.

2.3 MODELAGEM DO PROBLEMA

O processo decisório para a alocação dos recursos é executado a cada *timestep*, de forma que a divisão desses recursos seja atualizada à medida em que as condições do sistema mudem. Desta forma, o problema de alocação dos recursos em um *timestep* pode ser descrito matematicamente como:

$$\max_{b_{j,k},b_{q,k},p_{j,k},p_{q,k}} \sum_{l=1}^{L} \psi_l = \sum_{l=1}^{L}\sum_{k=1}^{K} \log_2(1 + b_{l,k}\zeta_{l,k}) \tag{2.10}$$

$$\text{s.t. } \psi_a \geq \psi_{\min}, \quad \forall\, a \in \mathbf{A} \tag{2.11}$$

$$P_q^d \geq \phi_{\min}, \quad \forall\, q \in \mathbf{Q} \tag{2.12}$$

$$p_{j,k} \leq p_{\max}, \quad \forall\, j \in \mathbf{J} \text{ e } \forall\, k \in \mathbf{K} \tag{2.13}$$

$$p_{q,k} \leq p_{\max}, \quad \forall\, q \in \mathbf{Q} \text{ e } \forall\, k \in \mathbf{K} \tag{2.14}$$

$$\sum_{k=1}^{K} b_{j,k} = 1, \quad \forall\, j \in \mathbf{J} \tag{2.15}$$

$$\sum_{k=1}^{K} b_{q,k} = 1, \quad \forall\, q \in \mathbf{Q} \tag{2.16}$$

em que:

$$b_{j,k} \in [0,1], \quad \forall\, j \in \mathbf{J} \text{ e } \forall\, k \in \mathbf{K} \tag{2.17}$$

$$b_{q,k} \in [0,1], \quad \forall\, q \in \mathbf{Q} \text{ e } \forall\, k \in \mathbf{K} \tag{2.18}$$

$$p_{j,k} \in \mathbb{R}, \quad \forall\, j \in \mathbf{J} \text{ e } \forall\, k \in \mathbf{K} \tag{2.19}$$

$$p_{q,k} \in \mathbb{R}, \quad \forall\, q \in \mathbf{Q} \text{ e } \forall\, k \in \mathbf{K} \tag{2.20}$$

$$\mathbf{J} = \mathbf{A} \cup \mathbf{L} \tag{2.21}$$

onde $b_{[\cdot],k}$ é a variável indicadora se a comunicação/sensor $[\cdot]$ será alocada no k-ésimo RB e $p_{\max}$ é o limite superior da potência de transmissão alocada para uma comunicação ou sensor.

Na modelagem proposta é possível destacar que:

- A Equação (2.10) define o objetivo do problema, que é maximizar a eficiência espectral das comunicações D2D;
- A Inequação (2.11) define que a eficiência espectral das comunicações primárias deve ser maior do que $\psi_{\min}$;
- A Inequação (2.12) define que a probabilidade de detecção dos sensores deve ser maior do que $\phi_{\min}$;
- As Inequações (2.13) e (2.14), respectivamente, definem que a potência alocada para cada comunicação e cada sensor devem ser menores que $p_{\max}$;
- As Equações (2.15) e (2.16) definem que cada comunicação ou sensor será alocado, respectivamente, em um único RB;
- A Equação (2.17) e a Equação (2.18) definem $b_{[\cdot],k}$ como variáveis binárias tanto para as comunicações quanto para os sensores;

- A Equação (2.19) e a Equação (2.20) definem $p^t_{[\cdot],k}$ como variáveis contínuas tanto para as comunicações quanto para os sensores, respectivamente;
- A Equação (2.21) define que o conjunto de comunicações é a união do conjunto de comunicações primárias e comunicações D2D.

A partir da descrição do problema, é possível classificá-lo como um problema combinatório não linear [37, 12]. Problemas desse tipo, assim como diversos problemas combinatórios, são naturalmente *NP-hard* [12].

Essas características do problema dificultam o uso de algoritmos clássicos de otimização matemática, uma vez que a não linearidade inviabiliza o uso de técnicas de Programação Linear Inteira Mista (MILP) como Simplex e *Cutting planes* [37].

Por outro lado, o fato de ser um problema com variáveis inteiras não convexo faz com que algoritmos clássicos de otimização não linear, como aqueles que utilizam o Método do Gradiente ou o Método de Newton, não sejam adequados [37].

Além disso, a alocação dos recursos deve ser feita dinamicamente com baixa latência, o que exige que o algoritmo aplicado resolva o problema de maneira rápida, para que a solução possa ser utilizada em tempo viável. Essa característica do problema dificulta a aplicação de algoritmos de busca estocástica, como Algoritmos Genéticos ou o *Simulated Annealing* [38, 39].

Devido à complexidade do problema, desenvolver um único algoritmo capaz de solucioná-lo completamente é difícil. Nesse cenário, a proposta de solução a ser apresentada separa este problema em 2 subproblemas, de forma que o primeiro seja o controle de potências e o segundo a alocação do espectro.

É possível dividir o problema ao se utilizar do fato de que a modulação OFDM é ortogonal, o que faz com que as comunicações e os sensores de um RB não gerem interferência nas comunicações e nos sensores de outro RB. Dessa forma, o controle de potências de um RB é independente do controle de potência de outro RB. Essa característica do sistema permite que as comunicações e sensores sejam alocados nos RBs disponíveis para, posteriormente, o controle de potência ser executado em cada RB separadamente.

Cada um dos próximos 2 capítulos deste trabalho será dedicado a cada um destes subproblemas, detalhando a solução desenvolvida para os subproblemas e, finalmente, a solução final para o problema completo.

2.4 CONCLUSÃO

O problema apresentado neste capítulo diz respeito a uma questão em aberto dos sistemas de comunicações modernos, que é a alocação de recursos em sistemas com comunicações e sensoreamento compartilhando os mesmos recursos do sistema.

A partir da modelagem do sistema, do canal e dos requisitos esperados para as comunicações e sensores do sistema, o problema foi definido matematicamente. A partir de tal definição, é perceptível que o problema é *NP-hard*, envolvendo características dinâmicas, não lineares, não convexas, combinatoriais e de alta dimensionalidade.

Nesse cenário, o problema será segmentado em 2 subproblemas (o controle de potências e a alocação do espectro) para possibilitar o desenvolvimento de soluções viáveis que respeitem as restrições do sistema em tempo hábil de atuação para o sistema. Tais soluções desenvolvidas para cada subproblema serão explicadas ao longo dos capítulos 3 e 4, utilizando estratégias diferentes.

3 CONTROLE DE POTÊNCIA

3.1 INTRODUÇÃO

A evolução das redes de comunicações móveis, particularmente as de quinta (5G) e sexta gerações (6G), destaca a tarefa de controle de potência como um aspecto crucial para otimizar o desempenho e a eficiência espectral dos sistemas. O controle de potência envolve a regulação da potência transmitida tanto no *uplink* quanto no *downlink*, com o objetivo de manter uma qualidade de serviço adequada (QoS) para as comunicações do sistema. Para isso, o controle de potência deve ser feito visando à minimização da interferência entre os usuários e, ao mesmo tempo, à maximização da eficiência espectral do sistema.

Em sistemas JCAS (*Joint Communication and Sensing)*, o controle de potências se torna ainda mais importante, uma vez que o sensoreamento divide os recursos com as comunicações, sejam primárias ou D2Ds, exigindo técnicas mais sofisticadas para garantia da QoS.

Especialmente em contextos 5G e 6G, a importância do controle de potências deriva de várias necessidades e expectativas crescentes em relação às redes de comunicações móveis. A demanda por maior largura de banda e a necessidade de suportar um número cada vez maior de dispositivos conectados, incluindo IoT (Internet das Coisas), requer um gerenciamento de recursos eficiente.

O interesse por técnicas de controle de potência mais eficientes surge como uma resposta direta a esses desafios. Técnicas avançadas de controle de potência podem ajudar a otimizar o uso do espectro, melhorar a experiência dos usuários e possibilitar novas aplicações para esses sistemas. Isso é particularmente relevante para o 5G e o emergente 6G, que prometem altas velocidades de transmissão de dados, baixa latência e alta confiabilidade.

No entanto, o desenvolvimento de técnicas eficazes de controle de potência enfrenta vários desafios. Um dos principais é a complexidade inerente aos sistemas 5G e 6G, que incorporam uma variedade de tecnologias, como o MIMO (*Multiple Input Multiple Output*) e o OFDM. A alta dinamicidade desses sistemas, juntamente com a sua alta dimensionalidade, com um número grande e variável de dispositivos comunicantes no sistema, tornam o controle de potência uma tarefa desafiadora.

3.2 DEFINIÇÃO DO PROBLEMA

O problema de alocação de recursos descrito no Capítulo 2 pode ser dividido nos subproblemas de controle de potências e alocação de espectro. Com base na formulação anteriormente apresentada, o subproblema de controle das potências das comunicações e dos sensores de um RB pode ser modelado matematicamente como:

$$\max_{p_j, p_q} \sum_{l=1}^{L} \psi_l \tag{3.1}$$

$$\text{s.t. } \psi_a \geq \psi_{\text{min}}, \quad \forall\, a \in \mathbf{A} \tag{3.2}$$

$$P_q^d \geq \phi_{\text{min}}, \quad \forall\, q \in \mathbf{Q} \tag{3.3}$$

$$p_j \leq p_{\text{max}}, \quad \forall\, j \in \mathbf{J} \tag{3.4}$$

$$p_q \leq p_{\text{max}}, \quad \forall\, q \in \mathbf{Q} \tag{3.5}$$

$$\psi_j = \log_2(1 + b_j \zeta_{j,k}), \quad \forall\, j \in \mathbf{J} \tag{3.6}$$

em que:

$$p_j \in \mathbb{R}, \quad \forall\, j \in \mathbf{J} \tag{3.7}$$

$$p_q \in \mathbb{R}, \quad \forall\, q \in \mathbf{Q} \tag{3.8}$$

$$\mathbf{J} = \mathbf{A} \cup \mathbf{L} \tag{3.9}$$

em que b_j e b_q não são variáveis de decisão deste subproblema, e sim parâmetros definidos pelo subproblema de alocação de espectro. Além disso, a dimensão K é eliminada do problema, uma vez que a solução de um RB é independente da solução de outro RB, já que a modulação OFDM é ortogonal.

O problema modelado segue sendo não linear, mas já não se configura como um problema combinatorial, além de não possuir variáveis inteiras. Entretanto, a exigência de baixa latência se intensifica, uma vez que o algoritmo de controle de potências será executado em cada RB separadamente, multiplicando o seu tempo de execução ou exigindo uma paralelização da execução.

Nesse contexto, os algoritmos de aprendizado por reforço, especialmente os algoritmos de aprendizado por reforço profundo, surgem como boas alternativas para resolução do problema.

Um dos motivos para isso é a capacidade que tais algoritmos têm de lidar com problemas que envolvem não linearidades, seja na função objetivo ou em outras partes da definição do problema. Na prática, a maioria dos algoritmos de aprendizado por reforço aprende a decidir ações em ambientes complexos a partir dos resultados observados em um processo

de tentativa e erro [13].

Dessa forma, para a aplicação de tais algoritmos, não são exigidas características lineares ou convexas na formulação do problema, apenas um processo de simulação e coleta dos resultados provenientes dessa simulação para servir no processo de aprendizagem do algoritmo.

Outro motivo é a capacidade que tais algoritmos possuem de encontrar soluções em tempo curto, não demandando longos períodos de tempo para inferência e, consequentemente, não inviabilizando a sua aplicação em contextos que necessitam de ações em tempo real. Os algoritmos de aprendizado por reforço possuem, em geral, processos de treinamento dos algoritmos relativamente demorados, mas o tempo de inferência para decisão de uma ação após o treinamento é curto [11, 13].

Esse fator faz com que esses algoritmos sejam priorizados em contextos complexos envolvendo tempo real em detrimento de algoritmos que fazem busca no espaço de soluções para decisão da ação, como algoritmos genéticos ou algoritmos de programação inteira e programação mista.

Além disso, algoritmos de aprendizado por reforço, principalmente os de aprendizado por reforço profundo, apresentam boa capacidade de generalização, de forma que os algoritmos aprendem padrões existentes no problema e consegue extrapolar o que aprendeu pra situações não vivenciadas [11]. Isso faz com que tais algoritmos sejam escaláveis em contextos complexos e com alta mutabilidade nos cenários que podem aparecer.

3.3 APRENDIZADO POR REFORÇO PROFUNDO

As técnicas de Aprendizado por Reforço Profundo (DRL) são técnicas de Inteligência Artificial (IA) que estão na interseção entre as áreas de Aprendizado Profundo (*Deep Learning* - DL) e Aprendizado por Reforço (*Reinforcement Learning* - RL) [40].

Os algoritmos de DRL integram as técnicas avançadas de DL e RL para criar algoritmos mais sofisticados. Os algoritmos de RL contribuem com flexibilidade, permitindo o aprendizado a partir de recompensas em vez de exigir dados rotulados. Em contrapartida, as técnicas de DL fornecem uma capacidade de generalização robusta, através do uso de múltiplas camadas lineares e não lineares para modelar problemas complexos [11].

3.3.1 Conceitos básicos

O campo de Aprendizado por Reforço (RL) dentro da Inteligência Artificial (IA) foca no desenvolvimento de algoritmos que capacitem máquinas a tomar decisões com base nas

recompensas obtidas após as ações implementadas [41].

A particularidade dos algoritmos de RL é que eles divergem das técnicas convencionais de Aprendizado de Máquina (ML) em sua abordagem de aprendizagem [40]. Isso se deve à complexidade de certos problemas, que podem apresentar dados de difícil rotulação ou uma grande variedade de entradas e saídas possíveis. Nesse cenário, surgem conceitos fundamentais exclusivos à área de RL.

Embora o domínio de RL seja amplo e contenha diversas estratégias e tecnologias, existem elementos fundamentais que são aplicáveis à maioria dos algoritmos: ambiente, estado, agente e recompensa [13].

O ambiente atua como um modelo do problema a ser resolvido, fornecendo as informações necessárias para o algoritmo decidir ações adequadas [42]. O estado, simbolizado por s, representa a configuração instantânea do ambiente. O agente, que opera de acordo com uma política μ, avalia o estado atual para escolher uma ação apropriada. A recompensa r, por sua vez, avalia a qualidade da ação escolhida [43].

O objetivo do agente é maximizar o retorno, que é a soma acumulada de recompensas em uma trajetória τ específica [42]. Uma trajetória é uma sequência de eventos pelos quais o agente passa, ou seja, um conjunto de estados do ambiente e transições a partir das ações decididas por ele. A meta não é apenas maximizar a recompensa imediata, mas a soma total das recompensas ao longo de uma trajetória τ. O retorno é matematicamente representado pela Equação (3.10) [13]:

$$R(\tau) = \sum_{t=0}^{\tau} \gamma^t r_t \tag{3.10}$$

em que t é um *timestep* do problema, que é a medida mínima da granularidade da discretização temporal do problema, γ é o fator de desconto, que prioriza recompensas mais próximas no tempo, isto é, que sejam mais recentes, em detrimento de recompensas mais do início da trajetória.

A trajetória pela qual o agente passa é definida tanto pelas ações decididas baseadas na política do agente, quanto pelas aleatoriedades intrínsecas do ambiente. Desta forma, não é possível definir com certeza em que estado o ambiente estará, conhecendo o estado inicial e a política do agente.

Por esse motivo, é necessário uma modelagem mais complexa dos problemas, considerando um aprendizado que não é baseado na previsão do futuro, mas no aprendizado a partir das experiências já vivenciadas e no entendimento de como uma ação impacta na trajetória vivenciada pelo algoritmo.

3.3.2 Política determinística e estocástica

A política μ pode ser tanto estocástica quanto determinística, dependendo de se ela permite incertezas ou não na escolha de ações. Um algoritmo com política determinística define qual ação deve ser realizada, seja ela de natureza contínua ou discreta. Por outro lado, algoritmos com política estocástica definem uma distribuição estatística da qual a ação definida será uma amostra [13, 12].

3.3.3 Model-free e model-based

Alguns algoritmos de RL, conhecidos como *model-based*, buscam modelar completamente o ambiente. Nesse tipo de algoritmo, o modelo do ambiente criado é usado para a definição das ações [13, 12]. Uma das maneiras mais utilizadas por algoritmos desse tipo é simular diferentes ações no ambiente não real modelado e, a partir dos resultados, definir a ação que vai ser realizada no ambiente real.

No entanto, a maioria dos problemas reais são complexos demais para serem modelados completamente, levando ao uso de algoritmos *model-free*, que aprendem através da experiência direta com o ambiente [13, 43].

Para esses algoritmos, torna-se crucial empregar métodos inteligentes capazes de otimizar o retorno ao longo de múltiplas trajetórias possíveis. Isso nos leva aos conceitos de função valor e função ação-valor.

3.3.4 Função Valor e Função Ação-Valor

A eficácia de um agente de aprendizado por reforço em tomar decisões baseadas em estados depende fortemente de sua capacidade de estimar o retorno esperado. Para alcançar isso, dois conceitos cruciais emergem no campo: a função valor e a função ação-valor.

A função valor, denotada como $V_\mu(s)$, é uma métrica que avalia o quão promissor é para o agente estar em um estado s sob uma política específica μ. Essa métrica é computada pelo valor médio dos retornos esperados quando se começa a partir do estado s e se segue a política μ. Portanto, ela fornece a esperança de retorno recebido pelo agente ao partir de s [13]. Tal função pode ser calculada a partir da Equação (3.11).

$$V_\mu(s) = \mathop{\mathbb{E}}_{\tau\sim\mu}[R_t|s_t = s] \tag{3.11}$$

Entender a função valor é crucial, mas não é suficiente quando se deseja saber o valor de tomar uma ação específica em um estado específico. Por outro lado, a função ação-valor, representada por $Q_\mu(s, a)$, estende a ideia da função valor ao incorporar a ação a tomada no

estado s.

Diferentemente da função valor, que é uma média dos retornos esperados a partir de um estado sob uma política, a função ação-valor fornece uma perspectiva mais granular. Ela informa o retorno esperado para um par estado-ação, permitindo ao agente medir o impacto da sua ação atual [13]. A função ação-valor pode ser calculada usando a Equação (3.12),

$$Q_\mu(s,a) = \mathop{\mathbb{E}}_{\tau\sim\mu}[R_t|s_t = s, a_t = a] \tag{3.12}$$

A relação entre a função valor e a função ação-valor é frequentemente expressa matematicamente, conectando-se as expectativas futuras de retorno entre estados e pares estado-ação. A Equação (3.13) define, matematicamente, tal relação.

$$V_\mu(s) = \mathop{\mathbb{E}}_{a\sim\mu}[Q_\mu(s,a)] \tag{3.13}$$

Como a função ação-valor incorpora a ação realizada no presente para estimar o retorno recebido no futuro, ela diminui, se comparado a função V, a variância dessa estimativa em função da ação decidida atualmente. Essa característica é especialmente útil em ambientes com uma alta dinamicidade ou ambientes em que as ações têm efeitos a longo prazo que são difíceis de calcular apenas observando o estado atual, ou seja, em ambientes em que a decisão atual pode mudar drasticamente o valor da estimativa do retorno a ser obtido.

As informações presentes nas funções valor e ação-valor e a capacidade de estimar seus valores é o núcleo do que torna os algoritmos de RL eficazes em navegar em ambientes complexos para maximizar um objetivo.

Além das funções apresentadas, um conceito adicional crucial no campo do Aprendizado por Reforço é a função vantagem (*advantage*), denotada como $A(s,a)$. Esta função busca quantificar o benefício relativo de escolher uma ação específica a em um dado estado s, em comparação com a média de todas as ações possíveis nesse estado. Formalmente, a função vantagem pode ser definida pela Equação (3.14) [13, 43]:

$$A_\mu(s,a) = Q_\mu(s,a) - V_\mu(s) \tag{3.14}$$

Nesta equação, $Q_\mu(s,a)$ representa o valor esperado do retorno ao se tomar uma ação a no estado s e seguir uma política μ a partir desse ponto. Por outro lado, $V_\mu(s)$ representa o valor esperado do retorno dado que o ambiente está no estado s quando a mesma política μ é seguida, independentemente da primeira ação tomada. A função vantagem, então, serve para isolar o efeito da escolha da ação a ao eliminar o valor base $V_\mu(s)$ do estado s.

A importância da função vantagem reside em sua capacidade de desambiguar o valor de

diferentes ações em um mesmo estado. Isso é especialmente útil em situações que demandam um equilíbrio entre exploração de ações desconhecidas e explotação de ações já conhecidas como benéficas, utilizando o conhecimento adquirido para definir ações e consolidá-lo. Essa necessidade de equilibrar a exploração e a explotação do espaço de ações é conhecida como *exploration-exploitation dilemma* [13].

3.3.5 Policy Gradient

O objetivo do aprendizado por reforço é encontrar uma estratégia de decisão das ações ótimas para o agente maximizar o retorno obtido. Para isso, os algoritmos podem ser categorizados em duas abordagens principais: baseados em valor e baseados em política.

Algoritmos baseados em valor, como o Q-learning [44], focam em aprender uma função parametrizada que estima a função ação-valor para cada ação em um determinado estado. O objetivo é decidir a ação baseada nas estimativas dessa função parametrizada. Em contraste, algoritmos baseados em política, como o REINFORCE [45], focam diretamente na otimização da política, ajustando os parâmetros da política para maximizar o retorno esperado.

Enquanto os métodos baseados em valor decidem as suas ações baseadas na estimativa da função ação-valor, os métodos baseados em política buscam otimizar a política diretamente.

Métodos baseados em políticas são mais adequados para ambientes com estado e/ou ação de natureza contínua. Isso ocorre porque há um número infinito de ações e estados para estimar os valores, tornando as abordagens baseadas em valor computacionalmente caras nesses casos, o que pode ser considerado intuitivo [45].

Para algoritmos baseados em política, a política é modelada como uma função parametrizada em relação a θ, denotada por $\mu_\theta(s)$. O parâmetro θ não possui significado físico, é apenas um parâmetro que pode ser otimizado para alcançar uma função que faça um mapeamento de um estado para uma ação que maximize o valor da função de custo, que representa a recompensa. Diversos algoritmos podem ser aplicados para otimizar θ, incluindo os algoritmos clássicos baseados em Gradiente Descendente [44].

Derivando analiticamente a partir da Equação (3.13), a função de custo pode ser expressa pela Equação (3.15) [12]:

$$J(\theta) = \sum_{s \in \mathcal{S}} d_\mu(s) V_\mu(s) = \sum_{s \in \mathcal{S}} d_\mu(s) \sum_{a \in \mathcal{A}} \mu_\theta(s) Q_\mu(s, a) \tag{3.15}$$

em que $d_\mu(s)$ é a distribuição estacionária da cadeia de Markov para μ_θ.

Usando o gradiente ascendente, é possível mover θ na direção sugerida pelo gradiente $\nabla_\theta J(\theta)$ para encontrar o melhor θ para μ_θ que produza o maior retorno.

A computação do gradiente $\nabla_\theta J(\theta)$ é desafiadora porque depende tanto da seleção da ação determinada por μ_θ quanto de fatores aleatórios do ambiente. Dado que o ambiente é geralmente desconhecido, é difícil estimar o efeito na distribuição de estados de uma atualização de política.

O teorema de Policy Gradient oferece uma reformulação da derivada da função objetivo para não envolver a derivada da distribuição estacionária de estados $d_\mu(.)$ e simplificar o cálculo do gradiente $\nabla_\theta J(\theta)$. O teorema está matematicamente definido na Equação (3.16) [45].

$$\nabla_\theta J(\theta) = \mathbb{E}_\mu[Q_\mu(s,a)\nabla_\theta \ln \mu_\theta(s)] \tag{3.16}$$

Esse teorema simplifica o problema de otimização de algoritmos de aprendizado por reforço, tornando os métodos de Policy Gradient uma abordagem poderosa para domínios que enfrentam a maldição da dimensionalidade [45].

3.3.6 Redes Neurais Profundas em Aprendizado por Reforço

No contexto de aprendizado por reforço e Policy Gradient, as funções parametrizadas por θ, mencionadas na seção anterior, que definem a política, podem ser implementadas por uma rede neural, ou mais especificamente, por redes neurais profundas. Nesse sentido, as redes neurais podem ser usadas para aproximação de uma função, já que os mecanismos de aprendizagem dessas técnicas permitem que elas aproximem funções extremamente não lineares e de alta complexidade, aprendendo apenas com as experiências do algoritmo [11].

Uma rede neural é um modelo computacional que busca imitar o funcionamento do cérebro humano para realizar uma variedade de tarefas, desde reconhecimento de padrões até tomada de decisões. A rede é composta de camadas de nós, ou neurônios, conectados por sinapses com pesos que associam os nós [11].

Os pesos da rede neural são parâmetros da rede cujos valores são ajustados durante o processo de treinamento. O poder da rede neural reside em sua capacidade de aprender representações complexas e não lineares a partir dos dados, otimizando esses parâmetros para minimizar uma função de custo.

Redes neurais profundas são essencialmente redes neurais com duas ou mais camadas ocultas. Essas redes são conhecidas por sua capacidade de aprender representações ainda mais complexas dos dados, podendo capturar abstrações de níveis múltiplos [11].

É possível combinar redes neurais profundas com técnicas de aprendizado por reforço, como Policy Gradient, ao se utilizarem dos parâmetros ajustáveis da rede neural para representar o θ. Assim, basicamente a política adotada pelo agente seria a própria rede neural,

que aprenderia por meio de $\nabla_\theta J(\theta)$ [12].

Utilizar uma rede neural profunda como a representação de θ permite que o agente aprenda políticas ótimas em ambientes complexos, não lineares e de alta dimensionalidade. A rede neural profunda é capaz de extrair características relevantes do ambiente e da política de forma autônoma, possibilitando que o agente aprenda a decidir ações mesmo em problemas complexos.

3.3.7 Algoritmos relevantes

No campo do aprendizado por reforço profundo, diversos algoritmos têm sido desenvolvidos para abordar uma ampla gama de problemas. Esses algoritmos podem ser classificados de diferentes maneiras.

Uma das classificações mais fundamentais é entre algoritmos *on-policy* e *off-policy*. Algoritmos *on-policy*, como o REINFORCE, atualizam a política que está sendo otimizada com base nos dados coletados por essa mesma política. Em contraste, algoritmos *off-policy*, como o DDPG (*Deep Deterministic Policy Gradient*) e o TD3 (*Time Delayed DDPG*), coletam dados através de uma política que pode estar em um estágio diferente da política que está sendo otimizada [13].

Outra categorização é se o algoritmo possui ou não estrutura *Actor-Critic*. O método *Actor-Critic* básico, por exemplo, utiliza um modelo de política (o agente) e um modelo de função de valor (o crítico) para atualizar os parâmetros do agente. Algoritmos como o DDPG e o PPO (*Proximal Policy Optimization*) são *actor-critic*, enquanto algoritmos como o REINFORCE e o DQN (*Deep Q-Network*) não possuem estrutura *Actor-Critic*.

Neste trabalho, foram implementados quatro algoritmos de aprendizado por reforço profundo: o REINFORCE, o PPO, o DDPG e o TD3. O objetivo é testar diferentes tipos de algoritmos no problema, incluindo algoritmos de política estocástica e determinística, algoritmos *on-policy* e *off-policy*, além de algoritmos de estrutura *actor-critic* e outros que não se baseiam nessa estrutura.

3.3.7.1 REINFORCE

O REINFORCE é um algoritmo *on-policy* de política estocástica que utiliza uma estimativa de Monte Carlo para calcular o retorno esperado e atualizar a política a partir da Equação (3.16) [46].

O REINFORCE foi um dos primeiros algoritmos de DRL baseados em política, é relativamente simples e de fácil implementação, se comparado com os demais algoritmos da área.

O REINFORCE apresenta problemas como a alta variância nas estimativas de gradiente devido à sua dependência da amostragem de trajetórias completas. Isso torna o processo de aprendizagem instável e ineficiente, especialmente em ambientes com um espaço de ações grande [46].

Existem algumas variações que tentam mitigar esse problema, buscando reduzir esta alta variância a partir de uma modificação da função que se quer estimar. Entretanto, mesmo tais variações possuem desempenho prejudicado em ambientes mais complexos [45].

Um diagrama ilustrando o seu processo de treinamento de maneira simplificada está disponível na Figura 3.1 e o seu pseudocódigo está disponível no Algoritmo 1.

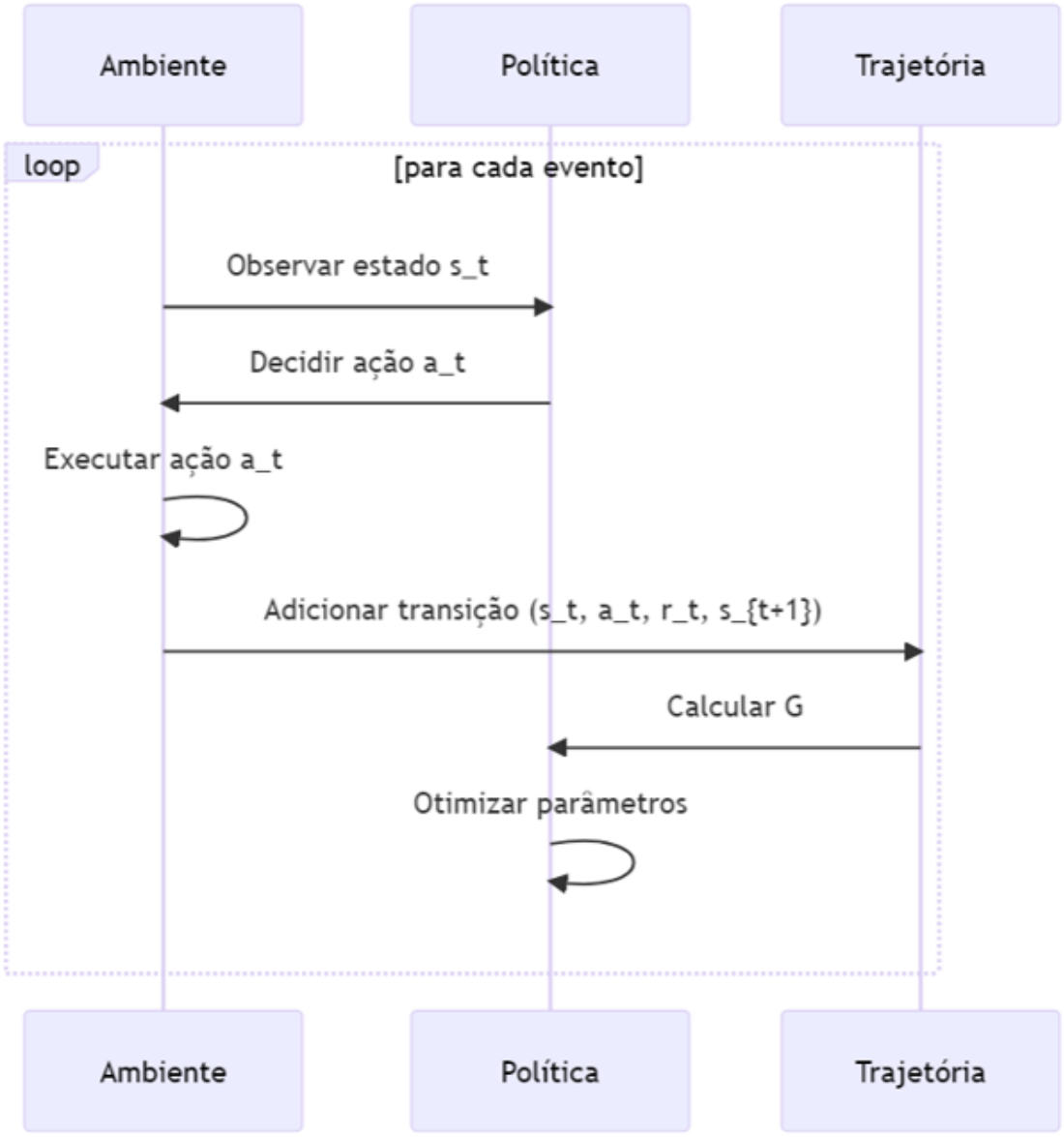


Figura 3.1 – Diagrama do processo de treinamento do REINFORCE. Fonte: autoria própria.

A complexidade computacional do processo de treinamento do REINFORCE pode ser

Algorithm 1 REINFORCE

Entrada: parâmetros θ da política $\mu(a|s,\theta)$
Saída: parâmetros θ da política otimizados
para cada $episodio = 1, 2, \ldots, n_{episodios}$ faça:
    Gerar uma trajetória $S_0, A_0, R_1, \ldots, s_{T-1}, a_{T-1}, R_T$, seguindo $\mu(\cdot|\cdot,\theta)$
    para cada evento $t = 0, 1, 2, \ldots, T$**:**
        $G \leftarrow \sum_{i=t}^{T} R_i \approx Q_\mu(s,a)$
        $\theta \leftarrow \theta + \alpha G \nabla \log \pi(A_t|S_t,\theta)$
    fim para
fim para

estimada por $O(n_{episodios} * T * \sum_{e=1}^{E-1} u_e u_{e-1})$ em que $n_{episodios}$ é o número de episódios do processo de treinamento, T é o número de eventos por episódio, E é o número de camadas da rede neural utilizada como agente e u_e é o número de neurônios da camada e da rede neural [47, 48]. Já a complexidade computacional para o agente realizar uma inferência, ou seja, para a decisão de uma ação, é de $O(\sum_{e=1}^{E-1} u_e u_{e-1})$ [47].

3.3.7.2 Proximal Policy Optimization

O PPO é uma abordagem *off-policy* de estrutura *Actor-Critic* com política estocástica que busca otimizar a política do agente de forma mais eficiente, em comparação com outros métodos *on-policy* a partir de modificações em sua função de custo [49].

O PPO introduz uma função de custo clipada (J^{CLIP}) para evitar atualizações excessivamente grandes ou pequenas na política, mantendo a otimização mais estável. Matematicamente, essa função de custo é dada pela Equação (3.17) [49]:

$$J^{\text{CLIP}}(\theta) = \mathbb{E}[\min(r(\theta)\hat{A}_{\theta_{\text{old}}}, \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta_{\text{old}}})] \tag{3.17}$$

em que $r(\theta)$ é a razão entre a política atual e políticas antigas, $\hat{A}_{\theta_{\text{old}}}$ é a função vantagem estimada usando políticas antigas e ϵ é um hiperparâmetro que define o tamanho limite de atualização da política [49].

Além da função de custo clipada, a função objetivo é aumentada com um termo de erro na estimativa de valor e um termo de entropia para incentivar a exploração do ambiente [45]. Matematicamente, a função objetivo do PPO é dada pela Equação (3.18):

$$J^{\text{CLIP'}}(\theta) = \mathbb{E}\left[J^{\text{CLIP}}(\theta) - c_1(V_\theta(s) - V_{\text{target}})^2 + c_2 H(s, \mu_\theta(.))\right] \tag{3.18}$$

em que c_1 e c_2 são dois hiperparâmetros de priorização, $V_\theta(s)$ é a função valor estimada pelo crítico, V_{target} é a função valor estimada por amostragem de Monte Carlo e $H(s, \mu_\theta(.))$ é a

entropia da política de decisão das ações.

Um diagrama ilustrando o processo de treinamento do PPO está disponível na Figura 3.2 e o pseudocódigo está descrito no Algoritmo 2.

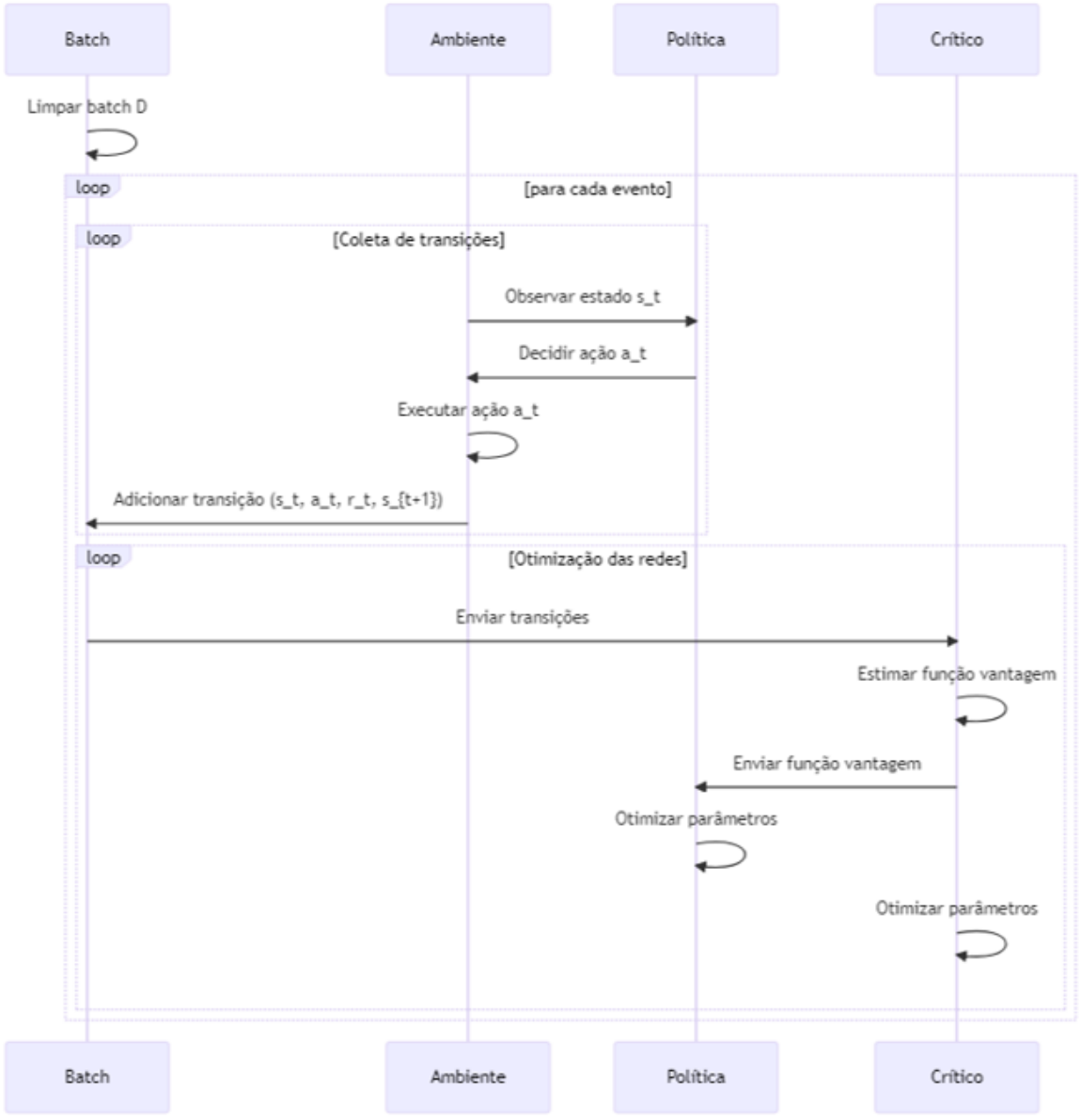


Figura 3.2 – Diagrama do processo de treinamento do PPO.Fonte: autoria própria.

Como o PPO utiliza duas redes neurais no processo de treinamento. a complexidade computacional desta etapa pode ser calculada por $O(n_{episodios} * T * 2 \sum_{e=1}^{E-1} u_e u_{e-1})$ em que $n_{episodios}$ é o número de episódios do processo de treinamento, T é o número de eventos por episódio, E é o número de camadas da rede neural utilizada como agente e u_e é o número de neurônios da camada e da rede neural [47, 48]. Como a inferência depende apenas da rede neural do agente e não da do crítico, a complexidade computacional para se definir uma ação é de $O(\sum_{e=1}^{E-1} u_e u_{e-1})$ [47].

Algorithm 2 PPO

Entrada: parâmetros iniciais da rede de política θ, rede do crítico ϕ e batch de treinamento D
Saída: parâmetros da rede de política e rede do crítico otimizados V_ϕ, μ_θ
para cada episódio $i = 1, 2, \ldots, n_{episodios}$ faça:
    Limpar o batch de treinamento D
    para cada evento $t = 1, 2, \ldots, T$ faça:
        Observar o estado do ambiente S_t
        Selecionar ação a_t de acordo com a política atual $\mu_\theta(a_t|s_t)$
        Executar ação a_t, obter recompensa r_t e transitar para o próximo estado s_{t+1}
        Adicionar as experiências (s_t, a_t, r_t, s_{t+1}) ao batch de treinamento D
    fim para
    para cada iteração de treinamento $k = 1, 2, \ldots, K$ faça:
        Estimar a função vantagem $\hat{A}_{\theta_{\text{old}}}$ baseado na estimativa $V_{\phi_k}(s)$ feita pelo crítico
        Atualizar a política usando gradiente ascendente de forma que:

$$\theta_{k+1} \leftarrow \arg\max_{\theta} \frac{1}{|D|T} \sum_{\tau \in D} \sum_{t=0}^{T} \min \left(\frac{\mu(a|s)}{\mu_{\text{old}}(a|s)} \hat{A}_{\theta_{\text{old}}}, g(\epsilon, \hat{A}_{\theta_{\text{old}}}) + H(s,a) \right)$$

        Ajustar os parâmetros do crítico por regressão no erro quadrático médio:

$$\phi_{k+1} \leftarrow \arg\min_{\phi} \left(\sum_{t=0}^{T} (R_t - V_{\phi_k})^2 - H(s,a) \right)$$

    fim para
fim para

3.3.7.3 DDPG

Deep Deterministic Policy Gradients (DDPG) é um algoritmo *off-policy* que estende as capacidades do DQN para espaços de ação contínuos. Enquanto o DQN foi projetado para espaços discretos de ação e estabiliza o aprendizado da função ação-valor através do *Experience-Replay* e do uso de redes *target*, o DDPG adapta esses princípios para espaços contínuos de ação usando uma estrutura *Actor-Critic* [50].

Ele utiliza uma rede neural para definir as ações e outra para estimar a função ação-valor. Especificamente, o DDPG aprende uma política determinística, em oposição às políticas estocásticas comumente usadas em outros métodos [50].

Uma das características-chave do DDPG é sua abordagem para atualizar os parâmetros das redes do ator e do crítico. Diferentemente do DQN, onde a rede *target* permanece congelada por um determinado período, o DDPG emprega atualizações suaves das redes *target*. Neste método, os parâmetros da rede alvo são atualizados de forma mais gradual, de acordo com a fórmula $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$, onde $\tau \ll 1$. Isso garante que os valores da rede alvo mudem lentamente, proporcionando um ambiente de aprendizado mais estável [44, 50].

O diagrama com a representação pictórica do treinamento do DDPG está disponível na Figura 3.3 e o pseudocódigo está no Algoritmo 3.

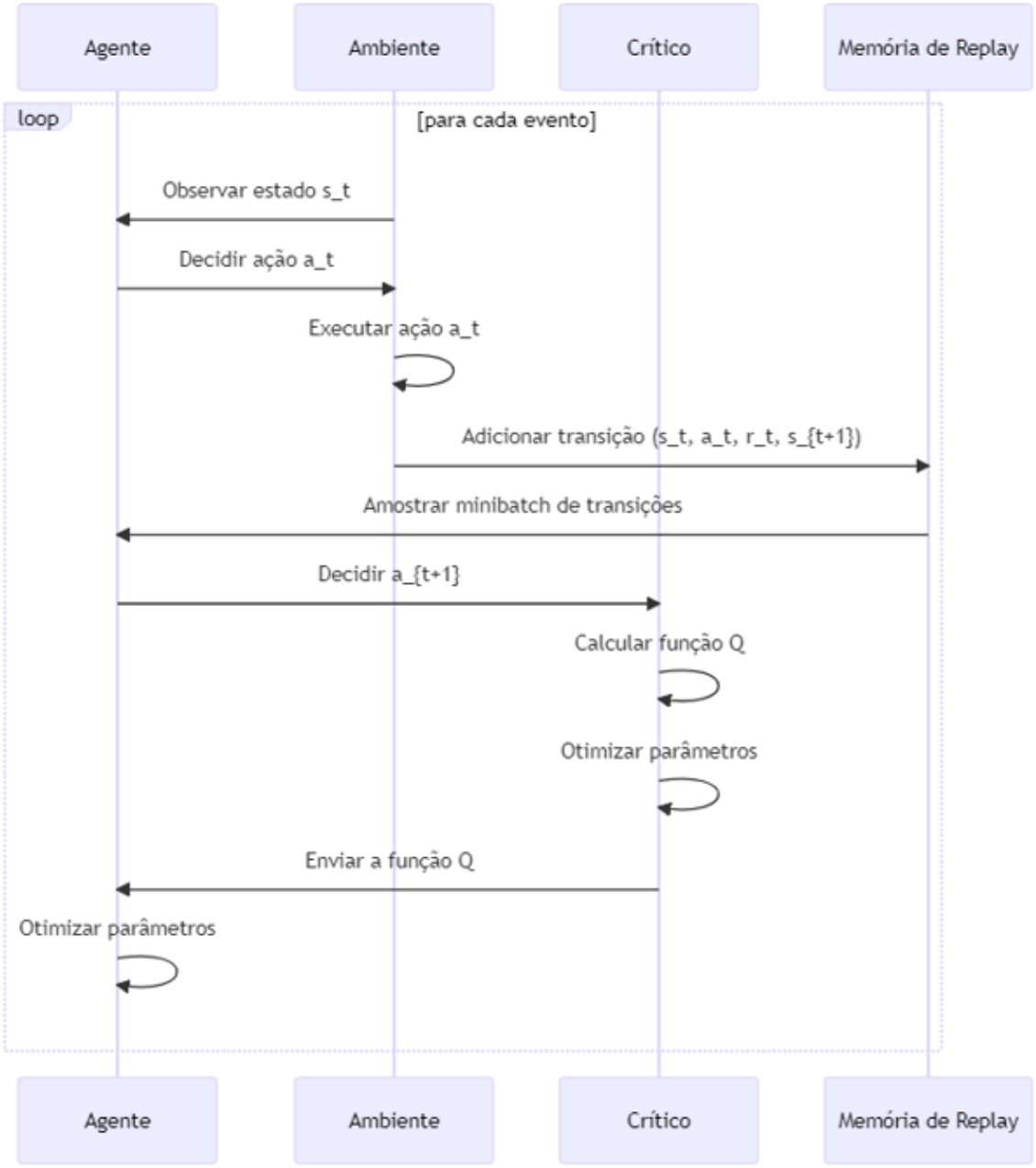


Figura 3.3 – Diagrama do processo de treinamento do DDPG e do TD3.Fonte: autoria própria.

As complexidades computacionais do DDPG e do TD3 são iguais à do PPO, tanto para treinamento quanto para inferência, podendo ser calculadas por $O(n_{episodios} * T * 2 \sum_{e=1}^{E-1} u_e u_{e-1})$ e $O(\sum_{e=1}^{E-1} u_e u_{e-1})$, respectivamente, em que $n_{episodios}$ é o número de episódios do processo de treinamento, T é o número de eventos por episódio, E é o número de camadas da rede neural utilizada como agente e u_e é o número de neurônios da camada e da rede neural [47, 48].

Algorithm 3 DDPG

Entrada: parâmetros iniciais das redes neurais do agente θ_μ e do crítico θ_Q e da memória de *replay* R
Saída: parâmetros otimizados das redes neurais do agente θ_μ
Inicializar redes neurais do agente $\mu(s;\theta_\mu)$ e do crítico $Q(s,a;\theta_Q)$
Inicializar redes neurais *target* $\mu'(s;\theta'_\mu)$ e $Q'(s,a;\theta'_Q)$
Sincronizar os pesos das redes *target*: $\theta'_\mu \leftarrow \theta_\mu$, $\theta'_Q \leftarrow \theta_Q$
Inicializar a memória de *replay* R
para cada episódio $i = 1, 2, \ldots, n_{episodios}$ faça:
    Inicializar a estratégia de exploração η
    Observar o estado inicial s_0
    para cada evento $t = 1, 2, \ldots, T$ faça:
        Selecionar a ação a_t através de $\mu(s_t;\theta_\mu)$ e η
        Executar a ação a_t, obter recompensa r_t e transitar para o próximo estado s_{t+1}
        Adicionar a transição (s_t, a_t, r_t, s_{t+1}) à memória R
        Amostrar um *minibatch* de N transições (s_j, a_j, r_j, s_{j+1}) de R
        Calcular $h_j = r_j + \gamma Q'(s_{j+1}, \mu'(s_{j+1};\theta'_\mu))$
        Atualizar θ_Q minimizando:

$$Loss_Q = \frac{1}{N}\sum_j \left(h_j - Q(s_j, a_j;\theta_Q)\right)^2$$

        Atualizar θ_μ através do gradiente ascendente:

$$Loss_\mu = \frac{1}{N}\sum_j Q(s_j, \mu(s_j;\theta_\mu);\theta_Q)$$

        Atualizar suavemente os parâmetros das redes *target*:

$$\theta'_\mu \leftarrow \tau\theta_\mu + (1-\tau)\theta'_\mu$$

$$\theta'_Q \leftarrow \tau\theta_Q + (1-\tau)\theta'_Q$$

    fim para
fim para

3.3.7.4 TD3

O TD3 é uma extensão do algoritmo DDPG, projetada para abordar o viés de sobreestimação que é comum em métodos *off-policy*. Com essa finalidade, o TD3 incorpora algumas modificações significativas em relação ao seu predecessor [51].

Primeiramente, adota uma abordagem de *Clipped Double Q-learning*, que utiliza duas redes críticas para minimizar o viés de sobreestimação na função de valor. Essa técnica favorece um viés de subestimação, minimizando a propagação de erros durante o processo de treinamento [51, 45].

Além disso, o TD3 introduz um atraso na atualização da política que é especialmente útil para estabilizar o processo de convergência durante treinamento. Diferentemente do DQN, onde a rede *target* é atualizada periodicamente, o TD3 atualiza a política com uma frequência menor do que atualiza o crítico, responsável por estimar a função ação-valor. Isso reduz a variância e torna o aprendizado mais robusto [51, 45].

Por último, o algoritmo implementa uma estratégia de suavização da política *target* para evitar *overfitting* a ações em que existe uma sobrestimação da função ação-valor. Essa suavização é realizada adicionando uma pequena quantidade de ruído aleatório às ações selecionadas para estimação da função Q [51].

Pela semelhança na estrutura, o diagrama do treinamento do TD3 não é diferente do diagrama do DDPG que está exposto na Figura 3.3. As principais diferenças aparecem em pontos mais específicos e estão detalhadas no pseudocódigo disponível no Algoritmo 4.

Algorithm 4 TD3

Entrada: parâmetros iniciais das redes neurais do agente θ_μ, dos críticos $\theta_{Q_1}, \theta_{Q_2}$ e memória de *replay* R
Saída: parâmetros otimizados da rede neural do agente θ_μ
Inicializar redes neurais do agente $\mu(s;\theta_\mu)$ e dos críticos $Q_1(s,a;\theta_{Q_1})$ e $Q_2(s,a;\theta_{Q_2})$
Inicializar redes neurais *target* $\mu'(s;\theta'_\mu)$, $Q'_1(s,a;\theta'_{Q_1})$ e $Q'_2(s,a;\theta'_{Q_2})$
Sincronizar os pesos das redes *target*: $\theta'_\mu \leftarrow \theta_\mu$, $\theta'_{Q_1} \leftarrow \theta_{Q_1}$ e $\theta'_{Q_2} \leftarrow \theta_{Q_2}$
Inicializar a memória de *replay* R
para cada episódio $i = 1, 2, \ldots, n_{episodios}$ faça:
    Inicializar estratégia de exploração η
    Observar estado inicial s_0
    para cada evento $t = 1, 2, \ldots, T$ faça:
        Selecionar ação a_t usando a estratégia de exploração η e a política $\mu_{\theta_\mu}(a_t|s_t)$
        Executar ação a_t, obter recompensa r_t e transitar para o próximo estado s_{t+1}
        Armazenar a transição (s_t, a_t, r_t, s_{t+1}) na memória R
        Amostrar um *minibatch* com N transições (s_j, a_j, r_j, s_{j+1}) de R
        Definir ação *target* $a' \leftarrow \mu'(s_j) + \text{clip}(\mathcal{N}(0, \sigma_\mu), -c, c)$
        Calcular a Equação de Bellman: $h_j = r_j + \gamma \min_{k=1,2} Q'_k(s_{j+1}, a')$
        Atualizar $\theta_{Q_1}, \theta_{Q_2}$ minimizando o erro quadrático médio:

$$\text{Loss}_{Q_k} = \frac{1}{N} \sum_j [h_j - Q_k(s_j, a_j)]^2$$

        se t **mod** $T_{\textbf{update}} = 0$ então:
            Atualizar θ_μ através do gradiente ascendente:

$$Loss_\mu = \frac{1}{N} \sum_j Q(s_j, \mu(s_j;\theta_\mu);\theta_Q)$$

            Atualizar suavemente os parâmetros das redes *target*:

$$\theta'_\mu \leftarrow \tau\theta_\mu + (1-\tau)\theta'_\mu$$

$$\theta'_Q \leftarrow \tau\theta_Q + (1-\tau)\theta'_Q$$

        fim se
    fim para
fim para

3.4 IMPLEMENTAÇÃO

Para que os algoritmos de DRL detalhados na última seção sejam implementados, é necessário desenvolver um ambiente capaz de modelar o problema real, adaptando-o aos conceitos de RL apresentados. Dessa maneira, é necessário definir qual será a representação dos estados do ambiente, qual será a função recompensa e como as ações definidas serão introduzidas no ambiente, além de outras configurações do processo de aprendizagem dos algoritmos.

Antes disso, entretanto, é necessário delimitar o escopo do controle de potência. Nos RBs de um sistema, o número de comunicações e sensores varia ao longo do tempo, além de variar de um RB para o outro. Isso faz com que o ambiente apresente um número variável de ações e, consequentemente, de informações a serem passadas ao agente para decisão do controle de potências.

Essa natureza complexifica a implementação de algoritmos de DRL, uma vez que a maioria das redes neurais possui um número fixo de entradas e saídas. Algumas estruturas são mais flexíveis nesse aspecto, porém possuem uma arquitetura mais complexa e um processo de convergência menos estável [52].

Por esse motivo, decidiu-se utilizar redes neurais profundas simples (*Deep Neural Networks* - DNNs), constituídas apenas por neurônios e funções de ativação, mas, para contornar o problema do número variável de entradas e saídas da rede, fixou-se o número máximo de comunicações e sensores por RB cujas potências serão controladas pelo algoritmo.

Dessa forma, considerando que cada RB é sempre preenchido com uma comunicação primária, em cada *timestep* seleciona-se essa comunicação primária e escolhem-se aleatoriamente cinco (5) comunicações D2D e quatro (4) sensores, cujas potências serão definidas pelo algoritmo proposto. Existem dois casos que requerem atenção:

- Em caso de haver menos comunicações ou sensores do que o máximo permitido, são passadas algumas entradas mascaradas ao algoritmo para representar essa sobra de capacidade e as saídas do mesmo são ajustadas apenas para as comunicações existentes;
- Caso contrário, ou seja, se houver mais comunicações ou sensores do que o máximo permitido, aqueles que não forem sorteados permanecem com a mesma potência que utilizaram no *timestep* anterior.

A Figura 3.4 ilustra esse processo por meio de um diagrama.

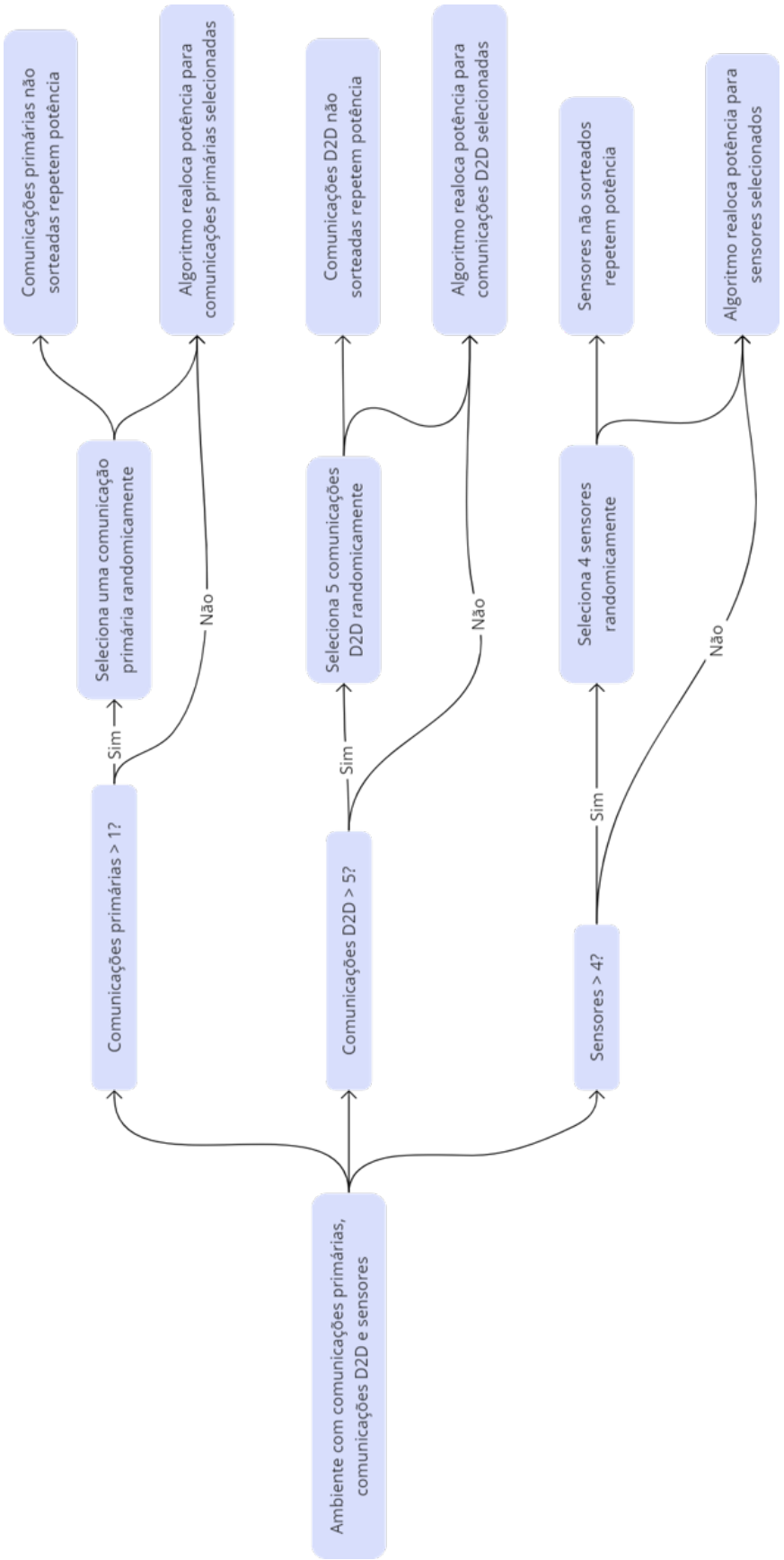


Figura 3.4 – Diagrama do processo de decisão de quais comunicações e sensores serão selecionados para terem suas potências realocadas pelo algoritmo. Fonte: autoria própria.

Além de controlar a dimensionalidade variável do problema, o processo apresentado evita um aumento rápido das dimensões do ambiente, um fator relevante para o desempenho dos algoritmos de DRL implementados [14].

Adicionalmente, a natureza dinâmica do problema assegura que a seleção aleatória não prejudique o processo, já que ainda que uma comunicação ou sensor não tenha potência realocada no *timestep* atual, provavelmente ele será sorteado em *timesteps* posteriores. Assim, desde que o número de comunicações e/ou sensores no RB não seja muito maior do que o número máximo, nenhum deles deve ficar muito tempo sem ter a potência ajustada.

3.4.1 Estado do ambiente

Com base na estratégia adotada para minimizar a variabilidade da dimensionalidade do problema, o estado selecionado para representar o ambiente é um vetor composto por 19 elementos. Considerando as comunicações D2D e os sensores selecionados pelo processo explicado anteriormente, o vetor é composto por conjuntos de elementos em que:

- O 1º elemento é o ganho de propagação do canal da comunicação primária;
- Entre o 2º e o 6º elemento, está o ganho de propagação do canal entre o transmissor de cada comunicação D2D e a ERB;
- Entre o 7º e o 11º elemento, está o somatório dos ganhos de propagação do canal entre o transmissor de cada comunicação D2D com o receptor de todos os sensores;
- Entre o 12º e o 15º elemento, está o ganho de propagação do canal de cada sensor;
- Entre o 16º e o 19º elemento, está o ganho de propagação do canal entre cada transmissor de um sensor e a ERB.

O primeiro elemento é utilizado pelo algoritmo para compreender a situação da comunicação primária; O segundo conjunto de elementos, informa ao algoritmo o nível de interferência que cada comunicação D2D gera na comunicação primária; o terceiro conjunto é utilizado para que o algoritmo entenda o nível de interferência que cada comunicação D2D gera nos sensores; o quarto conjunto permite que o algoritmo entenda a situação do canal de cada sensor; e, por fim, o quinto conjunto é usado para que o algoritmo entenda o nível de interferência que cada sensor gera na comunicação primária.

Em casos em que o RB não está preenchido com o número máximo de comunicações D2D ou de sensores, os elementos referentes a tais vagas serão preenchidos com "-1". O valor foi escolhido porque nenhum ganho de propagação tem valor negativo, de forma que tais elementos serão os únicos valores não positivos do vetor de entrada, facilitando a distinção pelo algoritmo.

3.4.2 Ação

A ação esperada por *timestep* é um vetor de dimensão igual a 10, em que cada elemento representa a potência a ser alocada para cada comunicação ou sensor selecionado.

Quando o RB não contém o número máximo de comunicações D2D ou de sensores, os elementos correspondentes no vetor de ação, que se referem às comunicações ou sensores ausentes, não são utilizados, implementando no sistema apenas as potências das comunicações e sensores selecionados.

3.4.3 Recompensa

A função de recompensa utilizada foi desenvolvida empiricamente, partindo inicialmente de uma equação idealizada para premiar o comportamento esperado e penalizar o comportamento não desejado do agente. Utilizando-se essa função de recompensa inicial, foram desenvolvidos alguns modelos cujos resultados foram avaliados. A partir da análise dos resultados obtidos pelos agentes treinados, a equação função foi continuamente ajustada até alcançar resultados que equilibrassem os múltiplos objetivos e evidenciassem a convergência dos algoritmos de DRL.

A função de recompensa alcançada ao final desse processo é composta por três fatores principais:

- O primeiro fator, r^{primaria}, é a recompensa referente à proteção das comunicações primárias, recompensando casos em que a comunicação não está em *outage*, isto é, em que a eficiência espectral da comunicação primária é superior à eficiência espectral mínima para essas comunicações (ψ_{min}).
- O segundo fator, r^{sensor}, relaciona-se à proteção dos sensores, atribuindo maior recompensa quando a probabilidade de detecção do sensor se mantém acima da probabilidade mínima de detecção (ϕ_{min}).;
- O terceiro fator, r^{D2D}, é a recompensa referente aos D2Ds, recompensando o algoritmo à medida em que a eficiência espectral dessas comunicações aumenta.

Assim, a recompensa total do ambiente pode ser calculada por:

$$r_{PC} = \omega_1 r_a^{\text{primaria}} + \omega_2 \sum_{q=1}^{Q} r_q^{\text{sensor}} + \omega_3 \sum_{l=1}^{L} r_l^{\text{D2D}} \tag{3.19}$$

em que ω_1, ω_2 e ω_3 são os pesos que ponderam cada fator da função de recompensa.

Conforme apresentado no Capítulo 2, a eficiência espectral das comunicações e a probabilidade de detecção dos sensores podem ser formuladas em função da SNIR dos respectivos canais. Assim, todas as componentes da função de recompensa são expressas em termos das SNIRs das comunicações e dos sensores. Assim, cada uma dessas parcelas foi definida conforme as Equações (3.20), (3.21) e (3.22).

$$r_a^{\text{primaria}} = \begin{cases} 5 - 0.1\zeta_a & \text{se } \zeta_a > \zeta_{\text{min}}^A \\ -\zeta_{\text{min}}^A + 0.5\zeta_a & \text{caso contrário} \end{cases} \tag{3.20}$$

$$r_q^{\text{sensor}} = \begin{cases} 2.5 - 0.05\zeta_q & \text{se } \zeta_q > \zeta_{\text{min}}^Q \\ -0.05\zeta_{\text{min}}^Q + 0.5\zeta_q & \text{caso contrário} \end{cases} \tag{3.21}$$

$$r_l^{\text{D2D}} = \zeta_l \tag{3.22}$$

em que $\zeta_{[\cdot]}$ é a SNIR da comunicação ou sensor, ζ_{min}^A é a SNIR mínima para garantia da eficiência espectral mínima das comunicações primárias do sistema e ζ_{min}^Q é a SNIR mínima para garantia de uma probabilidade de detecção dos sensores maior do que a probabilidade de detecção mínima ϕ_{min} do sistema.

No contexto do problema abordado, o sistema foi modelado de modo que as comunicações primárias atinjam uma eficiência espectral mínima de ψ_{min} igual a 2.6 bps/Hz. Isso implica uma SNIR mínima de cerca de $\zeta_{\text{min}}^A = 5$ dB.

Os sensores, por sua vez, foram projetados para alcançar uma probabilidade de detecção superior a $\phi_{\text{min}} = 0.99$. Levando em conta a recepção de $w = 512$ pulsos por segundo e um limiar de detecção $\lambda = 0.001$, eles devem assegurar uma SNIR mínima de $\zeta_{\text{min}}^Q = -10$ dB.

Finalmente, os pesos atribuídos a cada fator da função de recompensa foram definidos como $\omega_1 = 2$, $\omega_2 = 1.5$ e $\omega_3 = 0.1$. Essa escolha visa priorizar os aspectos relacionados às comunicações primárias e aos sensores, enquanto ainda se mantém um incentivo para a otimização da recompensa das comunicações D2D.

A Figura 3.5 mostra o gráfico de cada um dos fatores que compõem a função recompensa em função da SNIR das respectivas comunicações ou sensores.

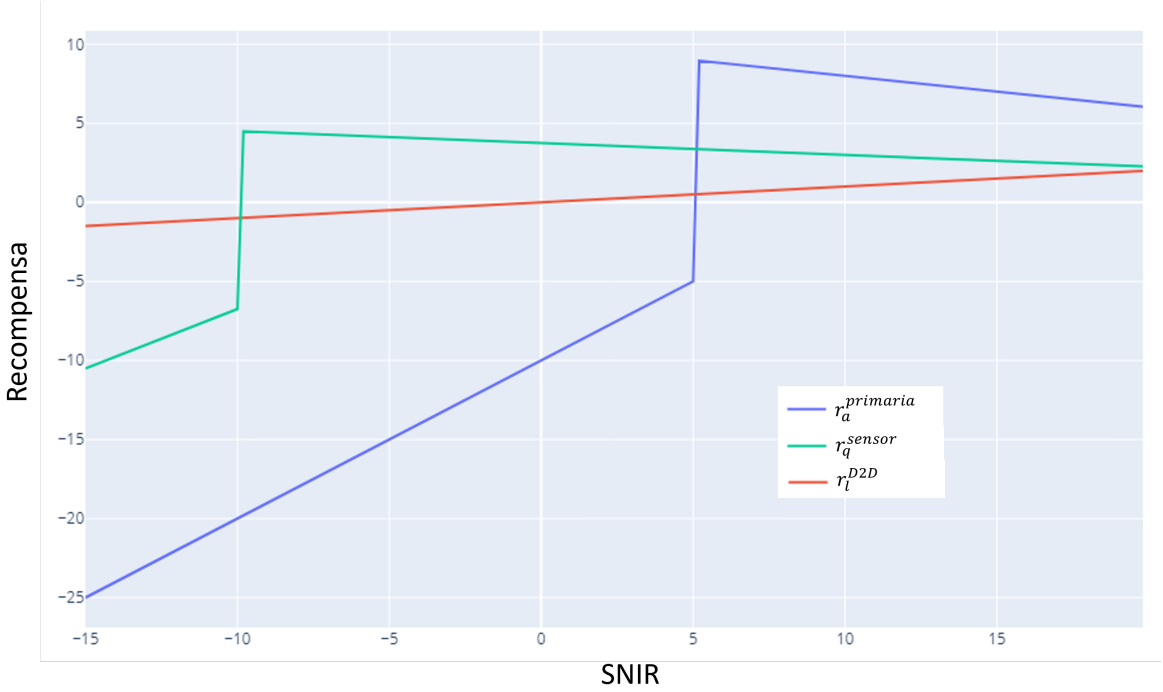


Figura 3.5 – Gráfico de cada um dos fatores que compõem a função de recompensa modelada. Fonte: autoria própria.

É possível identificar que as não linearidades presentes nas curvas referentes a r^{primaria} e a r^{sensor} na Figura 3.5 coincidem com os valores de ζ^{A}_{min} e ζ^{Q}_{min}. Tais não linearidades são utilizadas para penalizar fortemente os casos em que tais limites não são respeitados. Além disso, é perceptível que os valores de r^{primaria} e de r^{sensor} diminuem à medida que a SNIR se distanciam de ζ^{A}_{min} e ζ^{Q}_{min}, respectivamente.

O comportamento dessas funções quando a SNIR é menor do que a SNIR mínima evita que o algoritmo utilize muito tempo do processo de treinamento para encontrar a direção correta de atuação para minimizar a taxa de *outage*. Assim, o algoritmo é gradativamente mais recompensado à medida que se aproxima de ações que mantenham a SNIR das comunicações e sensores mais próximas da SNIR mínima. Por outro lado, o comportamento decrescente das curvas após a não linearidade induz o algoritmo a não desperdiçar recursos do sistema para manter a SNIR em valores maiores do que o necessário, uma vez que a SNIR mínima já está sendo atendida.

A Tabela 3.1 reúne todas as parametrizações definidas e detalhadas nessa seção, além de outros parâmetros detalhados no Capítulo 2, que são referentes ao sistema de comunicação.

3.4.4 Execução das simulações

A fim de assegurar a adaptação do algoritmo às diversas configurações do sistema, o número de comunicações D2D e de sensores variava a cada episódio das simulações, po-

Tabela 3.1 – Parâmetros para a implementação do ambiente

Parâmetro	Valor
ψ_{min}	2.6 bps/Hz
ζ^{A}_{min}	5 dB
ϕ_{min}	0.99
w	512 pulsos/s
λ	0.001
ζ^{Q}_{min}	-10 dB
ω_1	2
ω_2	1.5
ω_3	0.1
f_c	28 GHz
G_t^{sensor}	32
G_t^{D2D}	0
G_r^{BS}	32
G_r^{sensor}	32
G_r^{D2D}	0
SF	$\mathcal{N}(0, 4.2)$
σ_{RCS}	1
p_{max}	50 dBm

rém permanecia constante ao longo de um mesmo episódio. O número de comunicações primárias não variava e era sempre igual a 1.

O número de comunicações D2D e de sensores em cada episódio foi determinado seguindo uma distribuição de Poisson, porém com taxas de ocorrências distintas para cada tipo. O número de D2Ds seguia uma Poisson(4), enquanto o de sensores seguia uma Poisson(2).

A partir da definição do número de comunicações, eram criados os dispositivos comunicantes e os alvos de forma que:

- A BS e os sensores eram inicializados em suas posições (detalhadas no Capítulo 2) com velocidade igual a 0;
- O UE da comunicação primária era inicializado em alguma posição dentro do pátio com iguais probabilidades, ou seja, com probabilidade distribuída uniformemente ao longo do pátio. Além disso, os UEs foram modelados com velocidade por coordenada do plano cartesiano seguindo uma distribuição uniforme $\mathcal{U}_{[-20,20]}$ km/h em que o sinal negativo indica a direção decrescente da coordenada;
- Os transmissores D2D eram inicializados em posições aleatórias dentro do pátio, com igual probabilidade para cada localização e com velocidades por coordenada do plano cartesiano seguindo uma distribuição uniforme $\mathcal{U}_{[-5,5]}$ km/h;
- Os D2Ds receptores eram inicializados próximos aos seus respectivos D2Ds transmis-

sores. A distância entre o D2D transmissor e receptor de uma comunicação D2D seguia uma distribuição uniforme $U_{[-20,20]}$ em cada coordenada do eixo cartesiano. Suas velocidades por coordenada do plano cartesiano também seguiram uma distribuição uniforme $\mathcal{U}_{[-5,5]}$ km/h;

- os alvos eram inicializados nas parcelas das rodovias que estavam fora do pátio com iguais probabilidades. Suas velocidades eram constantes e iguais a 40 km/h e, ao chegar ao centro da célula, a sua direção de movimento era invertido.

Além disso, definiu-se que nenhum D2D podia se aproximar mais do que um raio de 100m da BS, com finalidade de evitar um alto nível de interferência das comunicações primárias em *uplink*. Finalmente, quando os dispositivos atingiam as extremidades da célula, eram refletidos de acordo com a Primeira Lei da Reflexão.

As simulações foram feitas com *timestep* de 1s, o que significa que cada comunicação e sensor tinham os seus canais de comunicação amostrados e as suas potências realocadas nesse intervalo.

3.4.5 Configuração dos algoritmos

A partir das configurações do ambiente de simulação detalhadas na última seção, os algoritmos foram configurados para os seus respectivos processos de treinamento.

3.4.5.1 Redes neurais

As redes neurais utilizadas nos algoritmos apresentaram grande similaridade, embora com adaptações pontuais para atender às especificidades de cada um

A estrutura comum das redes para os agentes em todos os algoritmos mencionados segue um padrão de múltiplas camadas lineares intercaladas por normalização de camada e pela aplicação da função de ativação *Rectified Linear Unit* (ReLU). Essas camadas são responsáveis por transformar o espaço de estados em ações possíveis. A arquitetura típica pode ser descrita como:

- Uma camada linear inicial que transforma a dimensão do espaço de estados em um espaço latente de 64 unidades.
- Normalização de camada e ativação ReLU para estabilizar o aprendizado e introduzir não-linearidade.
- Expansão subsequente para 128 unidades, seguida de redução para 64 unidades, mantendo a mesma estrutura de normalização e ativação posterior.

- Uma camada linear final que mapeia as unidades para o espaço de ações.

Quanto às redes usadas como crítico, a arquitetura desenvolvida foi muito similar à do agente mas com pequenas especifidades para cada algoritmo:

- A camada de saída da rede tem dimensão igual a 1, já que a rede é usada para estimação de uma grandeza apenas, a função valor (V), no caso do PPO, ou a função ação-valor (Q), nos casos do DDPG e do TD3.
- Como o crítico do DDPG e do TD3 estima a função ação-valor, é necessário passar a ação a partir da qual se quer estimar a função Q. Para isso, as redes neurais do crítico desses algoritmos foram modificados de forma que a segunda camada tivesse neurônios adicionais para receber a ação avaliada concatenada à saída da primeira camada linear da rede.
- O REINFORCE não possui estrutura *actor-critic*, de forma que nenhuma rede foi desenvolvida para a implementação do algoritmo.

Diversas arquiteturas diferentes de redes neurais foram testadas para os diferentes algoritmos. Contudo, não se observou melhorias significativas ao empregar arquiteturas distintas para cada algoritmo. Dessa forma, optou-se por utilizar uma mesma base para todos os algoritmos e refiná-la de forma experimental.

3.4.5.2 Parametrização dos algoritmos

Quanto às parametrizações utilizadas nos algoritmos, a Tabela 3.2 a seguir resume os valores dos principais hiperparâmetros adotados em cada algoritmo. Para os parâmetros que não são aplicáveis a determinado algoritmo, utilizou-se o símbolo '-' na tabela.

Os valores atribuídos a cada hiperparâmetro resultaram de múltiplos testes, nos quais diferentes configurações do algoritmo foram treinadas e os resultados analisados. As configurações selecionadas para cada algoritmo correspondem àquelas que obtiveram o maior valor médio de retorno nos últimos 200 episódios de treinamento.

Tabela 3.2 – Parametrização dos Algoritmos DDPG, TD3, PPO e REINFORCE

Parâmetro	**DDPG**	**TD3**	**PPO**	**REINFORCE**
Episódios de treinamento	2000	2000	2000	2000
Batch size	64	64	-	-
Fator de desconto do retorno (γ)	0.99	0.99	0.90	0.99
Intervalo de sincronização das redes *target*	10	10	-	-
Tamanho da memória de *replay*	10000	10000	-	-
Fator de suavização da atualização (τ)	0.10	0.10	1	-
Desvio padrão inicial do ruído de exploração (σ_0)	25	25	20	15
Coeficiente de adaptação do ruído de exploração	1.02	1.05	1.02	1.05
Intervalo para adaptação do ruído de exploração (T_{adapt})	10	10	8	5
Desvio padrão mínimo do ruído de exploração	-	-	0.001	0.001
Otimizadores das redes neurais do agente	Adam	AdamW	Adam	Adam
Taxa de aprendizagem do agente	0.0003	0.0001	0.0003	0.001
Otimizadores das redes neurais do crítico	Adam	AdamW	Adam	Adam
Taxa de aprendizagem do crítico	0.001	0.001	0.001	
Desvio padrão do ruído adicionado nas ações (σ_μ)	-	0.2	-	-
Limitantes do ruído adicionado nas ações (c)	-	0.8	-	-
Intervalo para atualização do agente (T_{update})	-	2	-	-
Épocas de iteração para treinamento das redes (K)	-	-	50	-
Limitantes da função de perda do agente (ϵ)	-	-	0.2	-

3.5 RESULTADOS

A partir da configuração dos algoritmos propostos, os algoritmos puderam ser treinados e testados para comparação de desempenho.

Os resultados obtidos estão divididos em resultados referentes ao processo de treinamento dos algoritmos, importantes para verificação de convergência e para comparação dos algoritmos, e ao processo de teste, importantes para analisar o comportamento do algoritmo em situações não utilizadas no processo de treinamento.

3.5.1 Processo de treinamento

A configuração de cada algoritmo detalhada anteriormente foi usada para treinar 5 instâncias de cada algoritmo de DRL. A criação de múltiplas instâncias de cada algoritmo permitiu a análise da estabilidade do processo de convergência, ou seja, verificar se o treinamento conduz a resultados consistentes ou se há uma variabilidade significativa entre as instâncias treinadas com a mesma configuração.

Com base nessas múltiplas instâncias, calcularam-se a média e o desvio padrão de várias variáveis importantes tanto para os algoritmos quanto para o sistema de comunicações. Tais valores foram calculados a partir dos resultados dos últimos 200 episódios do processo de treinamento de cada algoritmo. Essas variáveis analisadas foram:

- Retorno: média do valor do retorno obtido nos episódios em questão;
- *Outage* da comunicação primária: percentual de ocorrências em que a SNIR da comunicação primária esteve abaixo da SNIR mínima do sistema para esse tipo de comunicação;
- *Outage* dos sensores: percentual de ocorrências em que a probabilidade de detecção dos sensores esteve abaixo do limiar de detecção do sistema;
- SNIR das comunicações primárias: média da SNIR das comunicações primárias;
- SNIR das comunicações D2D: média da SNIR das comunicações D2D;
- SNIR dos sensores: média da SNIR dos sensores.

Os resultados derivados deste processo detalhado são apresentados na Tabela 3.3 a seguir.

É possível perceber que todos os algoritmos de DRL alcançaram valores de retorno médio maiores do que a alocação aleatória, mostrando que todos desenvolveram alguma estratégia mais inteligente do que esta.

Tabela 3.3 – Resultados dos algoritmos de controle de potência

Variável		**aleatório**	**REINFORCE**	**DDPG**	**TD3**	**PPO**
Retorno	média	-4335.47	409.78	117.68	-1393.67	**1089.70**
	DP	-	179.99	406.70	2175.17	43.96
Outage da comm. primárias (%)	média	64.35	49.65	51.48	56.89	**11.75**
	DP	-	11.58	10.83	16.26	2.71
Outage do sensor (%)	média	38.49	**0.81**	5.04	20.30	4.38
	DP	-	0.52	4.33	18.70	0.92
SNIR das comm. primárias (dB)	média	-9.94	5.13	3.97	1.58	**10.68**
	DP	-	1.59	1.90	7.68	0.61
SNIR dos D2Ds (dB)	média	-25.64	-11.58	-12.57	-25.45	**-7.47**
	DP	-	0.71	4.83	10.75	3.85
SNIR dos sensores (dB)	média	-3.17	**9.41**	6.84	3.31	3.83
	DP	-	0.63	2.69	4.56	0.72

Dentre os algoritmos, o PPO se destacou, alcançando os maiores valores de retorno nos episódios finais, indicando o desenvolvimento de uma estratégia de alocação superior. Em sequência, REINFORCE apresentou o segundo melhor desempenho, seguido pelo DDPG e, por último, pelo TD3.

Entre as variáveis do sistema de comunicação, existe interesse especial naquelas que estão na definição da função de recompensa, que são aquelas relacionadas a *outage*, que se deseja reduzir ao máximo, e a SNIR dos D2Ds, que se deseja maximizar.

Todos os algoritmos de DRL superaram os valores médios obtidos pela alocação aleatória, confirmando que desenvolveram estratégias efetivamente mais avançadas do que a randômica.

Quanto ao *outage* das comunicações primárias, o PPO consegue reduzir essa variável de forma muito mais significativa que os demais, de 64% para 12%. O segundo algoritmo que melhor consegue controlar essa variável é o REINFORCE, mas os valores alcançados são mais do que 4 vezes piores do que os obtidos pelo PPO.

Essa redução significativa no *outage* das comunicações primárias pelo PPO pode ser atribuída ao aumento da SNIR média das mesmas. Enquanto os outros algoritmos mantêm a SNIR entre 0 e 5.13 dB, o PPO a eleva para 10.68 dB, assegurando que esta geralmente esteja acima do limiar mínimo do sistema.

Com relação ao *outage* dos sensores, o REINFORCE é o algoritmo que melhor consegue controlá-lo, reduzindo sua ocorrência para menos de 1% do tempo. O REINFORCE é seguido pelo PPO, em segundo lugar, e depois pelo DDPG e, por último, pelo TD3.

Essa redução pode ser explicada pela SNIR dos sensores, uma vez que o REINFORCE consegue alcançar uma média de 9.41 dB, enquanto os demais ficam entre 3 e 7 dB.

É curioso perceber que o PPO, apesar de ser o segundo que melhor controla o *outage* dos sensores, alcança valores médios de SNIR dos sensores bem abaixo dos valores obtidos pelo DDPG, mas consegue controlar o *outage* de forma mais eficiente.

Isso sugere que o PPO é eficaz em identificar as situações críticas que requerem proteção dos sensores, mesmo sem manter constantemente níveis elevados de SNIR para os mesmos. Isso também pode ser percebido pelo fato de o valor médio da SNIR obtido pelo PPO estar próximo ao valor obtido pelo TD3, mas o percentual de ocorrências de *outage* estar quase cinco vezes menor.

Quanto à SNIR dos D2Ds, o PPO é aquele que alcança os maiores valores médios também, seguido pelo REINFORCE, DDPG e TD3. Aqui, é interessante perceber que o PPO e o REINFORCE, ambos os algoritmos que melhor conseguiram proteger as comunicações primárias e os sensores, são aqueles que possuem maiores valores de SNIR dos D2Ds. Isso indica que a discrepância dos resultados não se deve à priorização de uma ou outra comunicação/sensor e sim à capacidade de desenvolvimento de uma estratégia de alocação mais eficiente.

Finalmente, é relevante notar que os desvios padrão dos resultados do PPO são, em geral, menores que os dos outros algoritmos, particularmente em comparação com os do DDPG e do TD3. Isso indica que o PPO tem um processo de treinamento mais estável do que os demais, convergindo geralmente para valores mais próximos entre si, enquanto os outros chegam em resultados muito diferentes ao se reexecutar o processo de treinamento em outra instância.

Além dos valores obtidos ao final do treinamento, é interessante analisarmos a evolução dessas variáveis ao longo do processo. A Figura 3.6 ilustra a evolução do retorno obtido por cada algoritmo ao longo do processo de treinamento.

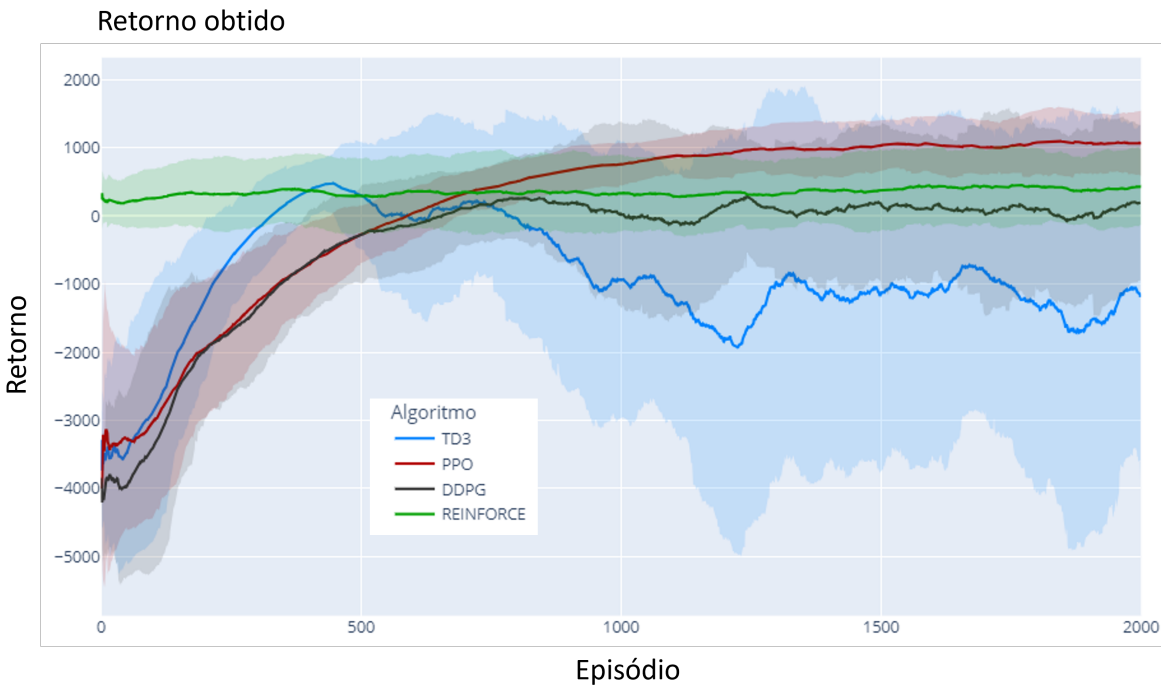


Figura 3.6 – Retorno obtido ao longo do processo de treinamento dos algoritmos. As linhas representam o valor médio, enquanto a região sombreada é o intervalo de confiança referente à linha de mesma cor. Fonte: autoria própria.

Observa-se que a convergência do REINFORCE ocorre de maneira quase imediata, sugerindo que ele capta as relações básicas nas primeiras iterações do treinamento Monte Carlo, mas não progride para a compreensão de relações mais complexas. Isso se torna evidente quando ele é posteriormente superado pelo PPO, que demora mais para alcançar valores de retorno alto mas depois se estabiliza em valores superiores aos do REINFORCE.

Quanto ao DDPG, o algoritmo também demora mais para convergir, mas chega a valores de retorno similares aos do REINFORCE, porém, com maior instabilidade, haja vista a amplitude do seu intervalo de confiança.

Por último, o TD3 demonstra algum tipo de convergência nos primeiros 500 episódios, mas depois o algoritmo parece divergir, o que é demonstrado pelo intervalo de confiança extremamente amplo. Essa instabilidade ainda maior do algoritmo faz com que o retorno médio obtido seja bem inferior aos demais.

Dado o caráter multi-objetivo do problema, é fundamental analisar as curvas das métricas do sistema que são consideradas no cálculo das recompensas. A Figura 3.7 ilustra a evolução da taxa de *outage* das comunicações primárias ao longo do processo de treinamento.

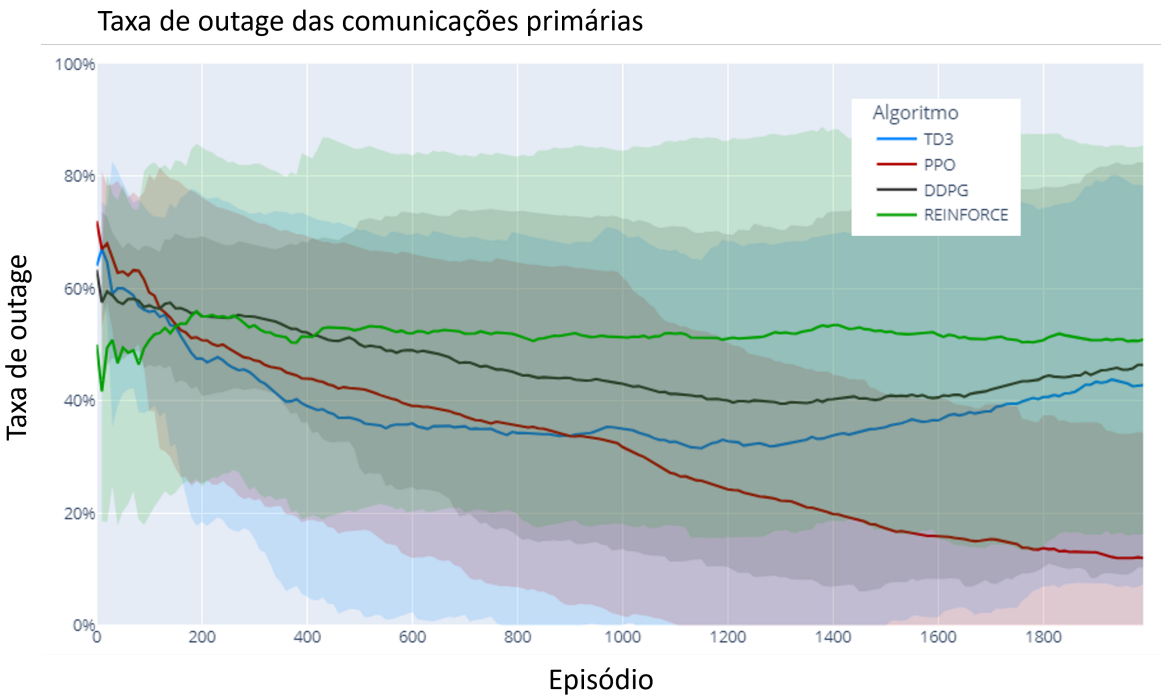


Figura 3.7 – Taxa de *outage* das comunicações primárias ao longo do processo de treinamento dos algoritmos. As linhas representam o valor médio, enquanto a região sombreada é o intervalo de confiança referente à linha de mesma cor. Fonte: autoria própria.

É perceptível que o PPO converge ao longo dos episódios, aprendendo, consistentemente, a reduzir a taxa de *outage* das comunicações primárias. Por outro lado, o DDPG e o TD3 esboçam certa redução nos primeiros episódios, mas acabam divergindo a partir do episódio 1000. Por fim, o REINFORCE não parece estar próximo da convergência em nenhum momento, aproximando-se, nos últimos episódios, dos valores obtidos pelo DDPG e TD3, muito acima do valor obtido pelo PPO.

Além da taxa de *outage* das comunicações primárias, a taxa de *outage* dos sensores também é considerada na otimização. A Figura 3.8 ilustra a evolução da taxa de *outage* dos sensores ao longo do processo de treinamento.

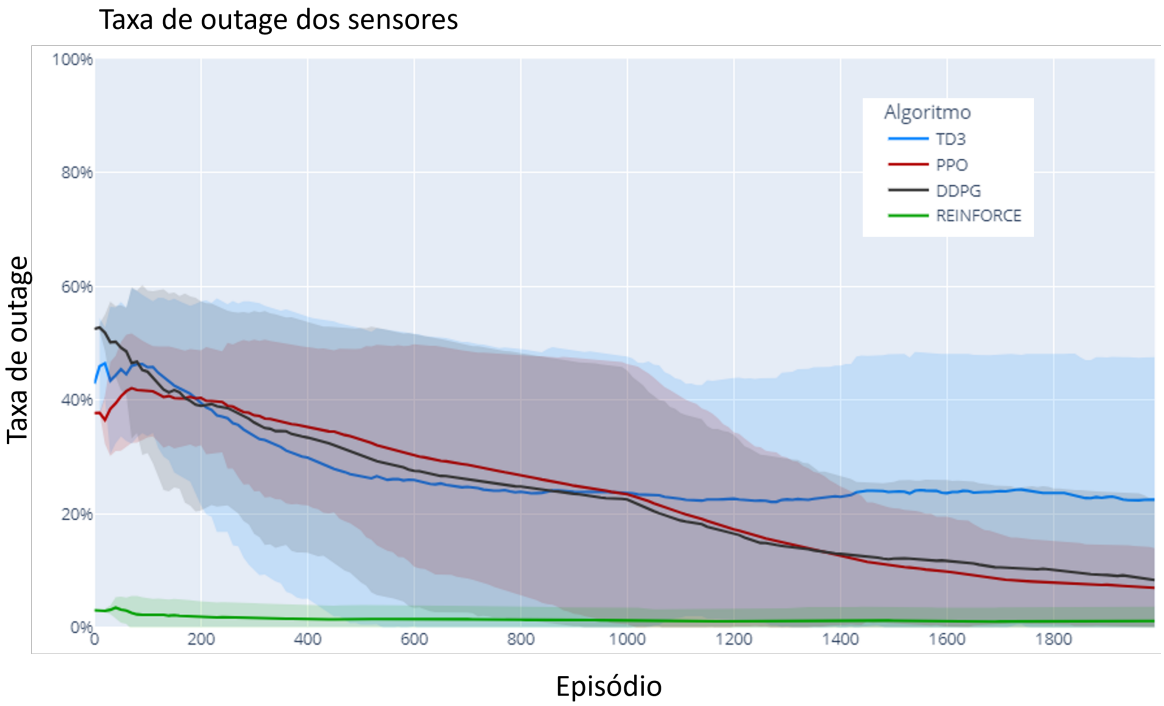


Figura 3.8 – Taxa de *outage* dos sensores ao longo do processo de treinamento dos algoritmos. As linhas representam o valor médio, enquanto a região sombreada é o intervalo de confiança referente à linha de mesma cor. Fonte: autoria própria.

O gráfico indica que o REINFORCE desenvolve rapidamente uma política eficaz de proteção aos sensores, conseguindo manter baixos índices de *outage* ao longo de todo o processo de treinamento.

Por outro lado, o PPO e o DDPG demoram mais para convergir, saindo de valores próximos a 30% nos primeiros episódios, para valores próximos a 5% nos últimos. O TD3 mostra uma ligeira redução na taxa de *outage* nos primeiros 500 episódios, mas depois se estabiliza em torno de 20%, sem alcançar o desempenho dos outros algoritmos até o final do treinamento.

Por fim, além das taxas de *outage*, o algoritmo objetivava maximizar a taxa de transmissão dos D2Ds, que pode ser analisada pela SNIR dos mesmos. Assim, a Figura 3.9 ilustra a evolução das SNIRs dos D2Ds obtidas ao longo do processo de treinamento.

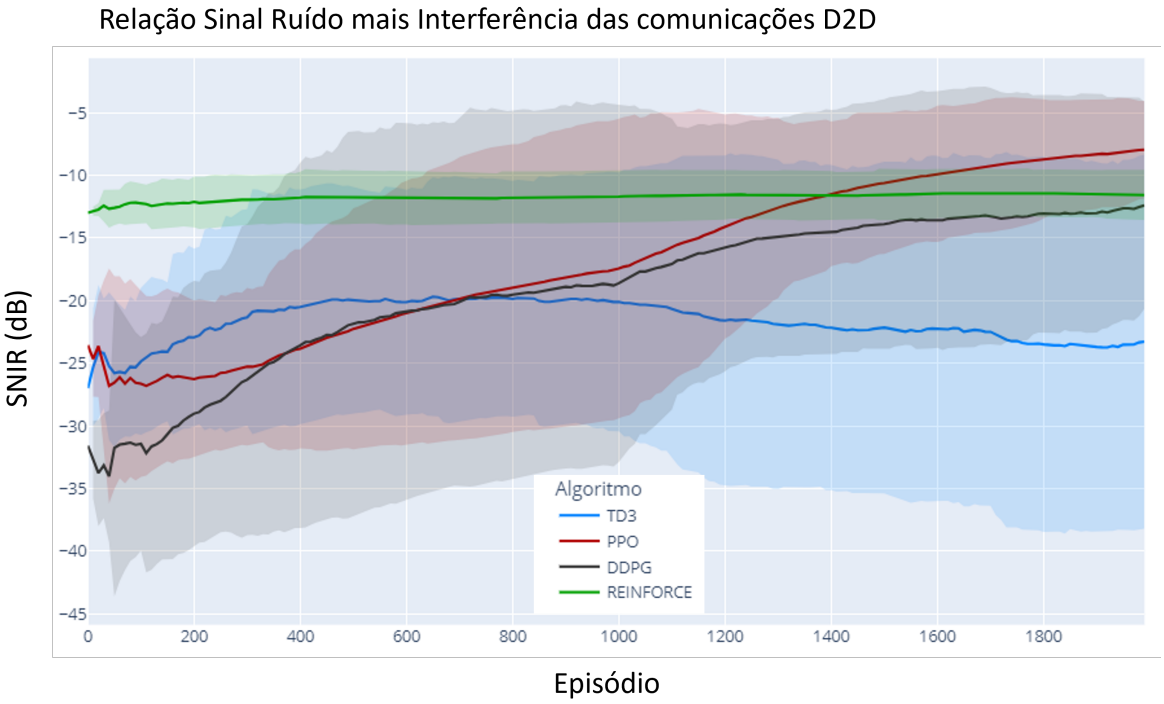


Figura 3.9 – SNIR dos D2Ds ao longo do processo de treinamento dos algoritmos. As linhas representam o valor médio, enquanto a região sombreada é o intervalo de confiança referente à linha de mesma cor. Fonte: autoria própria.

Este gráfico mostra que, ao final do processo de treinamento, o PPO supera o REINFORCE, alcançando valores mais elevados. O PPO parece acompanhar o DDPG em boa parte do processo de treinamento, mas após o milésimo episódio o seu processo de treinamento parece se tornar mais eficiente, superando os valores do DDPG.

Mais uma vez, observa-se a rápida convergência do REINFORCE em relação aos outros algoritmos, porém, ao final do treinamento, o DDPG atinge valores semelhantes a este. Em contrapartida, o TD3 parece não convergir, e termina o processo de treinamento com valores de SNIR similares aos obtidos no início do processo de treinamento.

Os resultados do processo de treinamento mostram que o PPO foi aquele que alcançou os melhores valores de retorno médio ao final do processo de treinamento, objetivo final de todo algoritmo de DRL. Além disso, foi o algoritmo que teve menor desvio padrão do retorno obtido, o que mostra que também foi o mais estável entre os algoritmos testados.

O REINFORCE, que obteve o segundo maior valor médio de retorno, demonstrou eficácia na proteção dos sensores, porém, mostrou-se menos eficiente na proteção das comunicações primárias, que era a primeira prioridade da função de recompensa desenvolvida.

Os gráficos mostraram uma convergência do REINFORCE em pouquíssimos episódios e uma estabilização praticamente completa nos episódios seguintes, com pouco ou nenhum aprendizado posterior. Esse comportamento pode indicar que o REINFORCE foi capaz de

encontrar um mínimo local rapidamente, uma estratégia que focasse na proteção dos sensores, em detrimento às comunicações primárias e aos D2Ds. Entretanto, depois disso o algoritmo parece não conseguir sair desse mínimo local e estabiliza em valores distantes aos obtidos pelo PPO, por exemplo.

Os outros dois algoritmos, DDPG e TD3, que possuem estruturas similares, iniciaram o processo de treinamento com resultados comparáveis ao PPO, mas divergiram na sua segunda metade. O DDPG ainda consegue desenvolver estratégias que protejam os sensores, mas não consegue proteger as comunicações primárias. O TD3 não é capaz de reduzir as taxas de *outage* e nem de maximizar a SNIR dos D2Ds.

Uma hipótese para o desempenho inferior do DDPG e do TD3 em relação aos outros, especialmente ao PPO, poderia ser a utilização de políticas determinísticas por estes, em contraste com as políticas estocásticas adotadas pelos outros dois algoritmos [53].

Além da comparação pelos valores de retorno obtidos, as mesmas conclusões podem ser tiradas a partir da análise dos resultados de cada uma das variáveis que compõe a função de recompensa. Dessa forma, a superioridade do PPO fica clara em relação aos demais, já que este foi o que melhor protegeu as comunicações primárias, o que alcançou taxas de transmissão mais elevadas para os D2Ds e o segundo que melhor protegeu os sensores.

Com base nesses resultados, optou-se pelo PPO como o algoritmo de escolha para o controle de potência. Assim, realizou-se uma análise aprofundada do desempenho deste algoritmo para compreender a estratégia de alocação por ele desenvolvida.

3.5.2 Processo de teste

Para realizar testes mais específicos, selecionou-se a instância do PPO que obteve os melhores resultados em termos de retorno ao final do processo de treinamento. A partir dessa instância, a rede neural do agente foi utilizada em 2 mil episódios de teste, seguindo a mesma configuração dos episódios de treinamento, mas com todos os pesos das redes congelados.

Nesses episódios de teste, foi analisada a execução do controle de potência das comunicações e dos sensores pelo algoritmo. A Figura 3.10 expõe a distribuição das potências alocadas para cada tipo de comunicação e sensor.

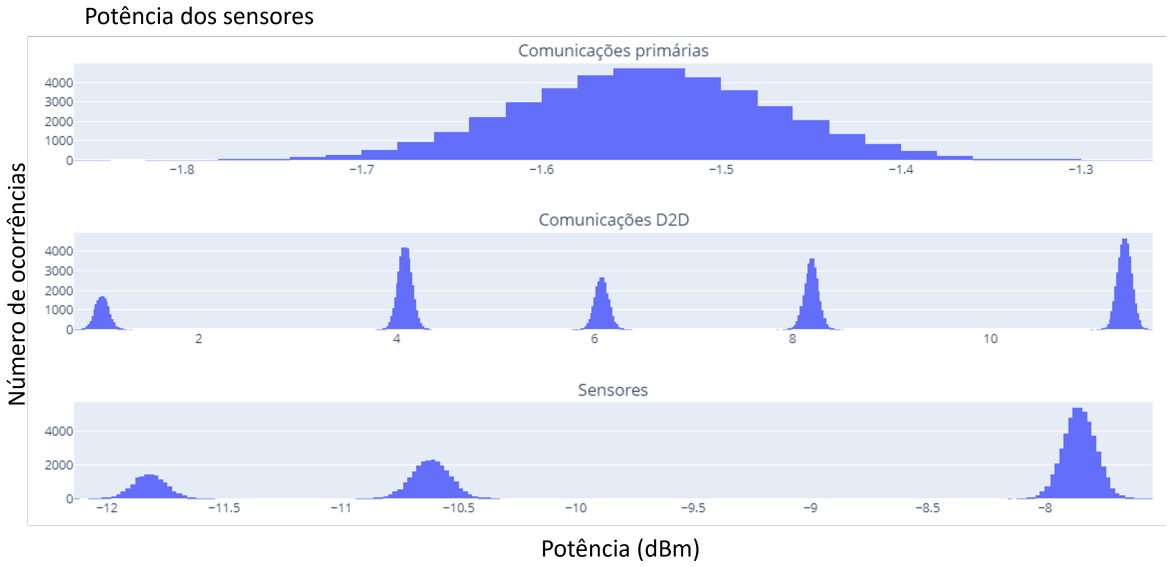


Figura 3.10 – Distribuição das potências alocadas para as comunicações primárias, D2Ds e sensores. Fonte: autoria própria.

A partir do gráfico apresentado, é possível notar um padrão de alocação das potências para cada tipo de comunicação e sensor. A distribuição das potências alocadas para as comunicações primárias assemelha-se a uma distribuição gaussiana, com média de -1.54 dBm e desvio padrão de 0.07 dBm. Isso mostra que as potências desse tipo de comunicação variam pouco de um cenário para o outro.

Por outro lado, as potências alocados para os D2Ds e para os sensores seguem um formato diferente de distribuições convencionais. A distribuição da potência alocada para os D2Ds tem 5 pontos de concentração, enquanto que a distribuição para os sensores possui 3 pontos. Valores próximos a esses pontos possuem probabilidade de alocação alta, mas para valores mais distantes, entre esses pontos de concentração, a probabilidade é próxima ou igual a 0.

Esse comportamento pode ser devido à incerteza presente no cenário de simulação, já que o número das comunicações D2D e dos sensores varia de um episódio para o outro. É possível que o algoritmo tenha aprendido que cada um desses cenários exige um padrão de alocação diferente, que não é fixo, mas varia pouco, em torno de um valor bem definido.

Um treinamento do algoritmo em um cenário com número fixo de comunicações e sensores poderia resultar em um comportamento distinto, porém, o objetivo deste trabalho foi desenvolver um algoritmo único, capaz de se adaptar a diversas configurações do sistema.

Para confirmar essa hipótese, a Figura 3.11 expõe a distribuição das potências alocadas em função do número de D2Ds e sensores no sistema. Essa imagem mostra como a potência alocada varia de acordo com as diferentes configurações do sistema.

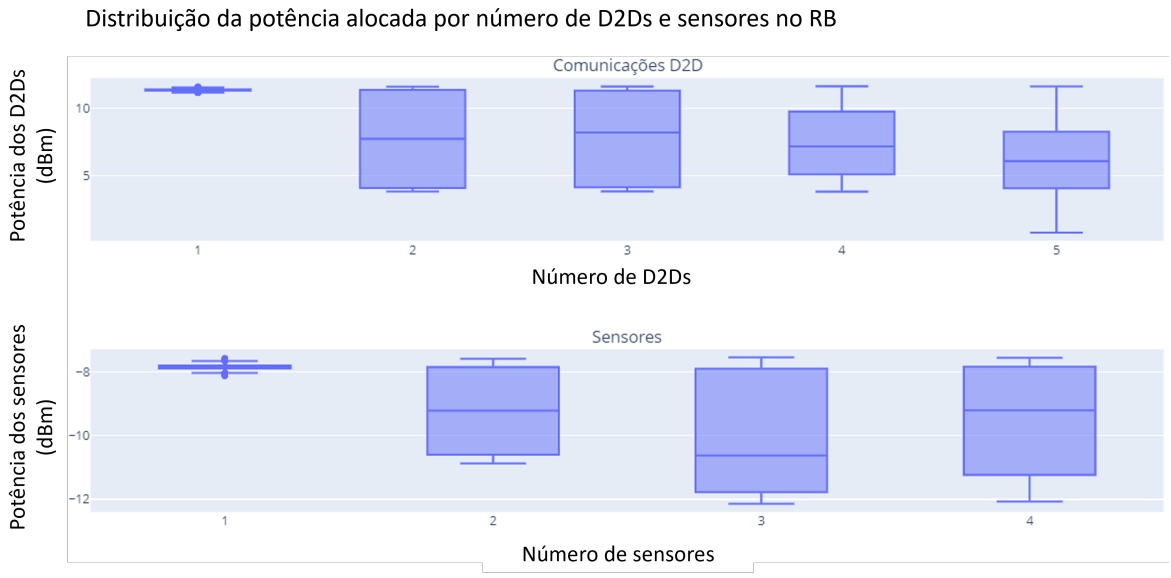


Figura 3.11 – Distribuição das potências alocadas para as comunicações D2D e sensores em função do número de D2Ds e sensores no sistema. Fonte: autoria própria.

A figura indica que a potência alocada para os D2Ds diminui à medida que o número de D2Ds no sistema aumenta. Isso provavelmente ocorre para controlar a interferência gerada por este tipo de comunicação nas demais, que possuem maior prioridade de proteção. Para os sensores, o comportamento é similar, com uma diminuição das potências alocadas à medida que o número de sensores aumenta.

Observa-se que, apesar dos intervalos claramente definidos para as potências dos D2Ds e sensores, diferentes valores dentro desses intervalos são utilizados mesmo para sistemas com o mesmo número de D2Ds e sensores. Entretanto, para certas configurações do sistema, o algoritmo não utiliza valores de certos intervalos.

Para sistemas com um único D2D, as potencias alocadas para esse tipo de comunicação está restrito apenas ao intervalo próximo a 10 dBm. Para sistemas com 2, 3 e 4 D2Ds, o algoritmo aloca potências de 4 intervalos diferentes, com a diferença que para sistemas com 2 e 3 D2Ds a mediana está no intervalo centrado próximo a 8 dBm, enquanto que para sistemas com 4 D2Ds, a mediana está no intervalo centrado próximo a 6 dBm. Para sistemas com cinco D2Ds, o algoritmo utiliza potências pertencentes aos cinco intervalos observados na Figura 3.10.

Já para os sensores, para sistemas com um único sensor, o algoritmo aloca potências de apenas um intervalo, assim como para o D2D, sendo este centrado próximo a -8 dBm. Para sistemas com 2 sensores, o algoritmo utiliza 2 desses intervalos, enquanto que para sistemas com 3 e 4 sensores, o algoritmo aloca potências dos 3 intervalos observados na Figura 3.10.

A potência alocada pelos D2Ds variou pouco em função do número de sensores no sistema, assim como a potência dos sensores variou pouco em função do número de D2Ds. Por

esse motivo, os gráficos dessa relação não foram expostos neste trabalho.

Os resultados obtidos da SNIR das comunicações e dos sensores foram igualmente monitorados, e a distribuição cumulativa desses dados é apresentada na Figura 3.12.

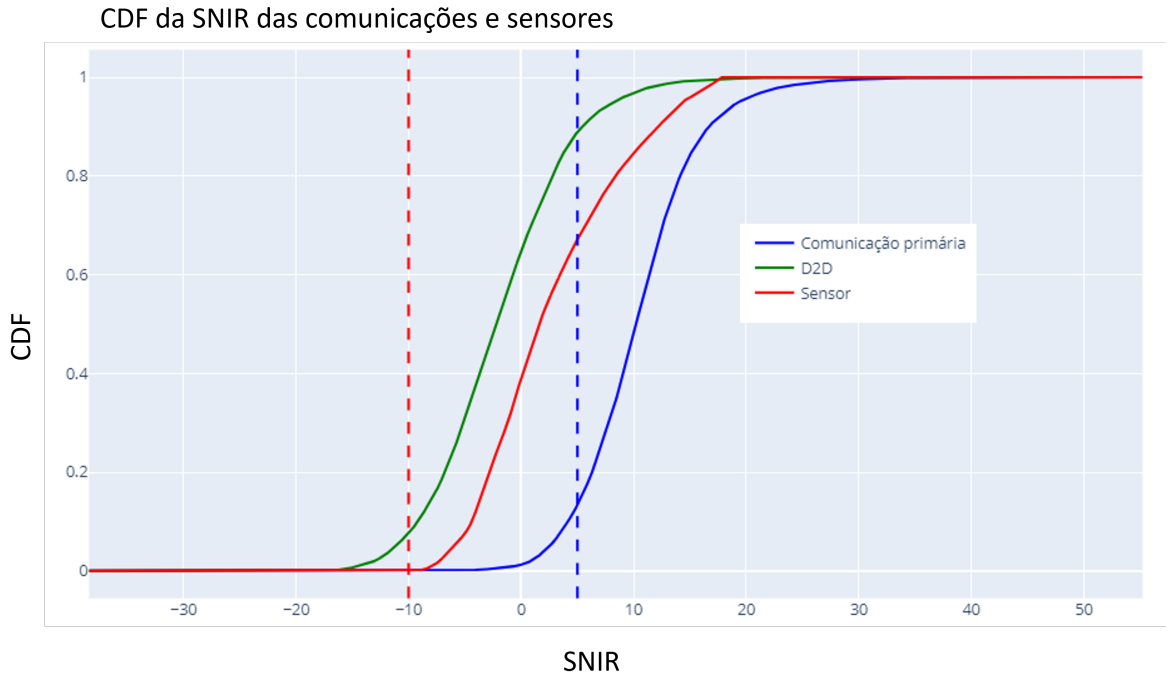


Figura 3.12 – Distribuição cumulativa das SNIRs obtidas durante o teste. As linhas verticais tracejadas representam o limiar mínimo da SNIR para a comunicação primária ou sensor não ocorrer em *outage*. Fonte: autoria própria.

É possível notar que a SNIR das comunicações primárias é maior do que a SNIR mínima em aproximadamente 88% dos eventos, resultando em *outage* em apenas 12% deles. Esse número se aproxima do valor visto ao final do processo de treinamento.

Em contraste, a instância utilizada para os testes alcançou uma taxa de *outage* dos sensores de 0% nos 2000 episódios de teste, uma melhora significativa em comparação aos 4.38% registrados ao final do treinamento.

Também é possível notar que a mediana da SNIR dos D2Ds ficou próximo a -2.2 dB, valor bem superior aos valores médios obtidos no final do processo de treinamento das diferentes instâncias, que foi igual a -7.47 dB.

Dessa forma, é perceptível que apesar da convergência do PPO ser mais estável que os demais algoritmos, os valores obtidos pela melhor instância são consideravelmente melhores do que os valores médios das diferentes instâncias ao final do processo de treinamento.

O número de comunicações D2D e sensores no RB influencia significativamente o desempenho do algoritmo de alocação de potência, o que era esperado, pois o problema se torna mais complexo à medida que esses números aumentam. Dessa forma, o modelo tenta

se adaptar ao cenário de interferências mútuas no RB para a realização da alocação.

Apesar de tal adaptabilidade, é possível que a taxa de *outage* para as comunicações primárias maior do que 0% esteja relacionado a tais cenários mais desafiadores. Assim, a Figura 3.13 ilustra a média da taxa de *outage* das comunicações primárias em função do número de D2Ds e sensores.

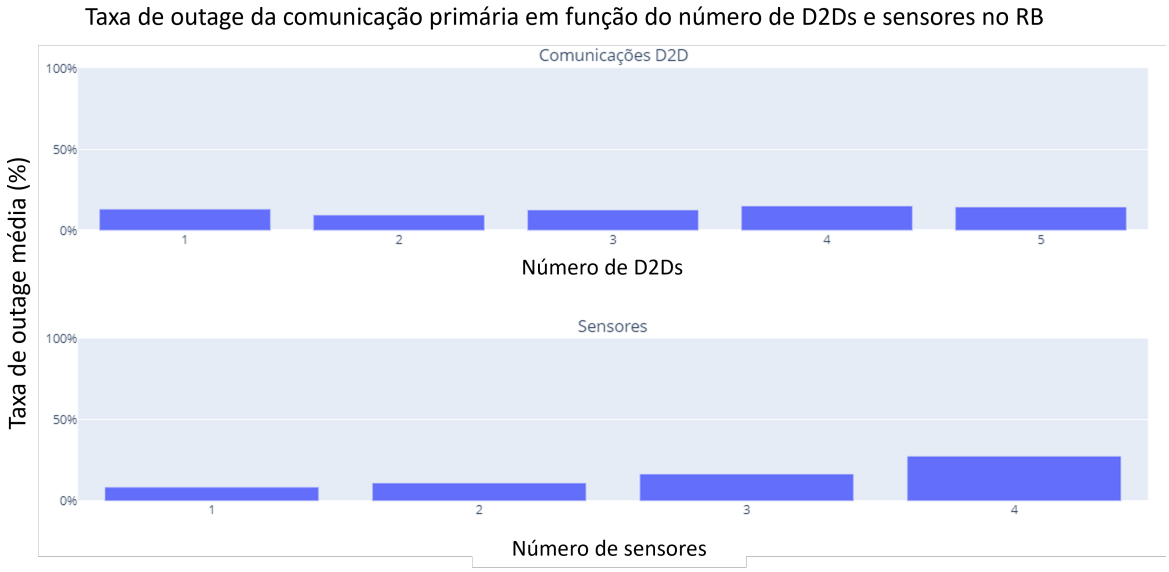


Figura 3.13 – Taxa de *outage* média em função do número de comunicações D2D e sensores no RB. Fonte: autoria própria.

Os gráficos mostram que a taxa de *outage* média das comunicações primárias não é muito afetada pela mudança do número de D2Ds no RB, mas, por outro lado, que ela é fortemente afetada pelo aumento do número de sensores. Para RBs com 1 sensor, a taxa de *outage* média das comunicações D2D ficam próximas a 8%, ao passo que para RBs com 4 sensores, essa taxa passa a ser de 28%, mais de três vezes maior.

Esses resultados indicam que o algoritmo consegue neutralizar eficazmente o impacto da variabilidade no número de D2Ds. Contudo, apesar desta adaptabilidade, cenários com um maior número de sensores continuam sendo mais desafiadores do que aqueles com poucos sensores

3.6 CONCLUSÃO

Os algoritmos de DRL testados para a realização do controle de potências das comunicações e sensores de um RB demonstraram que esses algoritmos inteligentes foram capazes de controlar as potências de forma a maximizar o objetivo definido, se comparado com um algoritmo aleatório sem mecanismos de aprendizagem.

O objetivo dos algoritmo foi proteger as comunicações primárias e os sensores de forma mas eficiente, além de maximizar a SNIR das comunicações D2D de forma oportunística. O algoritmo de DRL que obteve melhor desempenho entre os testados foi o PPO.

O PPO foi o algoritmo que teve a menor taxa média de *outage* das comunicações primárias, a segunda menor taxa média de *outage* dos sensores e a maior média de SNIR das comunicações D2D.

Se comparado com a alocação aleatória, o PPO foi capaz de reduzir a taxa média de *outage* das comunicações primárias de 64.35% para 11.75%, reduzir a taxa de *outage* dos sensores de 38.49% para 4.38% e aumentar a média da SNIR das comunicações D2D de -25.64 dB para -7.47 dB, comparando em valores absolutos, um ganho de mais de 4752%.

Além disso, foi possível perceber que o desempenho do algoritmo não é afetado consideravelmente pelo aumento da quantidade de comunicações no RB, mostrando que uma instância treinada do algoritmo é capaz de atuar em RBs com diferente configurações sem grandes degradações de desempenho.

4 ALOCAÇÃO DE ESPECTRO

4.1 INTRODUÇÃO

A alocação de espectro em sistemas de comunicações móveis constitui uma área de estudo e desenvolvimento essencial na arquitetura e operação das redes de telecomunicações modernas, particularmente aquelas que se beneficiam das tecnologias de quinta (5G) e sexta geração (6G) [27, 3]. Esta tarefa estratégica envolve a designação eficaz de recursos de espectro, especificamente a definição de quais comunicações serão alocadas em cada bloco de recursos (RB) disponível no sistema.

Em um contexto onde o espectro radioelétrico é um recurso limitado e altamente regulamentado, a alocação eficiente do espectro garante a coexistência de diversos serviços e tecnologias dentro do mesmo espaço de frequência, incluindo tipos de comunicações diferentes, como as comunicações primárias e as comunicações D2D, além do sensoreamento, em sistemas JCAS.

O papel da alocação de espectro torna-se ainda mais crítico à medida que as redes móveis evoluem para atender às crescentes demandas por largura de banda mais alta, menor latência e maior confiabilidade. Os sistemas 5G e 6G estão sendo desenvolvidos para suportar uma nova geração de aplicações, desde realidade aumentada e veículos autônomos até cidades inteligentes e fábricas automatizadas da Indústria 4.0. Para materializar isto, uma gestão eficiente do espectro é imperativa, garantindo que as comunicações críticas obtenham a largura de banda necessária para operar da forma esperada, enquanto minimiza a interferência entre serviços e maximiza a taxa de transmissão das comunicações oportunísticas.

O interesse pela alocação de espectro eficiente e dinâmica se intensifica diante dos desafios impostos pelas novas demandas de tráfego e pelos requisitos de desempenho dos sistemas 5G e 6G. A capacidade de adaptar a alocação de espectro em tempo real, em resposta às variações na demanda de tráfego e às condições da rede, é fundamental para otimizar o desempenho da rede e a experiência do usuário. Além disso, a introdução de novas tecnologias, como o acesso ao espectro compartilhado e as redes definidas por software, oferece oportunidades para a utilização de estratégias de alocação de espectro inteligentes, mas também adiciona camadas de complexidade à sua gestão.

Contudo, a implementação eficaz de estratégias de alocação de espectro enfrenta numerosos desafios técnicos. Os sistemas de comunicações móveis operam em um ambiente altamente dinâmico, onde as condições de canal, a distribuição do tráfego e os canais interferentes podem variar significativamente em curtos períodos de tempo.

Neste cenário, a alocação de espectro emerge como uma questão central na concepção e operação de redes móveis de próxima geração, desempenhando um papel vital na maximização da eficiência espectral e no atendimento às expectativas de desempenho do sistema.

Neste capítulo será detalhado o algoritmo proposto para alocação de espectro do sistema, discorrendo sobre os aspectos teóricos das técnicas utilizadas, as configurações usadas para desenvolvimento do algoritmo, o ambiente de teste utilizado e os resultados obtidos.

4.2 DEFINIÇÃO DO PROBLEMA

Após o detalhamento da abordagem do primeiro subproblema, o controle de potências, no Capítulo 3, é preciso detalhar a abordagem utilizada para o segundo subproblema, o da alocação do espectro. Tal subproblema pode ser descrito matematicamente como:

$$\max_{b_{j,k},b_{q,k}} \sum_{l=1}^{L} \psi_l \tag{4.1}$$

$$\text{s.t. } \psi_a \geq \psi_{\text{min}}, \quad \forall\, a \in \mathbf{A} \tag{4.2}$$

$$P_q^d \geq \phi_{\text{min}}, \quad \forall\, q \in \mathbf{Q} \tag{4.3}$$

$$\psi_j = \sum_{k}^{K} \log_2(1 + b_{j,k}\zeta_{j,k}), \quad \forall\, j \in \mathbf{J} \tag{4.4}$$

$$\sum_{k=1}^{K} b_{j,k} = 1, \quad \forall\, j \in \mathbf{J} \tag{4.5}$$

$$\sum_{k=1}^{K} b_{q,k} = 1, \quad \forall\, q \in \mathbf{Q} \tag{4.6}$$

em que:

$$b_{j,k} \in [0,1], \quad \forall\, j \in \mathbf{J} \text{ e } \forall\, k \in \mathbf{K} \tag{4.7}$$

$$b_{q,k} \in [0,1], \quad \forall\, q \in \mathbf{Q} \text{ e } \forall\, k \in \mathbf{K} \tag{4.8}$$

$$\mathbf{J} = \mathbf{A} \cup \mathbf{L} \tag{4.9}$$

em que p_j^t e p_q^t não são mais variáveis de decisão do subproblema, e sim parâmetros definidos, ao passo que $b_{j,k}$ e $b_{q,k}$ passam a ser as variáveis de decisão.

As variáveis de decisão do subproblema são inteiras e ele também contém não-linearidades, além de não ser convexo. Tais características tornam o problema complexo e fazem com que diversas abordagens não sejam aplicáveis, como as técnicas de Programação Linear Inteira Mista (*Mixed Integer Linear Programming* - MILP)

Além disso, o problema é NP-hard, combinatorial e dinâmico. O campo da otimização combinatorial é marcado pela complexidade e diversidade de problemas, muitos dos quais possuem um vasto espaço de soluções possíveis, o que torna a busca por soluções ótimas ou aproximadas um desafio computacional significativo. Em cenários de otimização onde os problemas são dinâmicos, a constante evolução das instâncias do problema intensifica a necessidade de abordagens adaptativas.

Neste contexto, o uso de hiper-heurísticas emerge como uma abordagem promissora. As HHs, operando no espaço de heurísticas, oferecem uma resposta adaptativa a essas mudanças. Em vez de se restringir a uma solução estática, elas ajustam-se continuamente às variações do problema, proporcionando soluções relevantes mesmo diante de alterações no cenário combinatorial. Esta capacidade de resposta dinâmica posiciona as HHs como uma ferramenta promissora para abordar problemas que não apenas possuem vastos espaços de solução, mas também apresentam mudanças ao longo do tempo.

4.3 HIPER-HEURÍSTICAS

Historicamente, a abordagem para resolver problemas combinatoriais muito complexos, principalmente não convexos, envolve o desenvolvimento de heurísticas específicas, meticulosamente projetadas para atender às peculiaridades de cada problema.

No entanto, tal abordagem possui algumas limitações. O desenvolvimento de heurísticas especializadas é frequentemente um processo complexo, que exige um profundo conhecimento do problema em questão e, muitas vezes, resulta em soluções subótimas que não são facilmente adaptáveis a dinamicidade do problema.

Ao longo da última década, as hiper-heurísticas têm recebido uma atenção crescente da comunidade científica. O estudo [19] foi um dos pioneiros ao concluir que a combinação de diferentes heurísticas de baixo nível produzia soluções de melhor qualidade do que quando aplicadas isoladamente. Esta descoberta ressaltou que heurísticas individuais podem ser eficazes em certos estágios do processo de busca, mas podem não ser tão eficientes em outros.

Diferentemente das heurísticas tradicionais, que operam diretamente no espaço de soluções do problema, as hiper-heurísticas operam no espaço de heurísticas. Em outras palavras, enquanto as heurísticas tradicionais buscam soluções para o problema, as hiper-heurísticas buscam heurísticas que possam resolver o problema. A Figura 4.1 ilustra a relação e a diferença entre o uso de hiper-heurísticas e de heurísticas comuns.

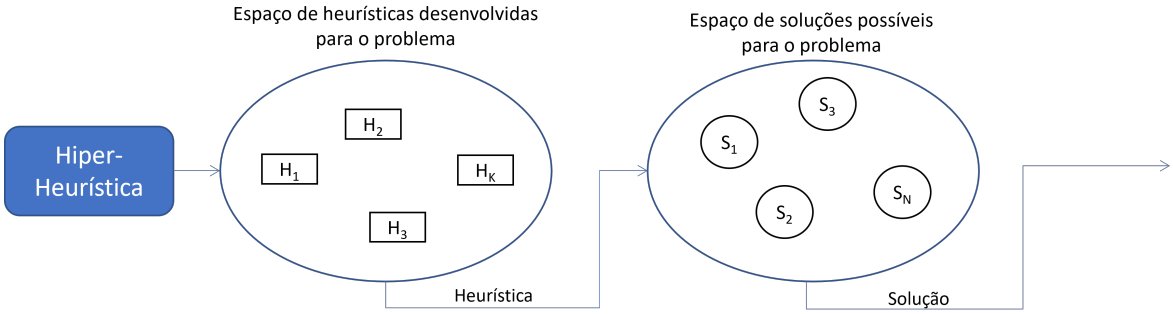


Figura 4.1 – Diagrama representando a atuação das hiper-heurísticas no espaço de heurísticas, enquanto as heurísticas são de fato as responsáveis por atuar no espaço de soluções do problema. Fonte: autoria própria.

As hiper-heurísticas, por definição, fazem uma busca em alto nível à heurística (ou a combinação delas) que oferece a melhor solução para o problema apresentado. Nesse cenário, essas heurísticas de busca por soluções são chamadas de Heurísticas de Baixo Nível (*Low Level Heuristic* (LLH)) [54].

As LLHs podem ser operadores de movimento desenvolvidos com base em um conhecimento intrínseco do problema ou podem ser algoritmos mais complexos, como metaheurísticas. Existem problemas que possuem heurísticas bem documentadas na literatura científica, que geralmente são usadas como LLHs, enquanto outros problemas precisam de mais esforço para elaboração das LLHs.

O objetivo da implementação de uma hiper-heurística é desenvolver uma abordagem mais generalizada, que não seja estática e que consiga se adaptar às mudanças do problema. Com isso, espera-se que elas possam ser aplicadas a uma variedade mais ampla de problemas e instâncias, reduzindo a necessidade de desenvolvimento contínuo de heurísticas especializadas.

A seguir, é feita a classificação das HHs sob diferentes aspectos:

4.3.1 HHs de seleção ou geração

As hiper-heurísticas podem ser categorizadas, principalmente, em dois tipos: de geração e de seleção.

- As HHs de geração concentram-se em criar novas heurísticas, combinando ou modificando heurísticas existentes;
- As HHs de seleção focam em escolher a heurística mais adequada de um conjunto preexistente, dependendo da instância em que o problema se encontra.

Ambas as estratégias podem ser utilizadas em diferentes contextos e problemas, entre-

tanto, para problemas dinâmicos combinatoriais, gerar uma heurística geral que seja adaptável o suficiente para os diferentes estados do problema é muito complexo [16]. Nesse contexto, HHs de seleção costumam ser mais utilizadas, de forma que a HH seja responsável por selecionar qual LLH deve retornar a melhor solução para a instância do problema apresentado [17, 16].

Como o problema abordado neste capítulo é dinâmico e combinatorial com um vasto espaço de soluções, focou-se em desenvolver HHs de seleção.

4.3.2 LLHs construtivas e perturbativas

Outra classificação possível é quanto ao tipo das LLHs, que podem ser construtivas ou perturbativas [55].

- As LLHs construtivas geram uma solução completa, sem depender de uma solução prévia. Nesse contexto, em cada etapa do processo, uma LLH é selecionada para adicionar um componente à solução, até que uma solução completa seja formada;
- As LLHs perturbativas operam em soluções iniciais. Essas LLHs realizam operações de busca local em soluções prévias. Este é um processo iterativo que continua até que algum critério de término seja atendido.

É importante destacar que as categorias construtivas e perturbativas não são mutuamente exclusivas. Uma HH pode combinar ambas as abordagens, utilizando LLHs construtivas para formar soluções iniciais e LLHs perturbativas para refiná-las, com o objetivo de alcançar resultados superiores [16, 55].

Para problemas dinâmicos, em que o problema muda aos poucos, é comum se utilizar abordagens que usem as soluções implementadas anteriormente para ir adaptando-as até a convergência [55, 15]. As LLHs perturbativas geram soluções que podem seguir esse paradigma, se a solução do *timestep* anterior for tratada como solução inicial em que as LLHs serão aplicadas para perturbá-la. Assim, focou-se em desenvolver HHs com LLHs perturbativas.

4.3.3 Mecanismo de aprendizagem das HHs

As HHs utilizam diferentes mecanismos de aprendizado para controlar e adaptar a seleção de LLHs. Estes mecanismos são categorizados da seguinte forma [15, 16]:

- As HHs com aprendizado *online* recebem uma avaliação, em tempo real, para cada seleção de LLH realizada, de forma similar às recompensas recebidas pelos algoritmos

de RL. Dessa forma, a HH aperfeiçoa sua estratégia de seleção à medida que recebem tais avaliações durante o processo de busca;

- As HHs com aprendizado *offline* aprendem a partir de um conjunto de instâncias do problema armazenadas, nas quais alguns padrões relevantes de solução são identificados e aprendidos. Tal estratégia de aprendizagem para as HHs são mais utilizadas para HHs de geração, em que se busca identificar padrões relevantes do problema para generalização em problemas ainda não vivenciados;
- As HHs com aprendizado misto combinam característica de ambos os tipos de aprendizagem. A abordagem mais comum é usar aprendizado *offline* inicialmente a partir de um conjunto de instâncias guardadas e, posteriormente, refinar a sua atuação em tempo real, adaptando-se às novas informações recebidas durante o processo de busca;
- As HHs sem processo de aprendizado são aquelas desenvolvidas a partir de um conhecimento intrínseco do problema, sem nenhum processo de aprendizagem.

Hiper-heurísticas de seleção para problema dinâmicos utilizam geralmente processos de aprendizado *online* ou misto, para possibilitar uma adaptação às mudanças do problema. Entretanto, a depender da complexidade do problema, se torna muito custoso desenvolver algoritmos inteligentes capazes de aprender a selecionar as LLHs. Nesses cenários, as HHs sem processo de aprendizagem podem se tornar uma opção viável.

O problema abordado neste capítulo possui características que o tornam complexo, como a não linearidade, a não convexidade além da sua natureza combinatorial e dinâmica. Por esse motivo, focou-se tanto em HHs com processo de aprendizado *online* quanto em HHs sem processo de aprendizado.

4.3.4 Parametrização das LLHs

As hiper-heurísticas podem ser classificadas com base na maneira como os parâmetros das LLHs são configurados ao longo dos *timesteps* do problema. Esta classificação é fundamental para entender como as HHs se adaptam e otimizam a solução de problemas. As categorias são [17]:

- As LLHs com parametrização estática são aquelas cuja configuração não muda ao longo do tempo;
- as LLHs com parametrização dinâmica são aquelas cuja configuração varia ao longo do tempo seguindo uma estratégia pré-configurada;
- As LLHs com parametrização adaptativa são aquelas cuja configuração varia de forma reativa, adaptando-se à dinamicidade do ambiente;

- As LLHs com parametrização auto-adaptativa são aquelas cuja configuração não apenas varia ao longo do tempo, como é decidida em conjunto com a melhor solução para a escolha das LLHs em cada *timestep*.

O uso das diferentes estratégias de parametrização das LLHs depende muito do problema abordado, das LLHs disponíveis e da estratégia de aprendizagem da HH. Existem LLHs que sequer são parametrizáveis.

4.3.5 Algoritmos frequentemente usados como HHs

Diferentes tipos de algoritmos podem ser usados como Hiper-heurísticas, dos mais simples aos mais complexos. Entre os mais simples, é possível citar algoritmos desenvolvidos pela definição de regras de negócio, ou seja, rotinas fixas, geralmente seguindo lógicas condicionais, criadas a partir do conhecimento humano especializado na área [15, 56].

Entre os algoritmos de complexidade moderada, é possível utilizar técnicas de meta-heurísticas conhecidas, como os Algoritmos Genéticos ou a Colônia de Formigas [54, 56].

Algoritmos desse tipo são, em geral, de fácil implementação e suficientes para encontrar soluções úteis, ainda que nem sempre em um ótimo global [56]. Entretanto, algoritmos desse tipo possuem inferência custosa, isto é, para se encontrar uma solução, o algoritmo faz diversas buscas no espaço de soluções do problema, o que demanda tempo e capacidade computacional [38].

Por fim, entre os algoritmos de complexidade elevada, estão algoritmos de aprendizado profundo, como as redes neurais profundas treinadas por meio do DRL [18]. Essa técnica permite que as HHs desenvolvidas aprendam a escolher LLHs de forma a otimizar a sua função de custo [54].

Diferentemente das meta-heurísticas, o custo computacional de inferência de um algoritmo desse tipo é pequeno, de forma que as suas soluções podem ser usadas em problemas *online* com baixa latência. Em contrapartida, o desenvolvimento de tais algoritmos é mais complexo, exigindo a elaboração e a execução de um processo de treinamento das redes neurais em milhares de iterações [11].

Por conta da exigência de baixa latência do problema abordado neste trabalho, focou-se em algoritmos cujo processo de inferência fosse pouco custoso computacionalmente. Dessa forma, optou-se por desenvolver algoritmo baseados em DRL e algoritmos baseados em regras de negócio, para servir como linha de base de comparação [54].

4.4 IMPLEMENTAÇÃO

O desenvolvimento de uma HH de seleção em qualquer problema depende da elaboração das LLHs que servirão de opção de escolha. Apesar dessas LLHs não precisarem abarcar todas as possibilidades de solução para o problema completo, dada a implementação da HH, desenvolver LLHs que viabilizem um bom funcionamento da HH não é trivial.

Além disso, o papel da HH se torna mais relevante quanto maior for a especificação da ação das LLHs desenvolvidas. Isso se dá porque a tarefa de generalização do problema fica a cargo da HH, que tem a sua disposição uma série de heurísticas de baixo nível que executam tarefas simples.

Por esse motivo, neste trabalho, cada LLH foi desenvolvida para atuar focada em um único RB por vez, de forma que a solução de alocação do espectro do sistema se dá pela chamada de um conjunto de LLHs, passando por todos os RBs disponíveis. A Figura 4.2 ilustra a forma de uso da HH por RB para atualização da alocação do espectro de um *timestep* para o próximo.

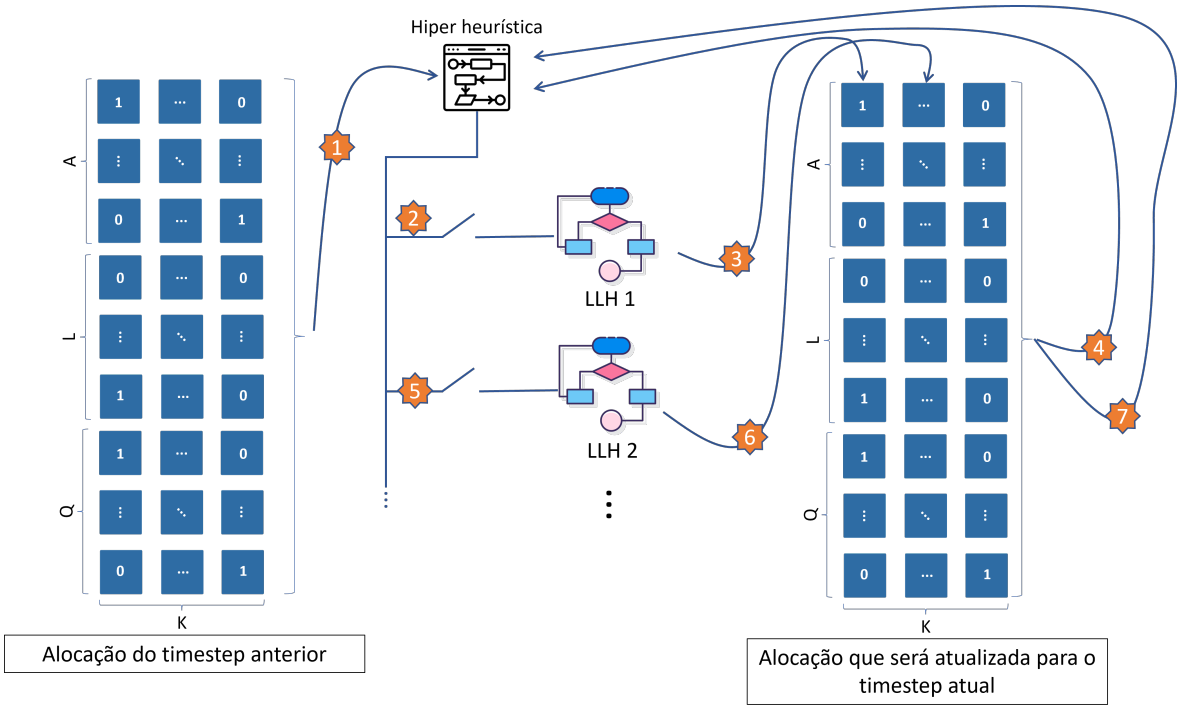


Figura 4.2 – Diagrama representando as múltiplas chamadas da hiper-heurística para definição de uma LLH por RB até a atualização da alocação do espectro para o *timestep* seguinte. Fonte: autoria própria.

Como o problema é dinâmico, a alocação do espectro é refeita após um *timestep* para adequar as mudanças ocorridas no sistema. Isso faz com que as soluções sejam refinadas com o passar do tempo, de forma que ainda que a alocação nos instantes iniciais não sejam ótimas, a longo prazo a solução possa convergir para um ótimo global.

A opção de desenvolver as LLHs para atuarem especificamente em um RB se deu pela simplificação do problema e, consequentemente, do desenvolvimento das heurísticas de baixo nível. O problema de alocação do espectro é praticamente simétrico de um RB para o outro, o que permite a generalização do uso de uma HH para todos os RBs.

Além disso, desenvolver as LLHs a nível de RB é mais simples do que uma LLH que atuasse em todos os RBs ao mesmo tempo, já que é possível pensar apenas no estado do RB em questão, identificando se existe risco para as comunicações prioritárias ou para os sensores do RB ou se existe oportunidade para inserção de comunicações D2D.

Assim, foram desenvolvidas oito heurísticas de baixo nível baseadas em conhecimento humano do problema. Para a criação dessas LLHs, objetivou-se cobrir um conjunto de subobjetivos que desse opções à HH de corrigir ou otimizar o sistema baseado do estado atual deste, buscando englobar os cenários mais prováveis relacionados à alocação do espectro. Assim, as oito LLHs estão descritas a seguir, descrevendo o que cada uma faz e qual é o seu subobjetivo dentro do problema:

- **Heurística 1:**
  - Ação: Retira o sensor com a maior interferência recebida do RB atual e o coloca no RB onde ele recebe a menor interferência;
  - Objetivo: Evitar que um sensor fique com probabilidade de detecção abaixo do limiar definido para o sistema.
- **Heurística 2:**
  - Ação: Remove a comunicação D2D que gera a maior interferência em um UE do RB atual e o coloca no RB cujos UEs recebem a menor interferência somada;
  - Objetivo: Evitar ocorrência de *outage* da comunicação prioritária do RB atual sem gerar *outage* na comunicação prioritária de outro RB.
- **Heurística 3:**
  - Ação: Retira a comunicação D2D que gera a maior interferência nos sensores do RB atual e o coloca no RB onde ele gera a menor interferência nos sensores;
  - Objetivo: Diminuir a interferência gerada pela comunicação D2D nos sensores, sem gerar risco de diminuição do desempenho dos sensores de outro RB para níveis abaixo do limiar de detecção definido para o sistema.
- **Heurística 4:**
  - Ação: Retira a comunicação D2D que recebe a maior interferência no RB atual e o realoca para o RB onde ele recebe a menor interferência;

  - Objetivo: Aumentar a eficiência espectral da comunicação D2D mais prejudicada no RB, passando-a para um RB em que possuirá mais oportunidade.

- **Heurística 5:**
  - Ação: Seleciona a comunicação D2D fora do RB atual que causa a menor interferência na comunicação prioritária e o realoca para o RB atual;
  - Objetivo: Aumentar a eficiência espectral do RB de maneira conservadora para a comunicação prioritária, isto é, trazendo a comunicação D2D que causa menor interferência nela.

- **Heurística 6:**
  - Ação: Escolhe a comunicação D2D fora do RB atual que receberia a menor interferência e o realoca no RB atual;
  - Objetivo: Aumentar a eficiência espectral do RB de forma mais agressiva.

- **Heurística 7:**
  - Ação: Identifica o sensor fora do RB atual que receberia a menor interferência e o realoca no RB atual;
  - Objetivo: Aumentar a probabilidade de detecção de um radar externo que possui oportunidade do RB atual.

- **Heurística 8:**
  - Ação: Mantém a alocação atual de recursos sem fazer alterações;
  - Objetivo: Preservar o estado atual da alocação de recursos quando não é identificada uma opção mais eficiente.

A partir das LLHs apresentadas, as HHs foram desenvolvidas para selecioná-las para geração de uma solução. Para comparação de resultados, foram desenvolvidas diferentes HHs, uma de decisão aleatória, para servir de linha de comparação, e outras baseadas em diferentes algoritmos de DRL.

4.4.1 Algoritmos de DRL relevantes para desenvolvimento da Hiper-heurística

Como os algoritmos de DRL são mecanismos de decisão com aprendizagem dinâmica, tais técnicas podem ser utilizadas como HHs de seleção com mecanismo e aprendizagem *online*. Desta forma, desenvolveram-se algumas HHs baseadas em diferentes algoritmos de DRL presentes na literatura.

A conceituação dos algoritmos de DRL apresentada na Seção 3.3 é válida para os algoritmos que serão apresentados nesta seção. No Capítulo 3, foram apresentados 4 algoritmos de DRL: o DDPG, o TD3, o REINFORCE e o PPO. Entre esses algoritmos, apenas o PPO é capaz de atuar em problemas com espaço de ações de natureza discreta; os demais só são capazes de definir ações de natureza contínua.

Como o problema de seleção de uma das LLHs desenvolvidas é um problema com espaço de ações de natureza discreta, apenas o PPO, entre os algoritmos implementados, será empregado como HH. Entretanto, existem outros algoritmos de DRL capazes de atuarem em problemas com ações discretas que podem ser aplicadas no problema em questão.

Entre os algoritmos mais clássicos de DRL para isso, está o *Deep Q-Network* (DQN), um dos primeiros e mais referenciados algoritmos de DRL [44]. Além do DQN, serão implementados o *Duelling DQN*, o *Double DQN* (D2QN) e o *Duelling Double DQN* (D3QN), evoluções do DQN que buscam melhorar o desempenho do algoritmo através de técnicas adicionais.

Estes algoritmos são baseados em valor (ver Seção 3.3.4), que estimam a função Q para cada ação no espaço de ações e escolhem aquela com maior função ação-valor. Por esse motivo, não são capazes de definir ações em um espaço de natureza contínua, assim como seu desempenho é afetado pelo aumento da cardinalidade do espaço de ações possíveis.

4.4.1.1 DQN

A DQN é um algoritmo de simples implementação, cujo objetivo é desenvolver um bom estimador da função Q para as ações possíveis em um estado qualquer. Uma vez que as estimativas feitas pelo algoritmo são próximas ao valor real, é uma boa estratégia para maximização da esperança do retorno escolher como ação a ser realizada aquela que o algoritmo estimar como maior função Q [13, 44].

Esta estrutura foi baseada no *Q-network*, algoritmo antecessor à DQN. O diferencial da DQN para os seus antecessores foi a utilização de redes neurais profundas como estimador da função Q, propondo um mecanismo de aprendizagem dessas redes neurais [13, 44]. Tal mecanismo se baseia na função de custo descrita na Equação (4.10).

$$J(\theta) = (\gamma \max_{a} Q'(s_{t+1}, a) + r_t) - Q(s_t, a_t) \tag{4.10}$$

A partir dessa equação é possível reduzir a distância entre o valor estimado para a função Q utilizando apenas a rede neural, com uma estimativa usando as recompensas coletadas ao longo das trajetórias pelas quais o algoritmo passou. Assim como o DDPG e o TD3, o DQN também estabiliza o processo de aprendizado a partir do *Experience-Replay* e do uso

de redes *target*.

O diagrama com uma ilustração do treinamento do DQN está mostrado na Figura 4.3 e o pseudocódigo está descrito no Algoritmo 5.

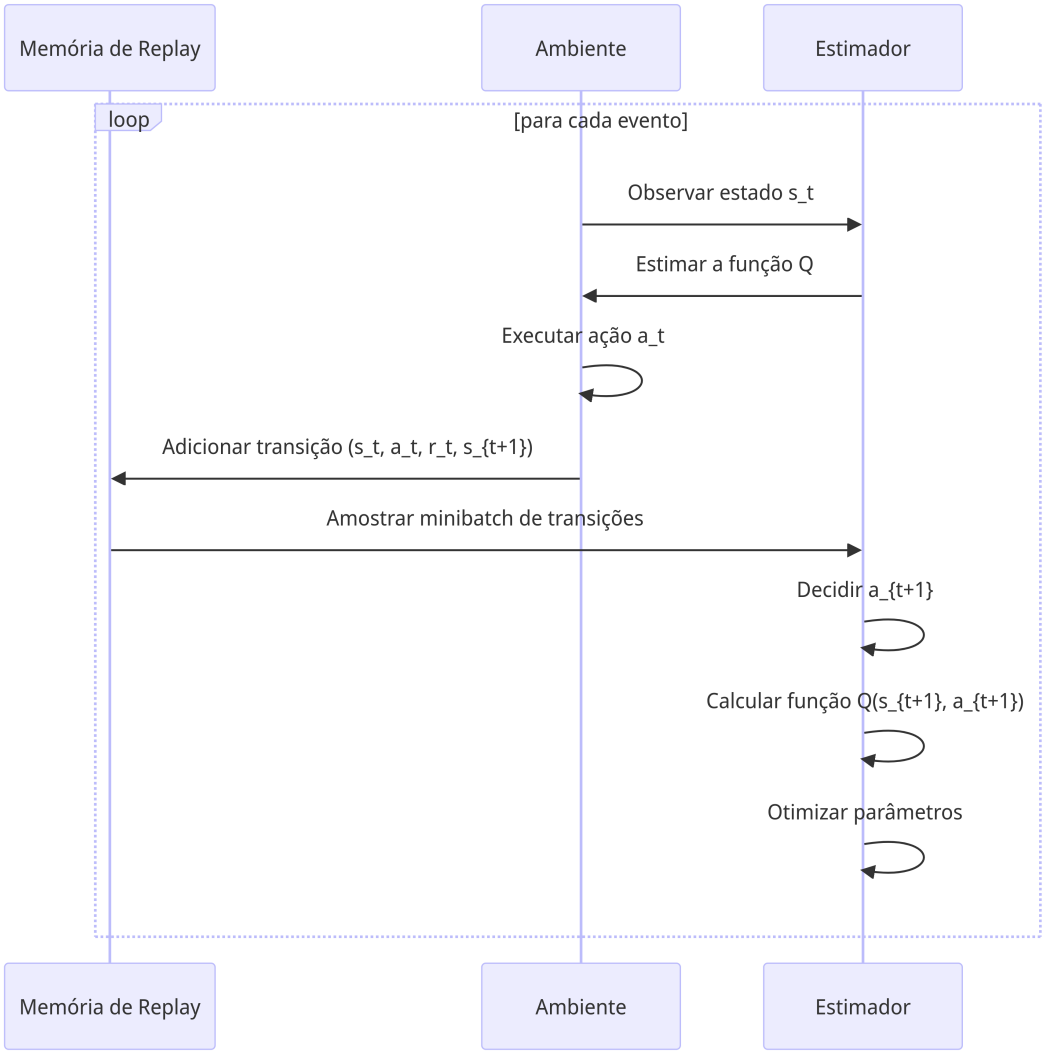


Figura 4.3 – Diagrama do processo de treinamento do DQN. Fonte: autoria própria.

A complexidade computacional do processo de treinamento do DQN pode ser estimado por $O(n_{episodios} * T * \sum_{e=0}^{E-1} u_e u_{e-1})$ [48], em que $n_{episodios}$ é o número de episódios do processo de treinamento, T é o número de eventos por episódio, E é o número de camadas da rede neural utilizada como agente e u_e é o número de neurônios da camada e da rede neural [47, 48]. Para o processo de inferência, a complexidade é de $O(\sum_{e=0}^{E-1} u_e u_{e-1})$ [47].

A partir da DQN, surgiram novas técnicas para melhorar o desempenho deste algoritmo. Entre as principais estão o *Double Q-Network* e o *Dueling Q-Network*.

Algorithm 5 DQN

Entrada: parâmetros iniciais das redes neurais do estimador θ e da memória de *replay* R
Saída: parâmetros otimizados da rede neural do estimador θ
Inicializar a rede neural do estimador $Q(s, a; \theta)$
Inicializar a rede neural $Q'(s, a; \theta')$
Sincronizar os pesos das redes *target*: $\theta' \leftarrow \theta$
Inicializar a memória de *replay* R
para cada episódio $i = 1, 2, \ldots, n_{episodios}$ faça:
    Inicializar a estratégia de exploração η
    Observar o estado inicial s_0
    para cada evento $t = 1, 2, \ldots, T$ faça:
        Selecionar a ação a_t através de $\max_a Q(s_t, a; \theta)$ e η
        Executar a ação a_t, obter recompensa r_t e transitar para o próximo estado s_{t+1}
        Adicionar a transição (s_t, a_t, r_t, s_{t+1}) à memória R
        Amostrar um *minibatch* de N transições (s_j, a_j, r_j, s_{j+1}) de R
        Atualizar θ minimizando:

$$J(\theta) = (\gamma \max_a Q'(s_{t+1}, a; \theta) + r_t) - Q(s_t, a_t; \theta)$$

        Atualizar suavemente os parâmetros das redes *target*:

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$$

    fim para
fim para

4.4.1.2 Dueling DQN

É uma técnica que consiste em separar a estimativa da função Q em duas partes: uma que estima a função valor (função V) e outra que estima a função vantagem (função A). Isso é feito utilizando a relação expressa na Equação (3.14) [57].

Na prática, isso é feito dividindo a última camada da rede neural estimadora em duas partes, uma responsável por estimar a função V e a outra por estimar a função A. Essa desagregação permite conhecer a função V para cada estado em específico, sem depender da ação realizada neste estado [57].

A partir dessa modificação, o algoritmo é capaz de aprender a estimar a função V a cada estimativa da função Q, independentemente de qual ação foi realizada no estado em questão. Esse fator faz com que o aprendizado da função valor seja mais eficiente e que mais atualizações sejam feitas ao longo do processo de aprendizagem, estabilizando a sua convergência [57].

Este algoritmo compartilha do mesmo pseudocódigo do DQN, apresentado no Algoritmo 5, diferenciando-se apenas na estrutura da rede neural para estimação da função Q. Dessa forma, este algoritmo também possui complexidade computacional igual à da DQN.

4.4.1.3 Double DQN (D2QN)

Essa técnica foi desenvolvida para diminuir uma sobreestimação da função Q para alguns estados, aumentando-se a estabilidade do processo de aprendizado do algoritmo e reduzindo-se o risco de estabilização em mínimos locais [58, 59].

A estratégia é estimar o valor Q do par estado-ação futuro da função de custo não pelo máximo estimado pelas redes *target* e sim pelo máximo estimado pela rede utilizada para coleta das experiências *online* [58]. Dessa forma, a função de custo se transforma para o que está mostrado na Equação (4.11).

$$J(\theta) = (\gamma Q'(s_{t+1}, a_{estimed}) + r_t) - Q(s_t, a_t) \tag{4.11}$$

em que

$$a_{estimed} = \max_a Q(s_{t+1}, a) \tag{4.12}$$

Essa modificação reduz sobreestimações da função Q e melhora o desempenho da rede em diversos cenários [58].

Assim como o Dueling DQN, o D2QN compartilha o mesmo pseudocódigo apresentado

no Algoritmo 5, com exceção da função de custo, que é substituída pela Equação (4.11).

Com a adição da segunda rede neural neste algoritmo, a complexidade computacional do processo de treinamento do D2QN é modificada para $O(n_{episodios} * T * 2\sum_{e=0}^{E-1} u_e u_{e-1})$ [47, 48], em que $n_{episodios}$ é o número de episódios do processo de treinamento, T é o número de eventos por episódio, E é o número de camadas da rede neural utilizada como agente e u_e é o número de neurônios da camada e da rede neural [47, 48]. Já para o processo de inferência, a complexidade é de $O(2\sum_{e=0}^{E-1} u_e u_{e-1})$ [47].

4.4.1.4 Dueling Double DQN (D3QN)

Este algoritmo é a união das duas técnicas explicadas nas seções 4.4.1.2 e 4.4.1.3, ou seja, utiliza uma rede neural desagregada para estimativa da função Q a partir das estimativas das funções A e V, além de usar a técnica para reduzir a sobreestimação utilizada no D2QN.

Dessa forma, este algoritmo se aproveita das vantagens fornecidas por cada uma dessas técnicas, unindo-as para formação de um algoritmo com melhor desempenho e estabilidade [57]. A complexidade computacional deste algoritmo é igual à do D2QN [48].

A partir dos algoritmos apresentados, desenvolveu-se uma implementação que permitisse um processo de treinamento adequado para a tarefa em questão. Tal implementação está descrita nas seções seguintes.

4.4.2 Estado do ambiente

O estado do ambiente utilizado para a definição de qual LLH seria selecionada pela HH, foi um vetor de dimensão fixa e igual a 8. O vetor foi constituído por algumas métricas do sistema que capacitam o algoritmo a entender a configuração do sistema para a sua tomada de decisão. As métricas serão explicadas abaixo, mas para fins didáticos, definiremos o que irá ser chamado de interferência para explicação dessas métricas ficar mais clara.

Na explicação abaixo, a interferência recebida por uma determinada comunicação ou sensor será referenciada como a soma dos ganhos de propagação dos canais entre os dispositivos transmissores das comunicações e sensores interferentes e o dispositivo receptor desta comunicação ou sensor. De maneira análoga, a interferência gerada por uma comunicação ou sensor é o ganho de propagação do canal entre o dispositivo transmissor desta comunicação e o dispositivo receptor das outras comunicações presentes no mesmo RB.

Apesar de esta métrica não representar exatamente a interferência recebida, dependendo de outros fatores externos à HH, como a potência de cada uma dessas comunicações e sensores, por exemplo, ela serve como um valor indicativo para o algoritmo. Dessa forma, o objetivo não é conhecer exatamente o valor de interferência sofrida ou gerada por uma

comunicação ou sensor, mas indicar à HH quais são as comunicações e sensores que provavelmente estão gerando ou recebendo mais potência interferente nos RBs do sistema.

A partir dessa conceituação, é possível detalhar as métricas usadas como estado do ambiente para treinamento das hiper-heurísticas:

1. Interferência recebida pelo sensor que recebe a maior interferência no RB atual;
2. A partir da identificação de qual é o sensor que sofre com a maior interferência no RB atual, calcula-se a interferência que este sensor receberia no RB cuja interferência seja mínima;
3. Interferência recebida pela comunicação primária gerada pelo D2D que gera mais interferência nela;
4. Interferência recebida pelos sensores gerada pelo D2D que gera mais interferência neles;
5. A partir da identificação de qual é o D2D que sofre com a maior interferência no RB atual, calcula-se esta interferência recebida para este D2D;
6. A partir da identificação de qual é o D2D que sofre com a maior interferência no RB atual, calcula-se a interferência que este D2D receberia no RB cuja interferência recebida por ele fosse mínima;
7. A partir da identificação de qual é o D2D de outros RBs que geraria a menor interferência no RB atual, calcula-se a interferência que este D2D geraria no RB atual;
8. A partir da identificação de qual é o D2D de outros RBs que receberia a menor interferência no RB atual, calcula-se a interferência que este D2D receberia no RB atual;
9. A partir da identificação de qual é o sensor de outros RBs que receberia a menor interferência no RB atual, calcula-se a interferência que este sensor receberia no RB atual;
10. Capacidade de transmissão da comunicação primária do RB atual;
11. Capacidade de transmissão do sensor que possui a menor capacidade de transmissão do RB atual.

A partir de tais métricas, é possível entender a situação geral do RB atual em que a HH está atuando, além de ter informações de outros RBs que são importantes para a decisão de enviar uma comunicação ou sensor para outro RB ou de trazer algum para o RB em questão.

4.4.3 Ação

A ação da HH é definir qual LLH seria selecionada a partir do estado atual do ambiente. Dessa forma, como foram desenvolvidas 8 LLHs, a ação da HH foi de natureza discreta, selecionando um número inteiro de 0 a 7, em que cada um desses possíveis números era o índice de uma das LLHs disponíveis.

4.4.4 Recompensa

A função de recompensa utilizada foi baseada na mesma função usada no Capítulo 3, detalhada na Equação 3.19 da Seção 3.4.3. Para o problema de alocação do espectro, optou-se por utilizar a diferença entre a função recompensa antes e depois de se executar a ação definida. A Equação 4.13 mostra tal função:

$$r_{SA} = 2 * \omega_1 \frac{r_{PC}^{new} - r_{PC}^{old}}{r_{PC}^{new} + r_{PC}^{old}} \tag{4.13}$$

em que r_{PC}^{old} é a função de recompensa detalhada na Equação 3.19 antes da execução da ação definida pela HH e r_{PC}^{new} é a função de recompensa após a execução da ação definida pela HH, ω_1 é um fator de ajuste do valor da função de recompensa.

Essa escolha foi baseada em testes empíricos que mostraram resultados melhores do que usando apenas a Equação 4.13. Um possível motivo para isso é o fato de uma LLH não modificar a alocação do espectro completamente de uma vez, apenas uma parte dela. Dessa forma, ao utilizar uma função de recompensa baseada na diferença entre o resultado antes e depois de definir a LLH que será utilizada, é possível identificar o efeito da ação de forma mais destacada.

4.4.5 Execução das simulações

As simulações executadas para treinamento e teste dos algoritmos seguiram basicamente as mesmas configurações detalhadas na Seção 3.4.4, com pequenas diferenças que serão detalhadas a seguir.

Foram utilizados sistemas com $K \in [5, 10, 15, 20]$ RBs para treinamento e teste dos algoritmos. O número de comunicações D2D e de sensores em cada episódio continuou sendo determinado a partir de uma distribuição de Poisson, mas os valores de taxa de ocorrência variavam com o número de RBs.

Para manter a mesma proporção utilizada no Capítulo 3, o número de comunicações primárias foi igual ao número de RBs (K), o número de D2Ds seguiu uma Poisson$(4K)$, enquanto o de sensores seguiu uma Poisson$(2K)$.

Nessas simulações, a HH era chamada para atuar em todos os RBs e definir a nova alocação do espectro. A partir dessa definição, cada RB foi separadamente enviado ao algoritmo responsável pelo controle de potências, neste caso uma instância do PPO resultante do treinamento detalhado no Capítulo 3. Após o controle de potências de cada RB, a nova alocação de recursos era direcionado ao sistema para a devida atualização.

4.4.6 Configuração dos algoritmos

A partir das definições do ambiente, definiu-se a configuração dos algoritmos a partir de testes empíricos.

4.4.6.1 Configuração das redes neurais

As redes neurais mantiveram basicamente a mesma estrutura detalhada na Seção 3.4.5.1. Para o PPO, a estrutura foi exatamente a mesma, apenas com mudança na camada de saída, já que a dimensionalidade muda de um problema para o outro.

Já para os modelos baseados na DQN, que não possuem estrutura *actor-critic*, a estrutura da rede neural estimadora da função Q foi idêntica à da rede neural do agente do PPO, com a única diferença sendo a camada de saída.

Como esses modelos baseados na DQN são *value-based* eles estimam a função Q para cada uma das possíveis ações a serem realizadas. Dessa forma, apesar de a ação escolhida por RB ser única, a camada de saída das redes neurais tem dimensão igual ao espaço de ações, neste caso, igual a 8, já que este é o número de LLHs disponíveis para atuação.

4.4.6.2 Parametrização dos algoritmos

Assim como no Capítulo 3, cada algoritmo foi parametrizado segundo suas especificidades a partir de testes realizados com diferentes valores. A partir disso, a configuração alcançada que obteve maior valor de retorno ao final do processo de treinamento está detalhada na Tabela 4.1.

A partir das configurações detalhadas acima, o processo de treinamento foi executado para cada algoritmo e os resultados estão detalhados a seguir.

Tabela 4.1 – Parametrização dos Algoritmos PPO, DQN, Dueling DQN, D2QN e D3QN

Parâmetro	**PPO**	**DQN**	**Dueling DQN**	**D2QN**	**D3QN**
Episódios de treinamento	1000	1000	1000	1000	1000
Batch size	-	64	64	64	64
Fator de desconto do retorno (γ)	0.99	0.90	0.90	0.90	0.90
Intervalo de sincronização das redes *target*	-	5	5	5	5
Tamanho da memória de *replay*	-	10000	10000	10000	10000
Fator de suavização da atualização (τ)	1	0.10	0.05	0.05	0.05
Probabilidade inicial de escolha de ação de exploração	-	0.9	0.9	0.9	0.9
Probabilidade final de escolha de ação de exploração	-	0	0	0	0
Intervalo de decaimento exponencial da probabilidade de exploração	-	100	100	100	100
Otimizadores das redes neurais do agente	Adam	-	-	-	-
Taxa de aprendizagem do agente	0.0001	-	-	-	-
Otimizadores das redes neurais do crítico/estimador	Adam	Adam	Adam	Adam	Adam
Taxa de aprendizagem do crítico/estimador	0.001	0.0001	0.0001	0.0001	0.0001
Intervalo para atualização do agente (T_{update})	100	-	-	-	-
Épocas de iteração para treinamento das redes (K)	30	-	-	-	-
Limitantes da função de perda do agente (ϵ)	0.2	-	-	-	-

4.5 RESULTADOS

O processo de treinamento foi executado seguindo as parametrizações apresentadas nas seções anteriores e os seus resultados serão apresentados a seguir.

4.5.1 Processo de treinamento

Assim como no Capítulo 3, foram treinadas 5 instâncias de cada algoritmo de DRL utilizando as configurações apresentadas na Seção 4.4. As diferentes instâncias servem para atestar que a convergência do processo de treinamento é estável para diferentes ocorrências.

A partir dessas instâncias, avaliaram-se a média e o desvio padrão de diferentes métricas do sistema, a fim de mensurar as vantagens que cada algoritmo apresentou em seus diferentes processos de convergência. Tanto a média quanto o desvio padrão foram computados nos últimos 200 episódios do processo de treinamento de cada algoritmo.

Além dos algoritmos de DRL, também utilizaram-se dois algoritmos para comparação, relativos à AE (Alocação de Espectro) e ao CP (Controle de Potência):

- AE aleatório: modelo em que a alocação do espectro é feita de forma aleatória, escolhendo-se uma das LLHs desenvolvidas de forma aleatória, em que cada LLH possui a mesma probabilidade de ser escolhida. Nesse modelo, o controle de potência é feito por uma rede neural treinada baseado no que foi apresentado no Capítulo 3, assim como o que foi feito para os algoritmos de DRL;
- AE e CP aleatórios: modelo em que tanto a alocação do espectro é aleatória (seguindo o que foi apresentado no AE aleatório), quanto o controle de potência também é randômico, em que a potência escolhida segue uma distribuição uniforme igual à utilizada para comparação no Capítulo 3.

As métricas usadas para avaliação dos algoritmos estão dispostas na Tabela 4.2.

A Tabela 4.2 mostra que três algoritmos alcançaram taxas de *outage* das comunicações primárias menores do que o AE aleatório: o DDQN, o D3QN e o PPO. Nesta métrica, o DDQN alcançou o melhor valor (13.23%), seguido pelo PPO por apenas 0.1% de diferença. A DQN e a D2QN alcançaram valores mais altos do que o AE aleatório.

Além disso, é possível observar que todos os algoritmos de DRL alcançaram valores menores de taxa de *outage* dos sensores do que o AE aleatório. Nesta métrica, o PPO alcançou o melhor valor, igual a 3.33%, mais do que 5 vezes menor do que o obtido pelo AE aleatório. Após o PPO, o segundo melhor algoritmo nesse quesito foi o D3QN, com taxa de *outage* de 3.56%.

Tabela 4.2 – Resultados dos algoritmos de alocação de espectro

Variável		**AE aleatório**	**AE e CP aleatório**	**DQN**	**DDQN**	**D2QN**	**D3QN**	**PPO**
Outage da comm. primárias (%)	média	14.24	65.75	17.80	**13.23**	14.66	14.01	13.33
	DP	-	-	0.84	2.25	2.70	2.37	3.16
Outage do sensor (%)	média	18.71	48.13	6.32	6.46	10.05	3.56	**3.33**
	DP	-	-	1.28	4.91	12.33	1.25	1.47
SNIR das comm. primárias (dB)	média	11.02	-6.32	9.88	10.46	10.44	10.47	10.62
	DP	-	-	0.17	0.37	0.15	0.31	0.49
SNIR dos D2Ds (dB)	média	-11.15	-24.33	-11.32	-11.37	**-10.98**	-11.25	-11.16
	DP	-	-	0.06	0.32	0.16	0.29	0.17
SNIR dos sensores (dB)	média	-5.28	-11.71	2.93	1.86	0.28	4.44	4.51
	DP	-	-	1.85	6.53	9.88	1.56	1.70

Além das taxas de *outage*, um dos objetivos do problema é maximizar a SNIR dos D2Ds. Para avaliar essa métrica, é possível observar que todos os algoritmos de DRL tiveram valores similares aos obtidos pelo AE aleatório, sendo o D2QN o melhor nesse quesito, com valor ligeiramente menor (-10.98 dB), seguido pelo PPO, com -11.16 dB.

É possível perceber que os algoritmos de DRL utilizados para alocação do espectro foram capazes de minimizar a taxa de *outage* dos sensores de forma significativa, reduzindo-se em quase 6 vezes o valor obtido pelo AE aleatório. Além disso, obteve-se uma redução da taxa de *outage* das comunicações primárias, sem prejuízo à SNIR das comunicações D2D.

A partir dos resultados apresentados, o PPO foi o escolhido como algoritmo proposto, uma vez que foi o que obteve menor taxa de *outage* dos sensores, além de ter ficado com o segundo lugar na avaliação das métricas de taxa de *outage* das comunicações primárias e de SNIR dos D2Ds.

Ao se analisarem os algoritmos de DRL para alocação do espectro com o AE aleatório, é possível isolar o efeito do controle de potência proposto no Capítulo 3 e verificar que existe ganho ao se utilizar desses algoritmos nas duas etapas da alocação de recursos do sistema.

Entretanto, ao se compararem os resultados obtidos na Tabela 4.2 com os resultados do AE e CP aleatórios, é possível identificar os reais ganhos da proposta deste trabalho. Ao combinar-se as duas estratégias para alocação de espectro e controle de potência utilizando-se instâncias treinadas do PPO, foi possível, se comparado ao AE e CP aleatório:

- Reduzir a taxa de *outage* das comunicações primárias em quase 5 vezes, de 65.75% para 13.33%;
- Reduzir a taxa de *outage* dos sensores em mais de 14 vezes, passando de 48.13% para 3.33%;
- Aumentar a SNIR dos D2Ds para mais que o dobro, passando de -24.33 dB para -11.16 dB, comparando em valores absolutos, um ganho de mais de 1974%.

Dessa forma, a Tabela 3.3 e a Tabela 4.2 mostram que existem ganhos consideráveis na utilização de algoritmos de DRL tanto para o controle de potência quanto para a alocação de espectro. Tais ganhos são aumentados combinando-se as duas estratégias, que é a proposta deste trabalho.

Além dos valores mostrados, é importante avaliar a evolução dessas variáveis ao longo do processo de treinamento. A Figura 4.4 ilustra a evolução da taxa de *outage* das comunicações primárias ao longo do processo de treinamento.

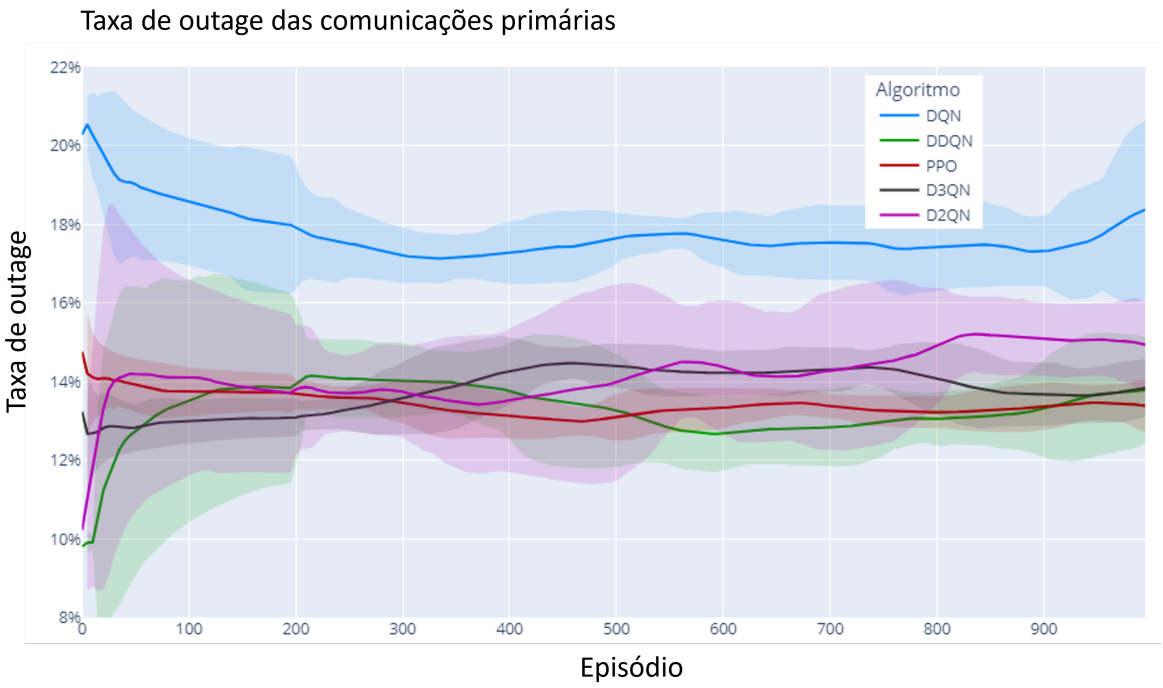


Figura 4.4 – Taxa de *outage* das comunicações primárias ao longo do processo de treinamento dos algoritmos. As linhas representam o valor médio, enquanto a região sombreada é o intervalo de confiança referente à linha de mesma cor. Fonte: autoria própria.

Através da curva, é possível perceber que, com exceção da DQN, todas os algoritmos treinados foram capazes de convergir para valores entre 13% e 15%. Além disso, é perceptível que a dispersão dos resultados obtidos diminui ao longo do processo para todos os algoritmos.

Os algoritmos que alcançaram os menores valores médios de taxa de *outage* ao final do processo de treinamento foram o PPO, o D3QN e o D2QN, com intervalo de confiança controlado.

Esse gráfico mostra que os algoritmos criados a partir de evoluções do DQN conseguiram alcançar valores de taxa de *outage* menores do que o DQN, indicando que as modificações implementadas foram importantes para o processo de treinamento no contexto em questão.

Além da taxa de *outage* das comunicações primárias, a taxa de *outage* dos sensores também foram coletados. A Figura 4.5 ilustra a evolução da taxa de *outage* dos sensores ao longo do processo de treinamento.

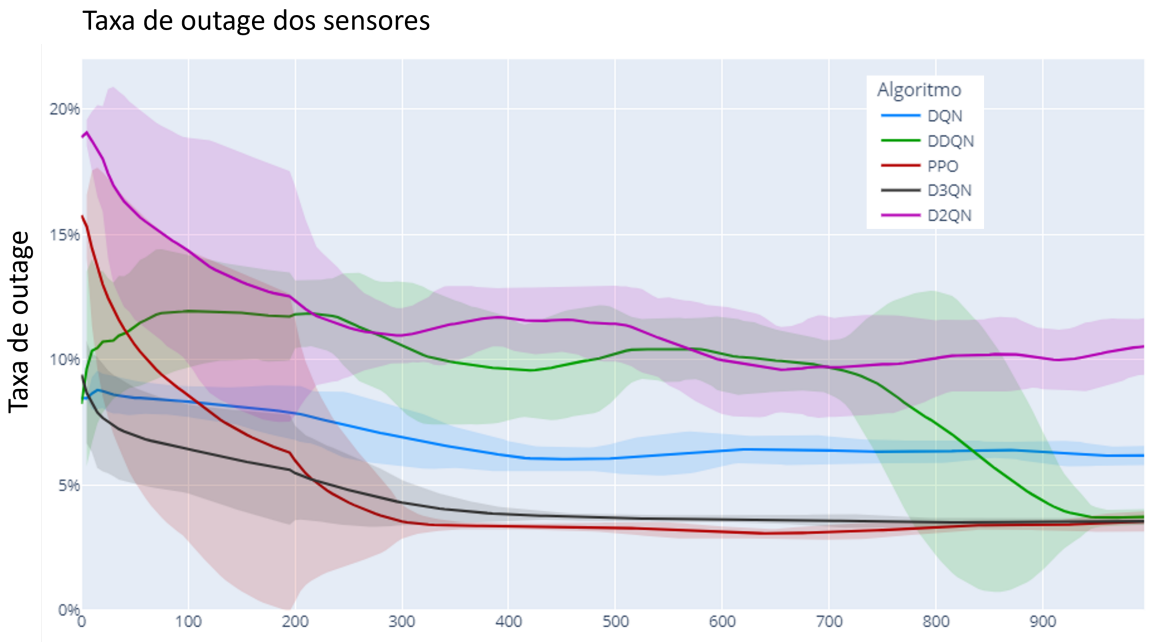


Figura 4.5 – Taxa de *outage* dos sensores ao longo do processo de treinamento dos algoritmos. As linhas representam o valor médio, enquanto a região sombreada é o intervalo de confiança referente à linha de mesma cor. Fonte: autoria própria.

A partir da Figura 4.5, é possível perceber que todos os algoritmos testados também convergiram para valores médios menores do que os obtidos no início do processo de treinamento, assim como é perceptível a redução da dispersão ao longo do tempo.

O gráfico também mostra que, ao final do processo de treinamento, o PPO, o D3QN e o D2QN obtiveram menores valores médios de taxa de *outage* dos sensores do que os demais. Novamente o DQN apresenta uma convergência para um mínimo local com resultados piores do que os obtidos pelos algoritmos citados.

Por outro lado, o D2QN, um dos algoritmos com melhores valores de taxa de *outage* das comunicações primárias, não consegue ser competitivo com o D3QN e nem com o PPO na métrica de taxa de *outage* dos sensores.

Ao focarmos nos 3 algoritmos com melhores desempenhos nesta métrica (o PPO, o D3QN e o D2QN), observa-se que os dois primeiros desenvolvem uma política capaz de reduzir a taxa de *outage* dos sensores próximo ao episódio 400, mantendo esses valores até o final do processo de treinamento com baixa dispersão.

Por outro lado, o D2QN reduz o valor médio para o mesmo patamar dos demais apenas após o episódio 700, com dispersão mais alta do que do PPO e do D3QN. Esses fatores indicam que o PPO e o D3QN conseguem aprender mais rapidamente e com mais estabilidade do que os demais algoritmos.

A partir dos resultados apresentados, é possível concluir que o PPO e o D3QN foram os

algoritmos que melhor conseguiram desenvolver políticas que protegessem tanto as comunicações primárias quanto os sensores, os objetivos primários definidos para os algoritmos.

Além das taxas de *outage*, a Figura 4.6 expõe a evolução da SNIR das comunicações D2D, o objetivo secundário para os algoritmos de alocação do espectro.

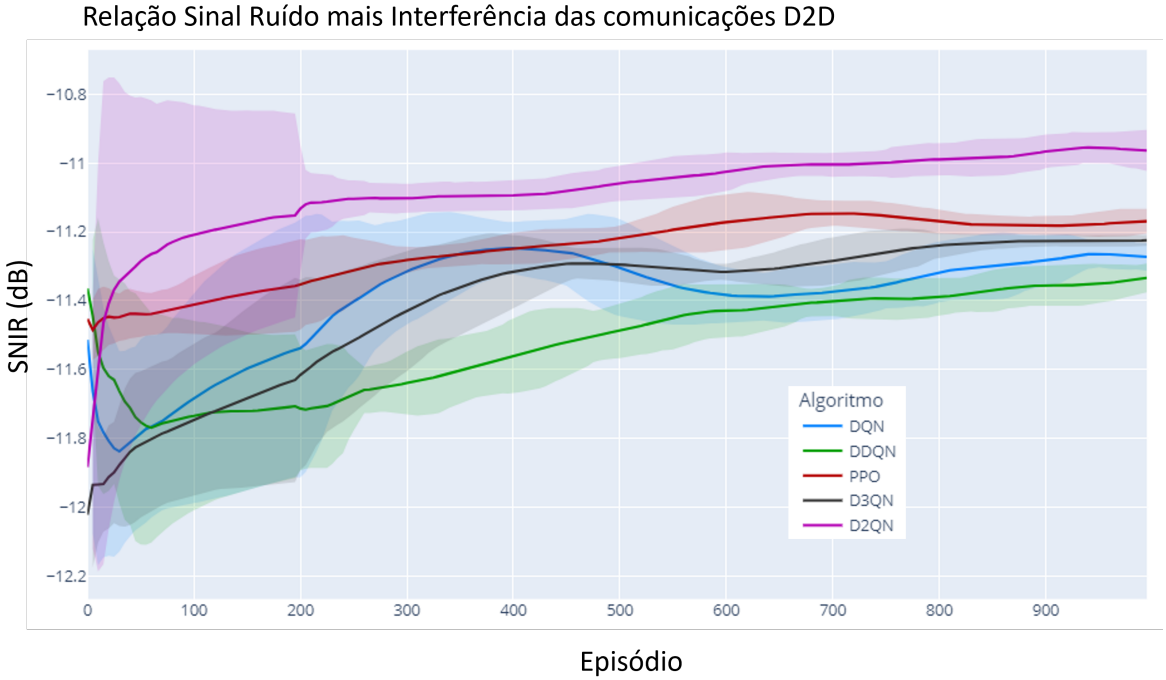


Figura 4.6 – SNIR das comunicações D2D ao longo do processo de treinamento dos algoritmos. As linhas representam o valor médio, enquanto a região sombreada é o intervalo de confiança referente à linha de mesma cor. Fonte: autoria própria.

Assim como os demais gráficos, a Figura 4.6 indica que todos os algoritmos testados convergiram ao longo do processo de treinamento, elevando-se a média da SNIR das comunicações D2D e reduzindo-se a dispersão ao longo dos episódios.

O algoritmo que obteve a melhor média ao final do processo de treinamento nessa métrica foi o D2QN, seguido pelo PPO e pelo D3QN, todos também com baixa dispersão ao final do processo.

A partir da análise dos resultados, é possível perceber que o D2QN consegue valores médios de SNIR maiores do que o PPO e o D3QN, mas a custa de uma maior taxa de *outage* dos sensores.

Dessa forma, é possível perceber que o PPO e o D3QN foram os algoritmos com melhor desempenho na alocação do espectro, já que estes algoritmos foram capazes de controlar melhor as taxas de *outage*, que é o objetivo primário dos algoritmos, além de obterem o 2º e 3º maior valor de SNIR das comunicações D2D.

Nos resultados obtidos, o PPO teve um desempenho ligeiramente melhor do que o D3QN

em todas as métricas avaliadas, mas em todas elas a diferença de desempenho foi pequena.

4.5.2 Análise em diferentes sistemas

A partir dos resultados apresentados, testaram-se o PPO e o D3QN em sistemas com diferentes quantidades de RB, mais especificamente em sistemas com 5, 10, 15 e 20 RBs. Os resultados relacionados à taxa de *outage* das comunicações primárias estão expostos na Figura 4.7.

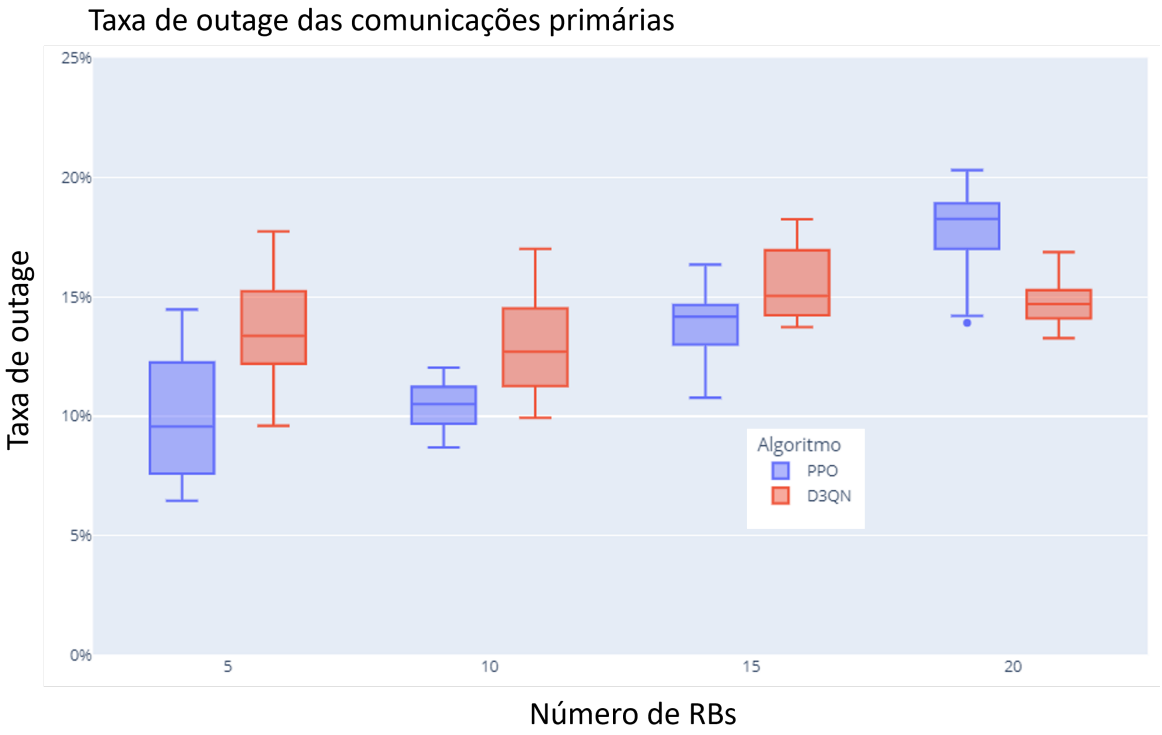


Figura 4.7 – Gráfico de caixa com as distribuições da taxa de *outage* das comunicações primárias em função do número de RBs presentes nos sistemas testados. Fonte: autoria própria.

Assim como observado nos resultados da seção anterior, o PPO obtém taxas de *outage* ligeiramente menores do que as obtidas pelo D3QN para sistemas com 5, 10 e 15 RBs. Entretanto, para sistemas com 20 RBs, é possível observar uma inversão deste comportamento, já que a distribuição dos valores obtidos pelo D3QN possui mediana consideravelmente inferior, além de menor dispersão.

Além disso, é perceptível a existência de uma tendência de aumento da taxa de *outage* à medida que o número de RBs aumenta, mas que tal aumento é pequeno. Isso é percebido ao se observar que a mediana do melhor algoritmo nessa métrica em um sistema com 5 RBs está próxima a 10%, ao passo que para um sistema com 20 RBs este valor está abaixo de 15%.

Dessa forma, apesar de a complexidade do problema aumentar em consequência do aumento da quantidade de RBs do sistema, que também significa um aumento da quantidade de comunicações primárias, sensores e comunicações D2D, os algoritmos apresentados não tiveram grande degradação de seu desempenho, mostrando capacidade de atuarem em sistemas cujas dimensionalidades são consideravelmente diferentes.

Com respeito à taxa de *outage* dos sensores, os resultados dos testes em sistemas com diferentes números de RBs também foram coletados e estão expostos na Figura 4.8.

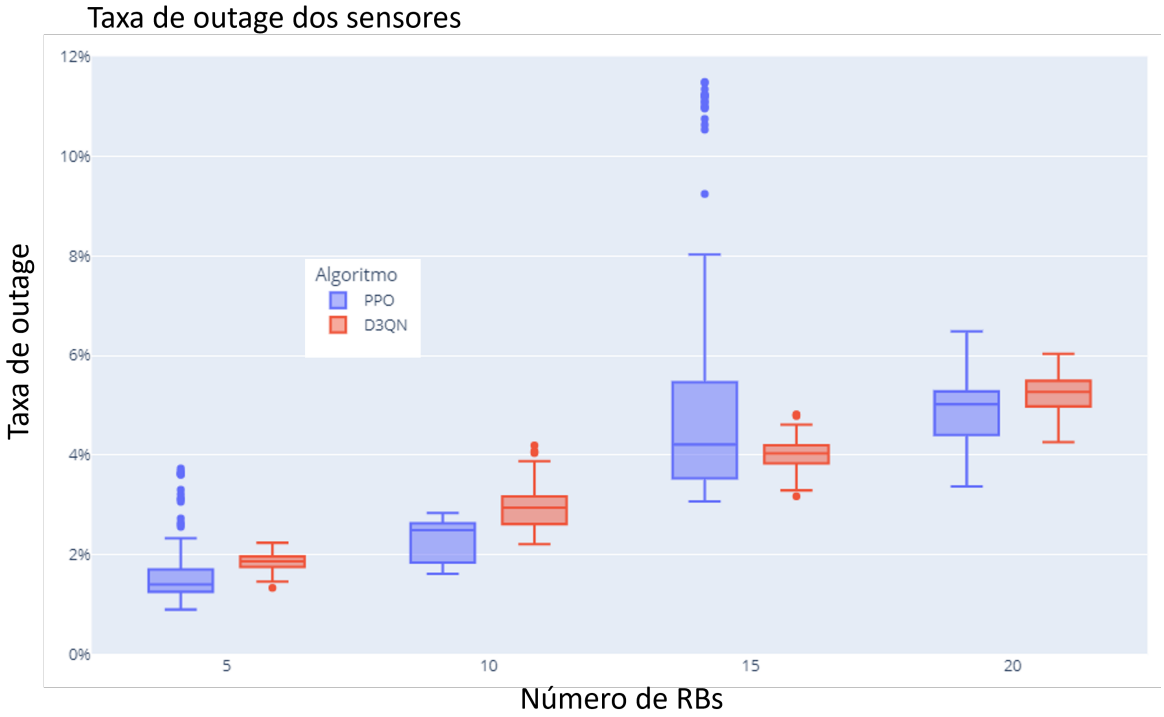


Figura 4.8 – Gráfico de caixa com as distribuições da taxa de *outage* dos sensores em função do número de RBs presentes nos sistemas testados. Fonte: autoria própria.

O comportamento para a taxa de *outage* dos sensores é similar, com as distribuições dos valores obtidos pelo PPO tendo mediana ligeiramente menor do que as obtidas pelo D3QN em quase todos os sistemas. Entretanto, neste caso, não vemos a inversão deste comportamento em sistemas com 20 RBs.

Além disso, o gráfico também mostra um aumento da taxa de *outage* dos sensores à medida que o número de RBs do sistema aumenta, mas tal aumento também não é muito grande, ficando abaixo de 6% em quase todos os casos.

Além disso, percebe-se que o PPO apresentou dispersões maiores nesta métrica do que o D3QN, principalmente para sistemas com 15 RBs. Isso provavelmente indica que as políticas desenvolvidas pelo PPO são um pouco mais instáveis às aleatoriedades do processo de treinamento.

Por fim, obtiveram-se os valores da SNIR das comunicações D2D para os mesmos testes, cujos resultados estão dispostos na Figura 4.9.

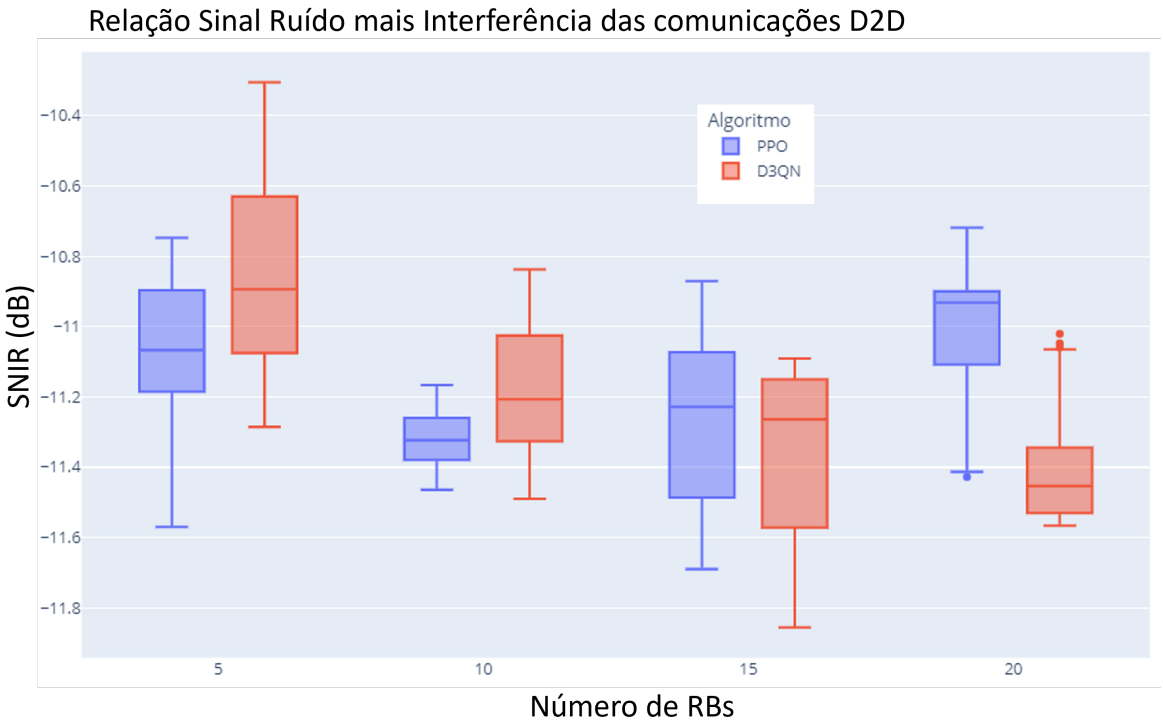


Figura 4.9 – Gráfico de caixa com as distribuições da SNIR dos D2Ds em função do número de RBs presentes nos sistemas testados. Fonte: autoria própria.

Neste gráfico, o comportamento difere dos outros dois mostrados anteriormente nesta seção. As distribuições dos valores de SNIR das comunicações D2D obtidas pelo PPO não apresentam tendência clara de aumento como visto nas taxas de *outage*, tanto da comunicação primária quanto dos sensores.

Tal comportamento indica que a natureza oportunística da comunicação D2D, isto é, o fato de o algoritmo aprender a se preocupar primariamente com as taxas de *outage* do sistema, faz com que os valores de SNIR obtidos pelo PPO sejam mais variáveis à política desenvolvida ao longo no processo de treinamento, não sendo necessariamente um fator relacionado à complexidade do problema abordado.

Por outro lado, os valores obtidos pelo D3QN mostram uma tendência clara de queda à medida que o númerode RBs do sistema aumenta. Isto indica que as políticas desenvolvidas por este algoritmo de fato são dependentes da dimensionalidade do problema.

Por fim, percebe-se que as SNIRs obtidas pelo D3QN são ligeiramente maiores do que as obtidas pelo PPO em sistemas com 5 e 10 RBs, mas que tal comportamento se inverte para 15 e 20 RBs.

A partir dos resultados apresentados, é possível perceber que não há grande diferença

entre os resultados obtidos pelo PPO e pelo D3QN, de forma que ambas são opções para a realização da alocação do espectro. Entretanto, o PPO obteve valores médios de taxa de *outage* das comunicações primárias e dos sensores menores do que os obtidos pelo D3QN, além de valores mais altos de SNIR das comunicações D2D.

Além disso, analisando as distribuições testadas em sistemas com diferentes configurações, o PPO também demonstrou robustez ao aumento da dimensionalidade do problema, com comportamentos diferentes dos mostrados pelo PPO, mas sem grande degradação de desempenho.

Por esses motivos, a proposta do trabalho é a utilização do PPO para alocação do espectro, assim como para o controle de potência, como mostrado no Capítulo 3. Tal resultado demonstra a capacidade de atuação desse algoritmo de DRL em superar algoritmos modernos da área tanto em problemas com espaço de ações discreto quanto contínuo.

4.6 CONCLUSÃO

Este capítulo apresentou a proposta de alocação de espectro do trabalho, unindo o algoritmo de controle de potências apresentado no Capítulo 3 com a estratégia de alocar o espectro utilizando uma Hiper-heurística inteligente baseada em algoritmos de DRL.

A proposta conjunta do trabalho para alocação do espectro e controle de potências utilizando algoritmos de DRL se mostrou capaz de proteger as comunicações primárias e o sensoreamento de forma muito mais eficiente do que os algoritmos utilizados para comparação de resultados. Se comparados com uma alocação completamente aleatória, o algoritmo proposto foi capaz de reduzir a taxa média de *outage* das comunicações primárias de 65.75% para 13.33%, além da reduzirem a taxa média de *outage* dos sensores de 48.13% para 3.33%.

Além da proteção às comunicações primárias e aos sensores, o algoritmo proposto foi capaz de aumentar a SNIR média das comunicações D2D, que era o objetivo secundário do algoritmo, já que tais comunicações foram tratadas com caráter oportunístico. Se comparado com a mesma alocação aleatória, o algoritmo proposto aumentou a SNIR das comunicações D2D de -24.33 dB para -11.16 dB, um aumento de magnitude da SNIR de 54%.

Dessa forma, a união do algoritmo para controle de potências, apresentado no Capítulo 3, com o algoritmo para alocação do espectro apresentado neste capítulo gerou uma alocação de recursos capaz de reduzir drasticamente a taxa de *outage* tanto das comunicações primárias quanto dos sensores, além de aumentar consideravelmente a SNIR das comunicações D2D.

Entretanto, a fim de isolar o efeito produzido pelo controle de potências inteligente feito pelo algoritmo desenvolvido no Capítulo 3, os resultados também foram comparados com um algoritmo cuja alocação do espectro era aleatória, mas o controle de potências era feito

pela mesma rede neural utilizada pelo algoritmo proposto.

Comparando com este algoritmo, o algoritmo proposto ainda mostra resultados superiores, identificando os ganhos do algoritmo de alocação do espectro de forma isolada. O algoritmo proposto foi capaz de reduzir a taxa média de *outage* de 14.24% para 13.33%, uma redução percentual de 6.4% nesta métrica. Além disso, o ganho na taxa média de *outage* dos sensores é mais representativo, passando de 18.71% para 3.33%, uma redução percentual de 82.2%. Por fim, não houve ganho ou perda representativa na SNIR média das comunicações D2D, passando de -11.15 dB para -11.16 dB.

Esses valores indicam que a alocação do espectro do algoritmo proposto conseguiu proteger o sensoreamento de forma muito mais eficiente do que os algoritmos de comparação, mesmo na situação em que o controle de potência era feito de forma inteligente por ambos os algoritmos. Além disso, o algoritmo proposto conseguiu reduzir um pouco mais a taxa de *outage* das comunicações primárias sem prejuízo significativo para a SNIR das comunicações D2D.

Além disso, os resultados indicaram que o PPO obteve o melhor desempenho, se comparado aos demais algoritmos testados, mas a diferença para o D3QN foi sutil, indicando que para essa aplicação, ambos os algoritmos podem ser considerados.

Por fim, a análise de desempenho em sistemas com diferentes números de RBs, indicou robustez do algoritmo em problemas com diferentes dimensionalidades, ou seja, o desempenho do algoritmo foi pouco afetado pelo aumento da complexidade do problema [14].

5 CONCLUSÃO E TRABALHOS FUTUROS

5.1 CONCLUSÃO

A revolução industrial conhecida como Indústria 4.0 é marcada pela digitalização e automação de processos produtivos e pelo intercâmbio de informações relevantes entre as tecnologias envolvidas em tal processo, tais como automação, controle e tecnologia da informação.

Neste cenário, as tecnologias de comunicação móvel 5G e, futuramente, 6G desempenham funções essenciais, especialmente o 6G, que deverá oferecer altas taxas de transmissão de dados, latência ultra-baixa e integração com técnicas de Inteligência Artificial, permitindo o surgimento de novas oportunidades de desenvolvimento de sistemas de produção inteligentes para a Indústria 4.0 e para a Indústria 5.0 [3, 4, 5].

No contexto da Indústria 4.0, as tecnologias de sensoreamento emergem como elementos cruciais, facilitando a coleta e análise dos dados em tempo real por meio de sensores e permitindo que os sistemas automatizados reajam de maneira adaptativa e inteligente às variações ambientais [8]. Em paralelo, a integração entre as tecnologias de sensoreamento e dos sistemas de comunicação móvel, particularmente com as inovações trazidas pelas redes 5G e 6G, é um desenvolvimento significativo por promover uma utilização mais eficaz do espectro e a otimização dos recursos do sistema e não apenas melhorar a comunicação entre dispositivos na Indústria 4.0, mas também elevar a capacidade de monitorar e controlar os ambientes de produção [9].

Neste trabalho, o foco na alocação de espectro emerge como um tema central devido aos desafios inerentes à implementação de sistemas conjuntos de comunicação e sensoreamento na Indústria 4.0. Tais sistemas introduzem a complexidade de gerenciar a interferência entre as transmissões de sinais distintos. A gestão é crítica para garantir a operação eficaz de ambas as funcionalidades dentro do mesmo espectro, exigindo estratégias avançadas de alocação de recursos para otimizar o desempenho e a eficiência espectral [9].

A eficiência na alocação de recursos limitados, como o espectro e a energia, é vital para o sucesso de sistemas que integram diferentes tipos de comunicações, como as comunicações D2D, além do sensoreamento. A coordenação eficaz desses recursos é necessária devido ao grande volume de dispositivos interconectados e à diversidade de funções que devem ser

sustentadas simultaneamente. Este manejo do espectro e da energia visa minimizar a interferência cruzada, permitindo que as operações de comunicação e de sensoreamento ocorram sem comprometimento mútuo [10].

Adicionalmente, a complexidade aumenta em contextos onde as comunicações D2D operam no modo *underlay*, compartilhando o espectro de maneira oportunística entre comunicações primárias e atividades de sensoreamento. Este cenário exige uma abordagem ainda mais refinada na alocação de recursos, pois é necessário dividir os recursos de forma equilibrada entre múltiplos usuários com diferentes necessidades. A abordagem adotada deve garantir que não somente a eficiência espectral seja maximizada, mas também que haja uma harmonia na coexistência e na funcionalidade conjunta desses sistemas [10].

Dessa forma, este trabalho propôs um algoritmo para solucionar os desafios indicados ao implementar dois algoritmos em conjunto para a realização da alocação de recursos em um sistema com as características citadas nos capítulos anteriores. A proposta envolveu a separação do problema de alocação de recursos em dois subproblemas: controle de potência e alocação de espectro. A partir dessa separação, desenvolveu-se um algoritmo para a solução de cada subproblema.

Para o controle de potência, foram desenvolvidas redes neurais treinadas utilizando-se técnicas de DRL para decisão da potência utilizada pelas comunicações e sensores de um RB. Para isso, testaram-se algoritmos de DRL no estado da arte, entre eles o PPO e o TD3.

A rede neural desenvolvida a partir do PPO foi a que obteve melhor desempenho, em comparação com os demais algoritmos. O algoritmo desenvolvido foi capaz de proteger as comunicações primárias e os sensores, controlando as taxas de *outage* de ambos, sem grande prejuízo às taxas de comunicação das comunicações D2D.

Para a alocação de espectro, utilizou-se uma hiper-heurística de seleção. Para o desenvolvimento dessa HH, geraram-se LLHs diversas, a partir de regras lógicas do problema e, com base nelas, desenvolveu-se uma rede neural para selecionar tais LLHs a partir do estado do sistema. Para o treinamento dessas redes neurais, também foram utilizadas técnicas de DRL considerando um contexto de decisões relativas a ações de natureza discreta. Dessa forma, para este subproblema, foram testados outros algoritmos no estado da arte, entre eles o D3QN.

Em comparação com uma alocação de recursos desprovida de aspectos inteligentes, o algoritmo proposto foi capaz de reduzir a taxa média de *outage* das comunicações primárias de 64.4% para 11.8%, e de reduzir a taxa média de *outage* dos sensores de 38.5% para 4.4%. Além disso, o algoritmo aumentou a média da SNIR das comunicações D2D de -25.6 dB para -7.5 dB.

Para a alocação de espectro, a hiper-heurística que alcançou os melhores resultados foi, também, a rede neural treinada a partir do PPO, mas com uma diferença pequena para o

D3QN. O modelo desenvolvido para a alocação de espectro foi capaz de reduzir ainda mais as taxas médias de *outage* das comunicações primárias e dos sensores.

Utilizando o modelo desenvolvido para controle de potências, mas comparando com uma alocação de espectro aleatória, o algoritmo proposto foi capaz de reduzir a taxa média de *outage* das comunicações primárias de 14.2% para 13.3% e reduzir a taxa média de *outage* dos sensores de 18.7% para 3.3%, mantendo a SNIR médias das comunicações D2D em valores próximos entre si.

Quando ambos os algoritmos são unidos para a realização completa da alocação de recursos, os ganhos são ainda maiores. Comparando com um controlador de potências e um alocador de espectro desprovidos de mecanismos de aprendizagem, o conjunto dos dois algoritmos propostos foi capaz de reduzir a taxa média de *outage* das comunicações primárias de 65.8% para 13.3% e reduzir a taxa média de *outage* dos sensores de 48.1% para 3.3%, além de aumentar o valor da média da SNIR das comunicações D2D de -24.3 dB para -11.2 dB, comparando em valores absolutos, um ganho de mais de 1974%.

Os algoritmos foram desenvolvidos e testados em ambientes criados computacionalmente, simulando um sistema de comunicações móveis cujas quantidades de comunicações e de sensores variaram, assim como a quantidade de RBs disponíveis para alocação. Dessa forma, os algoritmos foram desenvolvidos para conseguir atuar em um ambiente dinâmico, cujo estado muda constantemente ao longo do tempo. Outras propostas da literatura não apresentam tal característica, exigindo o desenvolvimento de um algoritmo para cada configuração possível de ambiente [14].

Além disso, os resultados obtidos mostraram que a solução proposta foi capaz de atuar com bom desempenho em todos esses sistemas e com diferentes configurações. Em sua atuação, mostrou robustez à mudança da dimensionalidade do problema em questão, um problema encontrado em trabalhos anteriores [14].

No conhecimento do autor, este é o primeiro trabalho que propõe um algoritmo para a alocação de recursos em sistemas com comunicações primárias, D2Ds e sensores, respeitando as respectivas necessidades de cada uma delas. Em adição a isso, no conhecimento do autor, este também é o primeiro algoritmo a utilizar hiper-heurísticas no contexto de alocação de recursos de comunicações móveis.

Além disso, o trabalho em questão testou mais de oito algoritmos de DRL, seja para controle de potência do sistema, seja como HH para seleção das LLHs de alocação de espectro, apresentando uma comparação ampla entre os algoritmos dessa área, detalhando os resultados obtidos em cada um dos subproblemas.

Dessa forma, entende-se que o trabalho apresenta uma proposta inovadora, que trata de um problema cuja solução ainda não foi abordada na literatura, utilizando-se de técnicas de *Machine Learning* no estado da arte, cuja utilização neste contexto também não foi

encontrada em outros trabalhos científicos. Tal proposta mostrou ganhos significativos no desempenho do sistema de comunicações modelado, além de apresentar versatilidade para ser utilizada em diferentes contextos.

5.2 TRABALHOS FUTUROS

Como possíveis propostas para a evolução do trabalho, sugere-se:

- Evolução do sistema de comunicações modelado, para um sistema com mais de uma célula;
- Comparação da solução proposta com outros algoritmos que possam vir a ser encontrados na literatura;
- Estudo e construção de *frameworks* que possam ser utilizados para aplicação do algoritmo em ambiente produtivo;
- Modelagem de um cenário de multi-serviços providos por um sistema de comunicações, de forma que o algoritmo precise atender a requisitos específicos de diferentes serviços;
- Avaliação das soluções propostas em cenários com conhecimento imperfeito da Informação do Estado do Canal (CSI);
- Extensão do trabalho para considerar cenários de emprego de *Flexible Antenna Systems* (FAS) e *Flexible Antenna Multiple Access* (FAMA).

REFERÊNCIAS BIBLIOGRÁFICAS

1 DUAN, L.; XU, L. Data analytics in industry 4.0: A survey. *Information Systems Frontiers*, p. 1–17, 08 2021.

2 RAO, S.; PRASAD, R. Impact of 5g technologies on industry 4.0. *Wireless Personal Communications*, v. 100, p. 1–15, 05 2018.

3 VISWANATHAN, H.; MOGENSEN, P. E. Communications in the 6g era. *IEEE Access*, v. 8, p. 57063–57074, 2020.

4 LETAIEF, K. B. et al. The roadmap to 6g: Ai empowered wireless networks. *IEEE Communications Magazine*, v. 57, n. 8, p. 84–90, 2019.

5 STRINATI, E. C. et al. 6g: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication. *IEEE Vehicular Technology Magazine*, v. 14, n. 3, p. 42–50, 2019.

6 GHILDIYAL, Y. et al. An imperative role of 6g communication with perspective of industry 4.0: Challenges and research directions. *Sustainable Energy Technologies and Assessments*, v. 56, p. 103047, 2023. ISSN 2213-1388.

7 KAR, U.; SANYAL, D. A critical review of 3gpp standardization of device-to-device communication in cellular networks. *SN Computer Science*, v. 1, 10 2019.

8 JAVAID, M. et al. Significance of sensors for industry 4.0: Roles, capabilities, and applications. *Sensors International*, v. 2, p. 100110, 2021. ISSN 2666-3511. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666351121000310>.

9 INTRODUCTION to Joint Communications and Sensing (JCAS). In: JOINT Communications and Sensing. John Wiley Sons, Ltd, 2022. cap. 1, p. 1–30. ISBN 9781119982944. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119982944.ch1>.

10 ALI, S.; AHMAD, A. Resource allocation, interference management, and mode selection in device-to-device communication: A survey. *Transactions on Emerging Telecommunications Technologies*, v. 28, n. 7, p. e3148, 2017. E3148 ett.3148.

11 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016.

12 POWELL, W. B. Policy function approximations and policy search. In: ____. *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions*. [S.l.]: John Wiley Sons, Ltd, 2022. cap. 12, p. 653–699. ISBN 9781119815068.

13 SEWAK, M. *Deep Reinforcement Learning*. [S.l.]: Springer Singapore, 2019.

14 CARDOSO, G. Pimenta de F.; CARVALHO, P. Henrique Portela de; GONDIM, P. Roberto de L. Deep reinforcement learning for resource allocation of mobile communication systems with device-to-device underlay. *International Journal of Communication Systems*, p. e5476. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.5476>.

15 A hyper-heuristic based framework for dynamic optimization problems. *Applied Soft Computing*, v. 19, p. 236–251, 2014. ISSN 1568-4946.

16 SANCHEZ, M. et al. A systematic review of hyper-heuristics on combinatorial optimization problems. *IEEE Access*, v. 8, p. 128068–128095, 2020.

17 DRAKE, J. H. et al. Recent advances in selection hyper-heuristics. *European Journal of Operational Research*, v. 285, n. 2, p. 405–428, 2020. ISSN 0377-2217.

18 ZHANG, Y. et al. A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. *European Journal of Operational Research*, v. 300, n. 2, p. 418–427, 2022. ISSN 0377-2217.

19 FISHER, H.; THOMPSON, G. L. Probabilistic learning combinations of local job-shop scheduling rules. *Prentice-Hall*, p. 225–251, 1963.

20 BICă, M.; KOIVUNEN, V. Multicarrier radar-communications waveform design for rf convergence and coexistence. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2019. p. 7780–7784.

21 AHMED, A. et al. Ofdm-based joint radar-communication system: Optimal sub-carrier allocation and power distribution by exploiting mutual information. In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. [S.l.: s.n.], 2019. p. 559–563.

22 SHI, C. et al. Joint optimization scheme for subcarrier selection and power allocation in multicarrier dual-function radar-communication system. *IEEE Systems Journal*, v. 15, n. 1, p. 947–958, 2021.

23 WANG, F.; LI, H. Power allocation for coexisting multicarrier radar and communication systems in cluttered environments. *IEEE Transactions on Signal Processing*, v. 69, p. 1603–1613, 2021.

24 WANG, F.; LI, H.; GOVONI, M. A. Power allocation and co-design of multicarrier communication and radar systems for spectral coexistence. *IEEE Transactions on Signal Processing*, v. 67, n. 14, p. 3818–3831, 2019.

25 FANG, X. et al. Radio map-based spectrum sharing for joint communication and sensing networks. In: *2022 IEEE/CIC International Conference on Communications in China (ICCC)*. [S.l.: s.n.], 2022. p. 238–243.

26 SHI, C. et al. Power control scheme for spectral coexisting multistatic radar and massive mimo communication systems under uncertainties: A robust stackelberg game model. *Digital Signal Processing*, v. 94, p. 146–155, 2019. ISSN 1051-2004. Special Issue on Source Localization in Massive MIMO.

27 MOURTZIS, D.; ANGELOPOULOS, J.; PANOPOULOS, N. Smart manufacturing and tactile internet based on 5g in industry 4.0: Challenges, applications and new trends. *Electronics*, v. 10, n. 24, 2021. ISSN 2079-9292.

28 HAN, B. et al. Digital twins for industry 4.0 in the 6g era. *IEEE Open Journal of Vehicular Technology*, v. 4, p. 820–835, 2023.

29 JU, X. et al. Path availability of the brownian motion mobility model for mobile ad hoc networks. In: *2010 International Conference on Internet Technology and Applications*. [S.l.: s.n.], 2010. p. 1–4.

30 HUANG, Y. et al. Coordinated power control for network integrated sensing and communication. *IEEE Transactions on Vehicular Technology*, v. 71, n. 12, p. 13361–13365, 2022.

31 MEI, F. *28 GHz applications, path loss models and coverage for 5G*. Tese (Doutorado) — Scuola di Ingegneria Industriale e dell'Informazione - Politecnico di Milano, 2019.

32 RAPPAPORT, T. S. *Wireless communications - principles and practice.* [S.l.]: Prentice Hall, 1996. 105, 248 p. ISBN 978-0-13-375536-7.

33 SHI, C. et al. Joint optimization scheme for subcarrier selection and power allocation in multicarrier dual-function radar-communication system. *IEEE Systems Journal*, v. 15, n. 1, p. 947–958, 2021.

34 FANG, X. et al. Radio map-based spectrum sharing for joint communication and sensing networks. In: *2022 IEEE/CIC International Conference on Communications in China (ICCC)*. [S.l.: s.n.], 2022. p. 238–243.

35 DELIGIANNIS, A. et al. Game-theoretic power allocation and the nash equilibrium analysis for a multistatic mimo radar network. *IEEE Transactions on Signal Processing*, v. 65, n. 24, p. 6397–6408, 2017.

36 LI, M.; LIU, W.; LEI, J. A review on orthogonal time–frequency space modulation: State-of-art, hotspots and challenges. *Computer Networks*, v. 224, p. 109597, 2023. ISSN 1389-1286.

37 KOCHENDERFER, M. J.; WHEELER, T. A. *Algorithms for Optimization*. [S.l.]: The MIT Press, 2019. ISBN 0262039427.

38 MENG, F. et al. Power allocation in multi-user cellular networks: Deep reinforcement learning approaches. *IEEE Transactions on Wireless Communications*, v. 19, n. 10, p. 6255–6267, 2020.

39 LI, Z.; GUO, C.; XUAN, Y. A multi-agent deep reinforcement learning based spectrum allocation framework for d2d communications. In: *2019 IEEE Global Communications Conference (GLOBECOM)*. [S.l.: s.n.], 2019. p. 1–6.

40 JI, H.; ALFARRAJ, O.; TOLBA, A. Artificial intelligence-empowered edge of vehicles: Architecture, enabling technologies, and applications. *IEEE Access*, v. 8, p. 61020–61034, 2020.

41 WILLIAMS, R. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, v. 8, p. 229–256, 2004.

42 MOUSAVI, S. S.; SCHUKAT, M.; HOWLEY, E. Deep reinforcement learning: an overview. In: SPRINGER. *Proceedings of SAI Intelligent Systems Conference*. [S.l.], 2016. p. 426–440.

43 WENG, L. A (long) peek into reinforcement learning. *lilianweng.github.io*, 2018. Disponível em: <https://lilianweng.github.io/posts/2018-02-19-rl-overview/>. Acesso em: 2 de março de 2024.

44 MNIH, V. et al. Playing atari with deep reinforcement learning. *ArXiv*, abs/1312.5602, 2013. Disponível em: <https://api.semanticscholar.org/CorpusID:15238391>.

45 WENG, L. Policy gradient algorithms. *lilianweng.github.io*, 2018. Disponível em: <https://lilianweng.github.io/posts/2018-04-08-policy-gradient/>. Acesso em: 13 de março de 2024.

46 ZHANG, J. et al. Sample efficient reinforcement learning with REINFORCE. *CoRR*, abs/2010.11364, 2020. Disponível em: <https://arxiv.org/abs/2010.11364>.

47 FREIRE, P. et al. Computational complexity optimization of neural network-based equalizers in digital signal processing: A comprehensive approach. *Journal of Lightwave Technology*, p. 1–25, 2024.

48 TAN, J.; GUAN, W. Resource allocation of fog radio access network based on deep reinforcement learning. *Engineering Reports*, v. 4, n. 5, p. e12497, 2022.

49 SCHULMAN, J. et al. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017. Disponível em: <https://api.semanticscholar.org/CorpusID:28695052>.

50 LILLICRAP, T. P. et al. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015. Disponível em: <https://api.semanticscholar.org/CorpusID:16326763>.

51 FUJIMOTO, S.; HOOF, H. van; MEGER, D. Addressing function approximation error in actor-critic methods. In: *International Conference on Machine Learning*. [s.n.], 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:3544558>.

52 SCABINI, L. F. S.; BRUNO, O. M. Structure and performance of fully connected neural networks: Emerging complex network properties. *ArXiv*, abs/2107.14062, 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:236493493>.

53 GRANDO, R. B. et al. Deterministic and stochastic analysis of deep reinforcement learning for low dimensional sensing-based navigation of mobile robots. In: *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*. [S.l.: s.n.], 2022. p. 193–198.

54 DOKEROGLU, T.; KUCUKYILMAZ, T.; TALBI, E.-G. Hyper-heuristics: A survey and taxonomy. *Computers Industrial Engineering*, v. 187, p. 109815, 2024. ISSN 0360-8352.

55 MACIAS-ESCOBAR, T.; CRUZ-REYES, L.; DORRONSORO, B. A study on the use of hyper-heuristics based on meta-heuristics for dynamic optimization. In: ____. *Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications*. Cham: Springer International Publishing, 2021. p. 295–314. ISBN 978-3-030-68776-2.

56 PILLAY, N.; QU, R. Selection constructive hyper-heuristics. In: ____. *Hyper-Heuristics: Theory and Applications*. Cham: Springer International Publishing, 2018. p. 3–16. ISBN 978-3-319-96514-7.

57 WANG, Z. et al. Dueling network architectures for deep reinforcement learning. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. [S.l.]: JMLR.org, 2016. (ICML'16), p. 1995–2003.

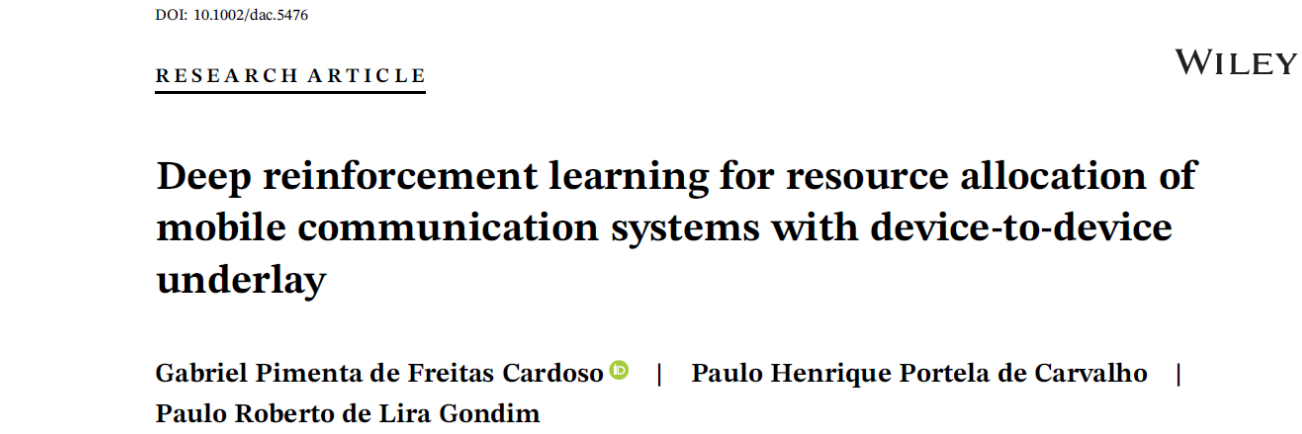
58 HASSELT, H. V.; GUEZ, A.; SILVER, D. Deep reinforcement learning with double q-learning. In: *AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2015.

59 HASSELT, H. Double q-learning. In: LAFFERTY, J. et al. (Ed.). *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2010. v. 23.

APPENDIX

APÊNDICE A - PUBLICAÇÃO REALIZADA

Durante o esforço da pesquisa, foi publicado um artigo no periódico *International Journal of Communication Systems*, cujo DOI é https://doi.org/10.1002/dac.5476. A Figura A.1 mostra o cabeçalho da publicação realizada.


DOI: 10.1002/dac.5476

RESEARCH ARTICLE

WILEY

Deep reinforcement learning for resource allocation of mobile communication systems with device-to-device underlay

Gabriel Pimenta de Freitas Cardoso | **Paulo Henrique Portela de Carvalho** | **Paulo Roberto de Lira Gondim**

Figura A.1 – Cabeçalho do artigo publicado ao longo da pesquisa realizada.