UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Vinicius Rodrigues Oliveira

Aprendizado por reforço na indústria 4.0

Belo Horizonte
2023

Vinicius Rodrigues Oliveira

Aprendizado por reforço na indústria 4.0

Versão Final

Trabalho de Conclusão de Curso

Orientador: Daniel Fernandes Macedo

Belo Horizonte
2023

Resumo

Este trabalho aborda a aplicação do aprendizado por reforço em ambientes industriais no contexto da indústria 4.0. O objetivo principal é investigar o impacto da latência na eficácia e viabilidade do aprendizado por reforço em cenários onde o agente e o ambiente estão em localidades diferentes.

Este trabalho tem como objetivo abordar três questões de pesquisa relacionadas ao tempo de aprendizado, ao impacto de tomar ação após um limite de tempo e à viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro.

O trabalho contribui para o avanço do conhecimento sobre o uso do aprendizado por reforço em cenários reais, nos quais a latência e a distribuição geográfica são fatores relevantes. Os resultados e discussões apresentados podem auxiliar na tomada de decisões na utilização de aprendizado por reforço em uma abordagem descentralizada, na qual o agente de aprendizado por reforço é separado geograficamente do ambiente em que atua.

Palavras-chave: aprendizado por reforço, latência, indústria 4.0.

Abstract

This study addresses the application of reinforcement learning in industrial environments within the context of Industry 4.0. The main objective is to investigate the impact of latency on the effectiveness and feasibility of reinforcement learning in scenarios where the agent and the environment are located in different locations.

This study aims to address three research questions related to learning time, the impact of taking action after a time limit, and the feasibility of applying reinforcement learning in a Brazilian scenario.

The study contributes to the advancement of knowledge regarding the use of reinforcement learning in real-world scenarios, where latency and geographical distribution are relevant factors. The presented results and discussions can assist in decision-making regarding the use of reinforcement learning in a decentralized approach, where the reinforcement learning agent is geographically separated from the environment it operates in.

Keywords: reinforcement learning, latency, industry 4.0.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

A indústria 4.0 tem como objetivo alcançar o mais alto nível de eficiência e produtividade, com o maior nível de automatização [16] e pode ser definida por diferentes mudanças, principalmente impulsionadas pela tecnologia, nos sistemas de manufatura [13]. Mudanças as quais são advento de avanços em áreas como internet das coisas, robótica, automação, computação em nuvem, inteligência artificial (IA).

O aprendizado por reforço pode ser definido como um agente que interage com um ambiente, onde o agente aprende uma política ótima através de tentativa e erro para problemas de tomada de decisão sequencial [14]. O aprendizado por reforço tem revolucionado as aplicações de IA e vem ganhando atenção pela vantagem de possibilitar a tomada de decisão adaptativa [10], além de permitir que problemas sejam resolvidos com eficiência. Aprendizado por reforço é utilizado em diversas áreas, como no processo de otimização em processos industriais [29], em navegação autônoma de veículos aéreos não tripulados [5], controle de comportamento de robôs [30] e muitos outros [11].

No contexto atual de aprendizado por reforço, o agente e o ambiente são modelados em uma arquitetura de controle local, ou seja, se encontram na mesma localidade. Tal arquitetura impacta em algumas limitações, como exemplo, o custo computacional para se ter um agente eficaz pode ser muito alto. Em [19] é discutido o custo computacional para se treinar um famoso algoritmo denominado Rainbow. Os autores relatam que mesmo utilizando hardware especializado (NVIDIA Tesla P100 GPU) o algoritmo levou cinco dias para ser totalmente treinado.

É financeiramente inviável equipar todos os dispositivos de uma indústria 4.0 com hardware especializado para a aplicação de aprendizado por reforço. Diante disso, este trabalho propõe uma abordagem descentralizada, na qual o agente de aprendizado por reforço é separado geograficamente do ambiente em que atua, permitindo assim avaliar o impacto da rede nessa arquitetura. O objetivo deste trabalho é investigar e responder às seguintes questões de pesquisa:

RQ1: Qual é o efeito da latência no tempo de aprendizado por reforço? Qual é o tempo necessário para treinar o algoritmo quando nenhuma ação é executada, apenas aguardando a chegada do pacote?

RQ2: Qual é o impacto de tomar uma ação após definir um limite de

tempo para a chegada do pacote? É mais eficaz tomar uma ação aleatória ou repetir a última ação após atingir o limite de tempo?

RQ3: Qual é a viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro onde o agente e o ambiente estão em localidades diferentes?

Este trabalho está organizado da seguinte forma: aspectos contextuais sobre instruia 4.0, aprendizado por reforço e métricas de redes são apresentados na Seção 2. Trabalhos relacionados são discutidos na Seção 3. Detalhes sobre a modelagem do ambiente de teste na Seção 4. Resultados e discussões na Seção 5. Finalmente, as conclusões e direções para trabalhos futuros são apresentadas na Seção 6.

Capítulo 2

Contextualização

2.1 Indústria 4.0

O termo indústria 4.0 foi mencionado a primeira vez em 2011, na Alemanha, durante uma das maiores feiras do mundo (Feira Hannover), a feira é dedicada ao tema de desenvolvimento industrial [28]. A indústria 4.0 é fruto de um projeto entre o governo da Alemanha com universidades e empresas privadas. Foi um projeto com objetivo de aumentar a produtividade e eficiência da indústria local [8].

A indústria 4.0 também pode ser chamada de Quarta Revolução Industrial. As palavras mecanização, eletrificação e informatização resumem as outras três revoluções, respectivamente. Sendo assim, a Quarta Revolução Industrial é definida pela palavra digitalização[7]. Ela é caracterizada pela introdução das seguintes tecnologias no contexto industrial: Internet da Coisas (IoT), sistemas ciber-físicos (CPS), *big data*, análise de dados, computação em nuvem, inteligência artificial, robótica, sensores inteligentes, impressoras 3D[22].

A digitalização promovida pela indústria 4.0 visa a auto-otimização, onde os dispositivos sejam capazes de aprender com a experiência e se comunicarem entre si para tomar por contra própria as melhores decisões [20].

Posto isto, algoritmos de aprendizado por reforço propõe justamente que dispositivos aprendam a tomar a melhor decisão com base nas experiências passadas. São vastos os exemplos da aplicação de aprendizado por reforço na indústria 4.0.

Em [26] é proposta a utilização de aprendizado por reforço para o controle de temperatura no processo de fabricação de plástico reforçado com fibra de carbono. Haja vista que a modificação das variáveis que afetam a transferência de calor durante o processamento do compósito é uma forma de otimizar o processo de fabricação.

Um exemplo adicional da aplicação do aprendizado por reforço ocorre na indústria de energia elétrica, que enfrenta o desafio de atender à demanda de energia de maneira eficiente, garantindo redes confiáveis e custos reduzidos. Os autores em [15] propõem um esquema de resposta à demanda baseado no aprendizado por reforço profundo de

múltiplos agentes para o gerenciamento energético de sistemas de manufatura discretos.

É incontestável que a indústria 4.0 apresenta perspectivas promissoras para incrementar a produtividade, competitividade e eficiência. Entretanto, sua plena implementação depara-se com desafios de natureza diversa, entre os quais se destacam a capacitação da força de trabalho, a integração de dispositivos heterogêneos e, sobretudo, a necessidade de investimentos substanciais em infraestrutura tecnológica.

Em conformidade com o exposto, é válido ressaltar que pesquisas e estudos, como o realizado pela Confederação Nacional da Indústria (CNI), evidenciam que o custo elevado de implantação figura como a principal barreira interna para a adoção de tecnologias digitais na indústria, sendo mencionado por 66% das empresas participantes do referido estudo [1]. Nesse sentido, percebe-se que o desafio preponderante no contexto da indústria 4.0 no Brasil reside na viabilidade de soluções com custos reduzidos, capazes de tornar a adoção dessas tecnologias mais acessível e atrativa para as organizações. A busca por alternativas que conciliem eficácia e economia torna-se, assim, um elemento essencial para impulsionar a inserção plena da indústria 4.0 no cenário industrial brasileiro.

2.2 Aprendizado por Reforço

Aprendizado por reforço, também conhecido como Reinforcement Learning (RL) é uma das áreas de aprendizagem de máquina. Aprendizado por reforço diz respeito ao paradigma de aprendizado em que um agente aprende um determinado comportamento por meio de interações de tentativa e erro com um ambiente dinâmico [10]. Diferente dos aprendizados supervisionado e semi-supervionado, no aprendizado por reforço, o agente não possui acesso a um conjunto de dados rotulados para treinamento, em vez disso, o agente aprende a tomar decisões com base nas recompensas ou penalidades que ele recebe do ambiente em respostas às suas ações [25].

No cenário de Aprendizado por Reforço, o agente está conectado ao ambiente através de estado e ação, conforme mostra a Figura 2.1. Em cada interação,t, o agente recebe do ambiente uma indicação do estado atual, em seguida o agente escolhe uma ação, A, para gerar um saída. A ação gerada pelo agente é repassada ao ambiente que tem seu estado alterado e o valor dessa transição é repassado para o agente por meio de uma recompensa, R. O comportamento do agente deve ser escolher açãoes que tendam aumentar a soma das recompensas ao longo do tempo [10].

Em outras palavras, o agente interage com o ambiente de maneira sequencial, onde ele toma uma ação em um determinado estado e, em seguida, recebe feedback do ambiente na forma de uma recompensa, que representa o quão boa ou ruim foi a ação tomada. O

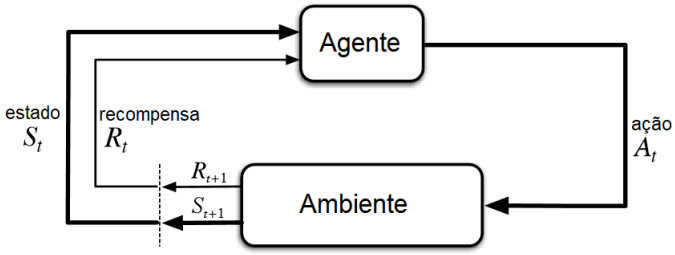

Figura 2.1: Diagrama do aprendizado por reforço

objetivo do agente é aprender uma política, ou seja, uma estratégia que mapeia estados para ações, de modo a maximizar a recompensa acumulada ao longo do tempo.

Um dos grandes desafios do Aprendizado por Reforço é o equilibrio entre *exploration* e *exploitation*, utilizadas neste trabalho em inglês, devido a tradução para o português de ambas as palavras serem exploração. A *exploration* refere-se à necessidade do agente de explorar o ambiente a fim de descobrir quais ações resultam em maiores recompensas. Por outro lado, a *exploitation* envolve a escolha das ações que foram previamente consideradas eficazes na obtenção de boas recompensas. O dilema entre a *exploration-exploitation* tem sido extensivamente estudado por matemáticos ao longo de várias décadas, mas ainda não foi completamente resolvido [25].

Aprendizado por Reforço conta com diversos algoritmos que foram propostos ao longo dos anos. Por exemplo, Q-Learning, Deep Q-Networks (DQN) e Proximal Policy Optimization (PPO).

Q-Learning é um algoritmo amplamente utilizado e ele se baseia em uma função de valor de ação, conhecida como *Q-function*, que estima o valor esperado de uma ação em um determinado estado. O Q-Learning atualiza iterativamente a *Q-function* com base nas observações do ambiente, permitindo ao agente ajustar sua política de ação de acordo com as recompensas recebidas [27]. O Deep Q-Networks é uma evolução do Q-Learning, ele utiliza uma arquitetura de rede neural para estimar a *Q-function* [18].

O Aprendizado por Reforço tem sido amplamente aplicado em várias áreas, abrangendo desde a otimização de processos industriais [29], até a navegação autônoma de veículos aéreos não tripulados [5], controle de comportamento de robôs [30], e uma ampla gama de outros domínios [11]. Essa abordagem permite que agentes aprendam a tomar decisões em ambientes complexos por meio de interações com o ambiente e a obtenção de recompensas.

A aplicação de aprendizado por reforço na indústria 4.0 é mais relevante em comparação com outras abordagens de aprendizado devido à necessidade de alcançar um alto nível de otimização nas ferramentas inteligentes utilizadas nesse contexto, o qual não pode ser alcançado por meio de intervenções humanas. Portanto, o aprendizado por reforço atende à demanda por soluções de autoaprendizado exigida nesse ambiente [11].

O Aprendizado por Reforço tem demonstrado ser uma técnica poderosa para enfrentar problemas desafiadores em diversas aplicações, contribuindo para avanços significativos em diferentes campos de estudo.

2.3 Qualidade de Serviço (QoS)

A qualidade de serviço (QoS), no contexto de redes de computadores, diz respeito à capacidade de uma rede ou sistema fornecer um nível adequado e confiável de desempenho, atendendo aos requisitos específicos de desempenho em termos de características como largura de banda, latência, jitter, perda de pacotes e disponibilidade para diferentes tipos de aplicações.

A latência é uma métrica de extrema importância em redes de computadores, especialmente em cenários que envolvem monitoramento e análise de dados em tempo real, bem como computação em nuvem. Ela se refere ao atraso ou tempo de espera desde o momento em que um pacote de dados é enviado de sua origem até o momento em que é recebido em seu destino [24].

Outra métrica crucial para aplicações que dependem de execução em tempo real é o jitter. O jitter refere-se à variação no atraso de entrega dos pacotes de dados. Enquanto a latência representa o atraso médio entre a transmissão e a recepção dos pacotes, o jitter quantifica a inconsistência ou a flutuação nesses atrasos.

A qualidade de serviço (QoS) desempenha um papel fundamental no funcionamento eficiente e confiável das redes de computadores. Métricas como latência e jitter desempenham um papel crítico em aplicações que dependem de execução em tempo real.

Capítulo 3

Trabalhos relacionados

Bernardo et al.(2022) investigam os aspectos de como o desempenho de um modelo de aprendizado por reforço pré-treinado em condições de redes ideais e com variações da rede pode ser impactado [3].

Marques et al.(2018) analisam os impactos das tecnologias software-defined networking em sistemas de computação em nuvem de aplicações que rodam em tempo real com baixa latência e requisitam alta largura de banda [17].

Já em [9], Jiang et al. (2020) propõem uma estrutura assistida por IA para redes sem fio para a otimização da latência de informações, considerando operação de multiagentes do ponto de vista de aprendizado por reforço. Foram estabelecidas métricas, como exemplo de confiabilidade de transmissão de pacotes, a partir de análises em nuvem dos desafios de otimização de latência.

No trabalho aprensentado em [24], Silva et al. (2018) realizão um experimento para aferir a latência simulando uma arquitetura de aplicações para a indústria 4.0. É medido valores de latência em um cenário intercontinental, entre servidores do Brasil e da Itália, para dois protocolos de TIC (Tecnologia da Informação e Comunicação), quais sejam: MQTT (Message Queuing Telemetry Transport) e WebSocket.

Em relação ao aprendizado por reforço de múltiplos agentes (MARL), abordado em [23], Roig et al. (2020) destacam os desafios de comunicação em um canal ruidoso, onde os agentes trabalham em conjunto para alcançar um objetivo comum. Eles propõem o uso de redes neurais profundas, utilizando o algoritmo Deep Q-Learning (DQN), para treinar os agentes a agirem e se comunicarem em canais com presença de erros.

Em relação aos trabalhos existentes, eles estão concentrado na definição de métricas de rede para garantir a operação eficiente das aplicações, bem como na otimização dessas métricas e no estudo do desempenho de modelos de aprendizado por reforço pré-treinados. No entanto, neste trabalho, a proposta é direcionada para compreender especificamente o impacto da latência na eficácia e na viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro.

Capítulo 4

Metodologia

Esta seção apresenta os componentes de software essenciais para os testes. Em seguida, é descrito o simulador de rede utilizado para avaliar o tempo de aprendizado do algoritmo de aprendizado por reforço. Por fim, é discutido os procedimentos aplicados para responder às perguntas de pesquisa.

4.1 Componentes de Software

Para emular o ambiente da indústria 4.0 foi utilizado a biblioteca OpenAI Gym[1]. Gym é uma biblioteca de código aberto em Python para treinar, testar e avaliar algoritmos de aprendizado por reforço. A biblioteca fornece uma API padrão para comunicação entre os algoritmos de aprendizado e os ambientes, juntamente com um conjunto de ambientes pré-definidos.

A escolha da biblioteca OpenAI Gym neste trabalho para *benchmarks* dos algoritmos de aprendizado por reforço se deve à sua ampla utilização na comunidade científica e à sua capacidade de combinar os melhores elementos de coleções de *benchmarks* anteriores em um pacote de software que oferece máxima conveniência e acessibilidade [4].

O ambiente utilizado neste trabalho é o Lunar Lander[2], uma simulação que representa um cenário em que um veículo espacial, denominado Lunar Lander, precisa realizar um pouso suave em uma superfície lunar, ilutrado na Figura 4.1. O espaço de ação desse ambiente é composto por quatro ações discretas, cada uma correspondendo a uma ação específica disponível para o agente: (0) não fazer nada, (1) acionar os motores principais para a esquerda, (2) acionar o motor principal e (3) acionar os motores principais para a direita. Considera-se que o ambiente tenha sido resolvido quando a média dos valores de recompensa ao longo de um episódio atinge a marca de 200.

Stable Baselines3[3] é uma biblioteca que oferece uma implementação confiável de

[1] https://github.com/openai/gym
[2] https://gymnasium.farama.org/environments/box2d/lunar_lander/
[3] https://stable-baselines3.readthedocs.io/en/master/

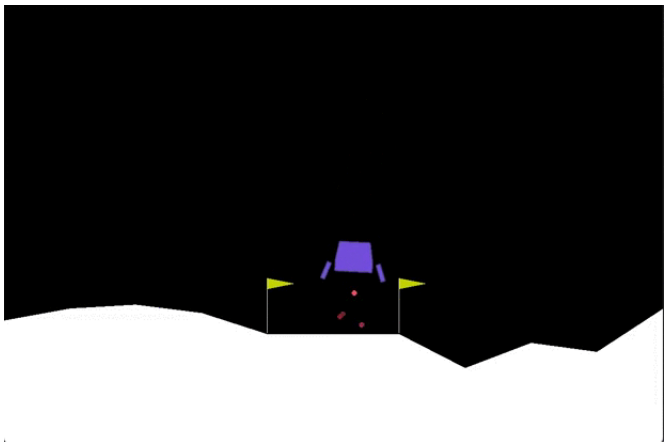

Figura 4.1: Lunar Lander

uma variedade de algoritmos de aprendizado por reforço em PyTorch. A biblioteca assume um papel central como a representação do agente neste trabalho, onde realiza as tarefas de aprendizado e tomada de decisões.

Neste trabalho, foi utilizado o algoritmo Deep Q-Networks (DQN), que emprega uma arquitetura de rede neural para estimar a *Q-function* [18]. A *Q-function* é uma medida que atribui um valor de utilidade para cada ação possível em um determinado estado. O DQN utiliza técnicas de otimização para treinar a rede neural de forma a maximizar os valores de Q para ações que levam a recompensas maiores.

Os hiperparâmetros utilizados no treinamento do algoritmo foram obtidos a partir de referências anteriores [21]. Essas referências fornecem configurações de hiperparâmetros que foram amplamente testadas e mostraram bom desempenho em problemas de aprendizado por reforço. A seleção desses hiperparâmetros visa garantir resultados comparáveis e confiáveis neste estudo.

4.2 Simulador de rede

O simulador de rede, foi baseado no que é apresentado no trabalho [3], onde é composto por duas máquinas virtuais (VMs), cliente e servidor, onde uma contém o agente e a outro o ambiente.

É utilizado a ferramenta network emulator (NetEm)[4] do Linux que fornece funcionalidades para emular rede para testar propriedades de redes do mundo real para alterar os parâmetros de rede entre as máquinas. A Figura 4.2 demonstra o simulador de rede.

[4] https://www.linux.org/docs/man8/tc-netem.html

Figura 4.2: Simulador de Rede. Fonte:autor

4.3 Procedimentos

Para investigar o impacto da latência no tempo de aprendizado por reforço, utilizou-se o simulador de rede descrito na seção anterior. De acordo o trabalho em [3], a latência foi variada no intervalo de 0ms a 50ms, enquanto em [2] é definido um valor de latência de 100ms como limite de sobrevivência para processos autônomos. Portanto, optou-se por gerar valores de latência entre 0ms e 100ms. Além disso, fixou-se o valor do Jitter em 20ms.

Neste trabalho, foram avaliadas apenas as métricas de latência e jitter, uma vez que são consideradas as mais relevantes em termos QoS para serviços em tempo real [12]. No entanto, em trabalhos futuros, é considerado avaliar outras métricas que também podem ter um impacto significativo, tais como corrupção de pacotes, perda de pacotes e reordenamento de pacotes.

Ainda no escopo do impacto da latência no tempo de aprendizado, foram realizadas modificações na biblioteca Stable Baselines3 para possibilitar a troca de informações por meio de uma rede. Com o objetivo de habilitar essa funcionalidade, foi implementada uma função específica dentro da biblioteca, permitindo o envio e recebimento de informações através de sockets. Essa adição tornou possível a comunicação entre o agente de aprendizado por reforço e o ambiente de simulação, estabelecendo um canal de troca de dados para a interação entre ambos.

No que diz respeito ao impacto de tomar uma ação após estabelecer um limite de tempo para a chegada de pacotes, optamos por realizar o treinamento em ambiente local a fim de reduzir o tempo necessário para o treinamento. Para simular o atraso, foram gerados dados aleatórios seguindo uma distribuição normal, em que a média representa a latência e o desvio padrão representa o jitter. O valor limite foi estabelecido com base em referências anteriores [2], que definem 60ms como o limite máximo aceitável de latência para aplicações de processos autônomos. Essa definição proporcionou um critério objetivo para avaliar o desempenho do sistema diante da restrição de tempo estabelecida.

A simulação dos atrasos é realizada no contexto da biblioteca Stable Baselines3.

Com o intuito de habilitar essa funcionalidade, foi desenvolvida uma função específica dentro da biblioteca, a qual verifica o valor do atraso para determinar se o agente deve seguir o fluxo normal do processo ou tomar uma ação aleatória, ou ainda repetir a última ação tomada.

A fim de avaliar a viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro no qual o agente e o ambiente estão localizados em diferentes localidades geográficas, realizou-se uma análise empírica, levando em consideração os resultados obtidos anteriormente. Essa análise foi enriquecida com informações relevantes sobre as métricas de rede específicas do Brasil, conforme apresentado na Tabela 4.1, que retrata as métricas de rede no país disponibilizadas por [6].

Estado	Latência	Jitter
Centro Oeste	31.7ms	1.24ms
Nordeste	46.2ms	1.27ms
Norte	61.6ms	1.00ms
Sudeste	15.1ms	1.10ms
Sul	23.2ms	1.12ms

Tabela 4.1: Métricas de rede por região do Brasil

Capítulo 5

Resultados e Discussões

Esta seção apresenta os resultados e discussões das três questões de pesquisa. A RQ1 investiga o efeito da latência no tempo de aprendizado por reforço. A RQ2 explora o impacto de tomar uma ação após estabelecer um limite de tempo para a chegada do pacote. Por fim, a RQ3 aborda a viabilidade da aplicação do aprendizado por reforço em uma arquitetura descentralizada no cenário brasileiro.

RQ1: Qual é o efeito da latência no tempo de aprendizado por reforço?

O objetivo dessa pergunta foi investigar o impacto da latência no desempenho do algoritmo quando nenhuma ação é executada, ou seja, quando o agente aguarda a chegada do pacote sem realizar qualquer ação, e também determinar o tempo necessário para treinar o agente.

Os resultados obtidos, conforme ilustrado na Figura 5.1, demonstram que, mesmo com valores altos de delay, o algoritmo converge para o valor ideal de recompensa, que é de 200. Portanto, é possível concluir que, ao considerar apenas a latência e desconsiderar outros fatores associados, esta não impacta significativamente no desempenho do agente. Tal resultado era esperado, uma vez que não há influência além do atraso na comunicação entre o agente e o ambiente.

Por outro lado, a Tabela 5.1 apresenta os valores de tempo de treinamento do agente, onde é evidente que o tempo aumenta consideravelmente de acordo com os atrasos gerados.

Latência	Tempo
0ms	00h15min
25ms	02h37min
50ms	04h55min
100ms	08h54min

Tabela 5.1: Tempo de treinamento

Esses resultados evidenciam claramente que a latência exerce um impacto significativo no tempo de treinamento do agente. Quanto maior a latência, maior é o tempo necessário para que o agente seja completamente treinado. Isso sugere que a latência

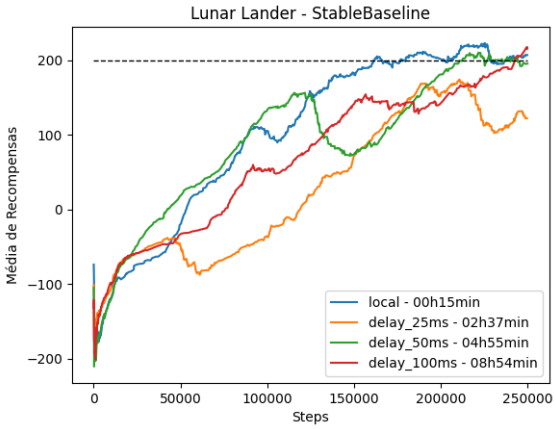

Figura 5.1: Impacto da Latência no tempo de treinamento. Fonte:autor

pode ser um fator crítico a ser considerado em projetos de arquitetura descentralizada com algoritmos de aprendizado por reforço.

Essas descobertas têm implicações importantes para a implementação de aprendizado por reforço que dependem de interações rápidas e eficientes entre o agente e o ambiente. Considerar e mitigar os efeitos da latência pode ser crucial para garantir um desempenho adequado e eficaz desses sistemas em contextos reais.

RQ2: Qual é o impacto de tomar uma ação após definir um limite de tempo para a chegada do pacote?

É investigado aqui o impacto de tomar uma ação, seja ela aleatória ou repetir a última ação, após estabelecer um limite de tempo para a chegada do pacote, sendo esse limite definido como 60ms. A análise foi realizada considerando diferentes valores de latência: 50ms, 60ms e 70ms.

Nos casos em que a latência foi de 50ms e 60ms, os resultados revelaram que tomar a última ação após o limite de tempo estabelecido teve um bom desempenho, conforme desmonstra as Figuras 5.2 e 5.3. Os valores de recompensa obtidos foram próximos do valor ideal de convergência, que é de 200. Isso indica que o agente conseguiu realizar ações adequadas e maximizar sua recompensa mesmo diante da latência presente no ambiente. Por outro lado, quando o agente tomou uma ação aleatória após o limite de tempo, o desempenho foi significativamente inferior, com valores de recompensa distanciados do ideal.

Em contrapartida, quando a latência aumentou para 70ms, o algoritmo apresentou um desempenho ruim, independentemente de tomar a última ação ou uma ação aleatória após o limite de tempo, conforme a Figura 5.4. Nesse caso, os valores de recompensa foram substancialmente baixos, indicando uma dificuldade do agente em realizar ações efetivas e alcançar resultados satisfatórios.

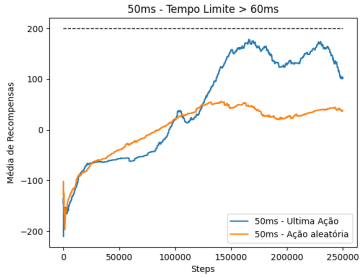

Figura 5.2: Tempo 50 ms. Fonte:autor

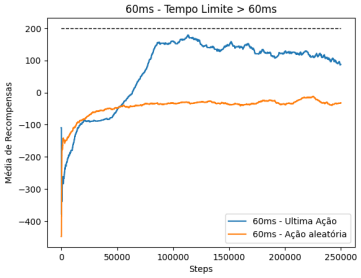

Figura 5.3: Tempo 60 ms. Fonte:autor

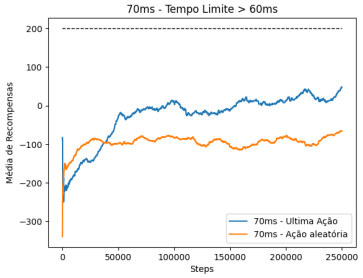

Figura 5.4: Tempo 70 ms. Fonte:autor

Essas conclusões destacam a importância de uma cuidadosa consideração do tempo limite para a tomada de ação em cenários com latência, juntamente com o desenvolvimento de estratégias e algoritmos capazes de lidar de forma eficiente com atrasos na comunicação. Fica claro que é mais viável empregar estratégias de tomar ação após um tempo limite em valores menores ou próximos do limite estabelecido. Essas considerações ressaltam a necessidade de um planejamento adequado e de abordagens adaptativas para garantir o desempenho desejado em ambientes com atrasos significativos de comunicação.

RQ3: Qual é a viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro onde o agente e o ambiente estão em localidades diferentes?

Os resultados obtidos na análise empírica, com base nos resultados anteriores, sobre a viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro, onde o agente e o ambiente estão localizados em regiões distintas do país, são sumarizados na Tabela 5.2. Esses resultados levam em consideração as métricas de rede nas cinco regiões do Brasil.

Observa-se que, no Centro-Oeste, um algoritmo de aprendizado por reforço levaria entre 2 horas e meia a 4 horas para ser totalmente treinado. No Nordeste, esse tempo seria de aproximadamente 5 horas. Na região Norte, o tempo estimado varia de 5 a 8 horas. Já nas regiões Sul e Sudeste, o tempo de treinamento seria inferior a 2 horas e meia.

É importante ressaltar que, em todos os casos, é necessário avaliar se o tempo de espera é ideal para a aplicação em questão. No entanto, com base nos resultados, pode-se inferir que, em geral, é mais vantajoso não estabelecer um tempo limite para a tomada de ação, a menos que seja uma situação crítica que exija uma resposta imediata.

Com base nessas constatações, pode-se concluir que existe viabilidade na aplicação de uma abordagem descentralizada de aprendizado por reforço no cenário brasileiro, considerando as particularidades de cada região. Essa abordagem permite adaptar o treina-

Estado	Latência	Jitter	Tempo de Treinamento
Centro Oeste	31.7ms	1.24ms	entre 2h30 e 4h
Nordeste	46.2ms	1.27ms	aproximadamente 5h
Norte	61.6ms	1.00ms	entre 5h e 8h
Sudeste	15.1ms	1.10ms	menos que 2h30
Sul	23.2ms	1.12ms	menos que 2h30

Tabela 5.2: Análise empírica de um abordagem descentralizada de aprendizagem por reforço no cenário brasileiro

mento e a tomada de decisão de acordo com as métricas de rede específicas de cada região, garantindo um desempenho adequado do algoritmo em diferentes contextos geográficos.

Capítulo 6

Conclusão

Neste trabalho, foram investigadas três questões de pesquisa relacionadas à viabilidade da aplicação do aprendizado por reforço em um abordagem descentralizada de aprendizado por reforço, no qual o agente e o ambiente estão em localidades diferentes. As questões de pesquisa abordaram o efeito da latência no tempo de aprendizado por reforço, o impacto de tomar uma ação após estabelecer um limite de tempo para a chegada do pacote e a viabilidade geral da abordagem descentralizada em diferentes regiões do Brasil.

Na primeira questão de pesquisa, foi constatado que a latência não exerce um impacto significativo no desempenho do algoritmo de aprendizado por reforço em termos de convergência para o valor ideal de recompensa. Mesmo com valores altos de delay, o algoritmo foi capaz de atingir o objetivo desejado. No entanto, a análise revelou que a latência afeta significativamente o tempo de treinamento do agente. Quanto maior a latência, maior é o tempo necessário para o agente ser completamente treinado. Esses resultados ressaltam a importância de considerar e mitigar os efeitos da latência ao projetar algoritmos de aprendizado por reforço em ambientes com restrições de tempo real.

Na segunda questão de pesquisa os resultados evidenciaram que é mais viável empregar estratégias de tomar ação após um tempo limite em valores menores ou próximos do tempo limite estabelecido. Além disso, constatou-se que repetir a última ação após o limite de tempo teve um desempenho superior em comparação com a realização de uma ação aleatória

Por fim, na terceira questão de pesquisa, os resultados demonstraram que o tempo de treinamento varia de acordo com as métricas de rede específicas de cada região. No Centro-Oeste, Nordeste e Norte, o tempo de treinamento é relativamente maior, enquanto nas regiões Sul e Sudeste, o tempo é menor. Apesar das diferenças regionais, os resultados sugerem que uma abordagem descentralizada de aprendizado por reforço é viável no cenário brasileiro, permitindo a adaptação do treinamento e da tomada de decisão de acordo com as particularidades de cada região.

Essas descobertas têm implicações importantes para a implementação de aprendizado por reforço em cenários reais, nos quais a latência e a distribuição geográfica são fatores relevantes. Considerar a latência, estabelecer limites de tempo adequados e desenvolver estratégias adaptativas são aspectos cruciais para garantir um desempenho eficaz

desses sistemas. Além disso, a abordagem descentralizada de aprendizado por reforço demonstra ser promissora para lidar com as particularidades de cada região, tornando-se uma alternativa viável para aplicação em cenários geograficamente distribuídos.

Como perspectivas para trabalhos futuros, propõe-se utilizar as métricas de rede como parâmetro para o treinamento do agente. Isso envolveria considerar não apenas a latência e o jitter, mas também outras métricas relevantes para o desempenho da rede. Além disso, seria interessante expandir o escopo do estudo para incluir ambientes mais complexos, que possuam um conjunto maior de ações.

Referências Bibliográficas

[1] Sondagem especial: Indústria 4.0: novo desafio para a indústria brasileira. Acessado: 28 de Junho de 2023.

[2] 5G Americas. 5g communications for automation in vertical domains. November 2018.

[3] Guilherme Bernardo, Gilson Miranda Jr., and Daniel Macedo. Analysis of network performance over deep reinforcement learning control loops for industry 4.0. In *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 1–14, Porto Alegre, RS, Brasil, 2022. SBC.

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[5] J. Wang C. Wang, J. Wang and X. Zhang. Deep-reinforcement-learning-based autonomous uav navigation with sparse rewards. *IEEE Internet of Things Journal*, 7(07):6180–6190, July 2020. doi: 10.1109/JIOT.2020.2973193.

[6] Ceptro.br. Covid-19 impactos na qualidade da internet no brasil. janeiro 2021. Disponível em: https://www.ceptro.br/assets/publicacoes/pdf/2021.01.05-relatorio-covid.pdf.

[7] Gustavo Dalmarco, Filipa R. Ramalho, Ana C. Barros, and Antonio L. Soares. Providing industry 4.0 technologies: The case of a production technology cluster. *The Journal of High Technology Management Research*, 30(2):100355, 2019.

[8] Alejandro Germán Frank, Lucas Santos Dalenogare, and Néstor Fabián Ayala. Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*, 210:15–26, 2019.

[9] Zhiyuan Jiang, Siyu Fu, Sheng Zhou, Zhisheng Niu, Shunqing Zhang, and Shugong Xu. Ai-assisted low information latency wireless networking. *IEEE Wireless Communications*, 27(1):108–115, 2020.

[10] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *J. Artif. Int. Res.*, 4(1):237–285, may 1996.

[11] Tamás Kegyes, Zoltán Süle, and János Abonyi. The Applicability of Reinforcement Learning Methods in the Development of Industry 4.0 Applications. *Complexity*, 2021:1–31, November 2021.

[12] Rafael Kunst, Leandro Avila, Alécio Binotto, Edison Pignaton, Sergio Bampi, and Juergen Rochol. Improving devices communication in industry 4.0 wireless networks. *Engineering Applications of Artificial Intelligence*, 83:1–12, 2019.

[13] Kemper H-G Feld T Hoffmann M Lasi H, Fettke P. Industrie 4.0. wirtschaftsinformatik. 2014. doi: 10.1007/ s11576-014-0424-4.

[14] Yuxi. Li. Deep reinforcement learning: An overview. 2017. doi: urlhttps://doi.org/10.48550/arXiv.1701.07274.

[15] Renzhi Lu, Yi-Chang Li, Yuting Li, Junhui Jiang, and Yuemin Ding. Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management. *Applied Energy*, 276:115473, 2020.

[16] Y. Lu. Industry 4.0: a survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6:1–10, 2017.

[17] Paulo Marques, Alexandre P. do Carmo, Valerio Frascolla, Carlos Silva, Emanuel D. R. Sena, Raphael Braga, João Pinheiro, Carlos A. Astudillo, Tiago P. C. de Andrade, Eduardo S. Gama, Luiz F. Bittencourt, Leandro A. Villas, Edmundo R. M. Madeira, Nelson L. S. da Fonseca, Cristiano Both, Gabriel Lando, Matias Schimuneck, Juliano Wickboldt, Ana P. V. Trevisan, Rafael de Jesus Martins, Raquel F. Vassallo, Felippe M. de Queiroz, Rodolfo Picoreti, Roberta L. Gomes, Cristina K. Dominicini, Víctor García, Rafael S. Guimarães, Rodolfo Villaca, Magnos Martinello, Moises R.N. Ribeiro, Daniel F. Macedo, Vinicius F. Silva, Julio C. T. Guimarães, Carlos Colman-Meixner, Reza Nejabati, Dimitra Simeonidou, Yi Zhang, Frank Slyne, Pedro Alvarez, Diarmuid Collins, Marco Ruffini, Luiz A. DaSilva, and Johann M. Marquez-Barja. Optical and wireless network convergence in 5g systems – an experimental approach. In *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–5, 2018.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[19] Johan S Obando-Ceron and Pablo Samuel Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2021.

[20] Fabian Quint, Katharina Sebastian, and Dominic Gorecky. A mixed-reality learning environment. *Procedia Computer Science*, 75:43–48, 2015. 2015 International Conference Virtual and Augmented Reality in Education.

[21] Antonin Raffin. Rl baselines3 zoo. https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

[22] Felipe Bastos dos. REIS. Indústria 4.0 em manufaturas no brasil: análise dos benefícios e barreiras de adoç. *ão Paulo : Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, 2021. Dissertação de Mestrado em Administração*, 2018. doi: urlhttps://doi.org/10.11606/D.12.2021.tde-05012022-204134.

[23] Joan S. Pujol Roig and Deniz Gündüz. Remote reinforcement learning over a noisy channel. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020.

[24] Diego R. C. Silva, Guilherme M. B. Oliveira, Ivanovitch Silva, Paolo Ferrari, and Emiliano Sisinni. Latency evaluation for mqtt and websocket protocols: an industry 4.0 perspective. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 01233–01238, 2018.

[25] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[26] Martin Szarski and Sunita Chauhan. Composite temperature profile and tooling optimization via deep reinforcement learning. *Composites Part A: Applied Science and Manufacturing*, 142:106235, 2021.

[27] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8, 1992.

[28] Li Da Xu, Eric L. Xu, and Ling Li. Industry 4.0: state of the art and future trends. *International Journal of Production Research*, 56(8):2941–2962, 2018.

[29] T. Chai-J. Li Y. Jiang, J. Fan and F. L. Lewis. Data-driven flotation industrial process operational optimal control based on reinforcement learning. *IEEE Transactions on Industrial Informatics*, 14(05):1974–1989, May 2018. doi: 10.1109/TII.2017.2761852.

[30] S. Pan-Y. Li Y. Liu, W. Zhang and Y. Chen. Analyzing the robotic behavior in a smart city with deep enforcement and imitation learning using iort. *Computer Communications*, 150:346–356, 2020.