

Projeto de Pesquisa

Iniciação Científica — Trabalho de Conclusão de Curso

DeepColoring - Colorindo grafos com Deep Reinforcement Learning e Graph Neural Networks

Estudante: Carlos Alberto Marques Rabelo
Bacharelado em Ciência da Computação

Orientadora: Nina S. T. Hirata
Departamento de Ciência da Computação

Instituto de Matemática e Estatística
Universidade de São Paulo

Resumo

O Problema da Coloração de Grafos (GCP, do inglês *Graph Coloring Problem*) consiste na tarefa de colorir os vértices de um grafo com a menor quantidade possível de cores e de modo que vértices adjacentes recebam cores distintas. Dado que se trata de um problema \mathcal{NP} -difícil e, dessa forma, não existir um algoritmo que o resolva em tempo polinomial para todas as instâncias, diversas abordagens são adotadas para, ao menos, soluções aproximadas serem obtidas. A aplicação bem-sucedida de métodos baseados em redes neurais em diferentes domínios tem motivado sua adoção também em problemas de Otimização Combinatória e Teoria dos Grafos, como é o caso do GCP. Nesse contexto, o Aprendizado por Reforço, que também se beneficia da adoção de redes neurais em seus métodos, destaca-se como uma abordagem promissora para o GCP, principalmente por não exigir um conjunto de dados previamente rotulado para o treinamento. Paralelamente, trabalhos na literatura vem demonstrando que existem arquiteturas específicas de Redes Neurais para Grafos (GNNs, do inglês *Graph Neural Networks*) adequadas para a resolução de instâncias do GCP. Este projeto visa investigar a integração das melhores práticas oriundas dessas duas abordagens, com o objetivo de aprimorar um algoritmo baseado em redes neurais voltado à resolução de instâncias do GCP, buscando soluções mais eficientes e generalizáveis.

São Paulo, 21 de abril de 2025

1 Introdução

O Problema da Coloração de Grafos (GCP, do inglês *Graph Coloring Problem*) é um problema clássico das áreas de Teoria dos Grafos e Otimização Combinatória. Trata-se da tarefa de colorir os vértices de um grafo, de modo que vértices adjacentes recebam cores distintas. Deseja-se utilizar o menor número de cores possível para essa tarefa, e, dado um grafo G , tal número é denominado *número cromático de G* (denotado por $\chi(G)$). De maneira mais formal, sendo $G = (V, E)$, em que V é o conjunto dos vértices de G e E é o conjunto das arestas de G , sendo k um inteiro, uma k -coloração de G é uma partição $\{V_1, V_2, \dots, V_k\}$ de V , em que V_i , o conjunto de vértices de G que possuem a cor i , é um conjunto estável, ou seja, os vértices de V_i não são adjacentes. Um grafo é dito *k -colorível* se possui uma k -coloração, e, dessa forma, $\chi(G)$ é o menor k para o qual G é *k -colorível*. (Boundy and Murty (2010))

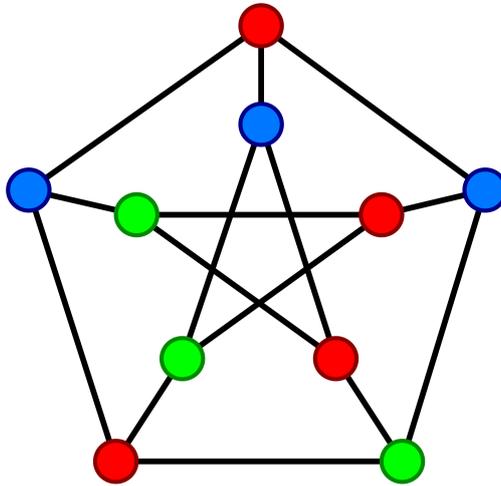


Figura 1: Coloração de vértices adequada no Grafo de Petersen, grafo cujo número cromático é 3 (Fonte: [Wikipedia contributors \(2024\)](#))

Encontrar uma k -coloração de um grafo G com $k = \chi(G)$ é um problema \mathcal{NP} -difícil: não se conhece um algoritmo que o resolva em tempo polinomial. Considerando esse entrave, algoritmos que constroem soluções sub-ótimas em tempo razoável são utilizados para resolver instâncias do GCP. Tais algoritmos utilizam as mais diversas abordagens em seu funcionamento: heurísticas gulosas, métodos de busca local, abordagens evolutivas e métodos híbridos.

Técnicas de *Deep Learning*, cuja aplicação é bem-sucedida em tarefas de diversos domínios, também vêm sendo desenvolvidas e utilizadas para a resolução de problemas relacionados a grafos, inclusive o GCP. Dentre elas destacam-se as *Graph Neural Networks*, redes neurais artificiais construídas especificamente para lidar com problemas estruturados em grafos. Muitas dessas técnicas recaem na categoria de *Aprendizado Supervisionado*, em que é necessário um conjunto de dados rotulado para a construção de modelos de aprendizado que operem com elas. Por outro lado, o *Aprendizado por Reforço*, uma outra

abordagem do Aprendizado de Máquina, reúne técnicas e métodos que não necessitam de um conjunto de dados rotulado para a construção de modelos de aprendizado. Mais recentemente, devido ao sucesso das diversas aplicações de *Deep Learning*, a área de Aprendizado por Reforço também começou a beneficiar-se delas, fazendo emergir um novo campo de estudo e aplicação: *Deep Reinforcement Learning*. No caso do GCP, abordagens baseadas em *Deep Learning*, Aprendizado por Reforço e *Deep Reinforcement Learning* surgiram como alternativas aos métodos clássicos de abordar o problema.

O estudo de técnicas de *Deep Learning*, de Aprendizado por Reforço e de *Deep Reinforcement Learning* para a obtenção de soluções sub-ótimas para instâncias do GCP é bastante adequado para a introdução de um estudante a diferentes ramos do conhecimento matemático e computacional. É um tópico que exige familiaridade com conceitos de Teoria dos Grafos, *Machine Learning*, *Deep Learning* e Aprendizado por Reforço, além de outros fundamentos e conceitos subjacentes a esses ramos do conhecimento.

2 Objetivos

Um exemplo do cenário acima explorado é o algoritmo descrito em [Watkins et al. \(2023\)](#), chamado de **ReLCol** e construído sob a junção de técnicas de *Deep Reinforcement Learning* e de *Deep Learning* aplicado a grafos. Dessa forma, o objetivo deste projeto de pesquisa é a exploração de possíveis maneiras de melhorar o desempenho do referido algoritmo. Os objetivos secundários do projeto são:

- Verificar a possibilidade de melhoria no desempenho de **ReLCol** utilizando métodos de Aprendizado Supervisionado, mais especificamente *Graph Neural Networks*, construídos especificamente para a resolução de instâncias do GCP.
- Verificar a possibilidade de melhoria no desempenho de **ReLCol** utilizando outras abordagens de *Deep Reinforcement Learning* distintas da abordagem utilizada no algoritmo.

3 Material e métodos

Os principais passos envolvidos no desenvolvimento deste projeto são:

- Estudo de fundamentos e ferramentas necessárias para o desenvolvimento do projeto
- Familiarização com a implementação do **ReLCol**
- Estudo e implementação de modificações no **ReLCol** buscando a melhoria de seu desempenho.
- Análise dos resultados e elaboração de relatórios

3.1 Estudo de conceitos e fundamentos

Este projeto pretende utilizar métodos de *Deep Learning* (especialmente *Deep Reinforcement Learning*) para solucionar instâncias de um problema clássico de Teoria dos Grafos. Dessa forma, é esperado um conhecimento mais aprofundado de conceitos de Teoria dos Grafos (Boundy and Murty (2010)), de *Deep Learning* (Goodfellow et al. (2016)) e de Aprendizado por Reforço (Sutton and Barto (2018)). Cabe ressaltar que o estudante já está familiarizado com a área de *Deep Learning* e com conceitos de Teoria dos Grafos pois, além de ter cursado disciplinas relativas a essas áreas do conhecimento ao longo da graduação, ele participa de um grupo de estudos sobre Aprendizado de Máquina coordenado pela orientadora, onde frequentes discussões sobre métodos e métricas de *Deep Learning* ocorrem.

Na área de *Deep Learning* aplicada a grafos e *Deep Reinforcement Learning*, o conhecimento das seguintes ferramentas é necessário:

Graph Neural Networks (GNNs): São redes neurais especificamente projetadas para lidar com grafos e problemas estruturados em grafos. Elas aprendem representações vetoriais de vértices e arestas de grafos (*embeddings*) direcionadas ao cumprimento das tarefas para as quais são utilizadas, como classificação de arestas. A construção de tais *embeddings* se dá por um mecanismo conhecido por *Message Passing*: em uma camada de GNN, a *embedding* de um determinado vértice do grafo é refinada com a utilização das *embeddings* dos vértices vizinhos a ele, além da *embedding* anterior do próprio vértice. Isso permite que, pela captura de informações de localidade existentes no grafo, as GNNs aprendam a construir representações vetoriais mais acuradas dos elementos constituintes dos grafos. (Hamilton (2020))

Para o estudo de GNNs, os seguintes materiais são recomendados:

- Livro *Graph Representation Learning* (Hamilton (2020))
- Curso *Machine Learning with Graphs*, ministrado na Universidade de Stanford em 2021 (<https://snap.stanford.edu/class/cs224w-2021/>), cujas aulas estão disponíveis no YouTube (<https://www.youtube.com/playlist?list=PLoROMvodv4rPLKxIpqhjhPgdQy7imNkDn>)

Deep Reinforcement Learning: Engloba técnicas baseadas em redes neurais para a solução de problemas em Aprendizado por Reforço. Começaram a ser utilizadas com mais frequência a partir de 2013, após a publicação do artigo *Playing Atari with Deep Reinforcement Learning* (Mnih et al. (2013)), em que é apresentada uma rede neural profunda (*Deep Q-Network - DQN*) que aprendeu, com sucesso, a jogar diversos jogos da plataforma de jogos eletrônicos *Atari*. A DQN é inspirada em um método clássico de Aprendizado por Reforço chamado *Q-Learning* (Watkins (1989)) e sofreu melhorias

em 2015, apresentadas em [Mnih et al. \(2015\)](#). O algoritmo descrito no artigo-base deste projeto é baseado na versão melhorada da DQN.

Para o estudo de conceitos de Aprendizado por Reforço e de *Deep Reinforcement Learning*, os seguintes materiais são recomendados:

- Livro *Reinforcement Learning* ([Sutton and Barto \(2018\)](#))
- Curso *Introduction to Reinforcement Learning*, ministrado pela *DeepMind* em 2021 e cujas aulas estão disponíveis no YouTube (<https://www.youtube.com/playlist?list=PLqYmG7hTraZDVH599EIt1EWsU0sJbAodm>).
- Artigos *Playing Atari with Deep Reinforcement Learning* ([Mnih et al. \(2013\)](#)) e *Human-level control through deep reinforcement learning* ([Mnih et al. \(2015\)](#))

3.2 Familiarização com a implementação do algoritmo ReLCol

Em [Watkins et al. \(2023\)](#), está disponível um *link* para um repositório no *GitHub* com a implementação do **ReLCol**. Essa implementação será estudada detalhadamente para que modificações sejam posteriormente realizadas nela.

3.3 Possíveis caminhos de melhorias para ReLCol

O objetivo dos problemas solucionáveis por ferramentas de Aprendizado por Reforço é aprender uma política ótima de tomada de decisões, isto é, uma política que resulte na maior quantidade de recompensa para o agente que toma tais decisões. Essa política pode ser aprendida de forma indireta ou de forma direta.

Algoritmos baseados em *Q-Learning* melhoram progressivamente estimativas da recompensa recebida pelo agente para cada estado em que ele possa estar e para cada ação que ele possa tomar, aprendendo a política ótima de forma indireta. O *Q-Learning* clássico armazena as estimativas em uma grande tabela de pares estado-ação, e a **DQN** busca aproximar os valores dessa tabela com a utilização de uma rede neural. Mais detalhadamente, a **DQN** contém uma CNN (*Convolutional Neural Network*) para extrair representações relevantes das imagens dos jogos de *Atari* que são fornecidas como entrada à rede. No algoritmo **ReLCol**, baseado em **DQN**, o papel da CNN é executado por uma GNN, e um dos pontos de melhoria possíveis para **ReLCol** é alterar a arquitetura dessa GNN para arquiteturas de GNNs construídas especificamente com o intuito de lidar com o Problema de Coloração de Grafos, como as descritas em [Li et al. \(2022\)](#), [Ijaz et al. \(2022\)](#) e [Zhang et al. \(2024\)](#).

Algoritmos baseados em *Policy Gradient* utilizam as recompensas recebidas para alterar as probabilidades de o agente executar uma determinada ação estando em um

determinado estado, aprendendo a política ótima de forma direta. Um outro modo de melhorar o desempenho de **ReLCol** é adaptá-lo para a referida abordagem de Aprendizado por Reforço. Abordar problemas clássicos de Teoria dos Grafos e Otimização Combinatória utilizando *Policy Gradient* não é uma tarefa inédita: [Bello et al. \(2016\)](#) utiliza *Policy Gradient* e Redes Neurais Recorrentes (RNNs) para resolver instâncias do Problema do Caixeiro Viajante (TSP, do inglês *Travel Salesman Problem*).

3.4 Parte experimental

A fase experimental do projeto contempla atividades de implementação e teste do algoritmo original e do algoritmo modificado com as melhorias propostas. É o momento em que as hipóteses de melhoria do **ReLCol** serão validadas e avaliadas.

Na primeira etapa, o estudante deverá familiarizar-se com a implementação de algoritmos de *Deep Reinforcement Learning* e, conseqüentemente, consolidar o conhecimento prático sobre os processos de treinamento e validação dos mesmos. Será explorado um problema clássico na área de Aprendizado por Reforço: o problema do *CartPole*, melhor explicado no *site* da plataforma *Gymnasium*: https://gymnasium.farama.org/environments/classic_control/cart_pole/.

Na segunda etapa, o estudante deverá testar a implementação original do **ReLCol** disponível em repositório do *GitHub* cujo *link* está em [Watkins et al. \(2023\)](#). A partir do sucesso desse teste e do entendimento da referida implementação, as melhorias no **ReLCol** serão implementadas e testadas seguindo o formato experimental disponível em [Watkins et al. \(2023\)](#), até mesmo para possibilitar a comparação de resultados de desempenho com a implementação original.

Durante o processo de experimentação, o planejamento dos experimentos, ajustes na implementação, avaliação dos resultados, e outros problemas que possam surgir serão amplamente discutidos. Ao final, espera-se que seja alcançado um bom grau de entendimento sobre o potencial da utilização de técnicas de *Deep Reinforcement Learning* para a resolução de problemas clássicos na área de Otimização Combinatória.

4 Plano de trabalho e cronograma de execução

1. Estudo de fundamentos: Corresponde às atividades descritas na seção [3.1](#)
2. Entendimento e testes de métodos básicos de *Deep Reinforcement Learning* e da implementação original do **ReLCol**
3. Estudo dos artigos científicos que apresentam arquiteturas de GNN próprias para o Problema da Coloração de Grafos

4. Implementação de possíveis melhorias pela alteração da GNN presente no **ReLCoI**
5. Estudo e adaptação do **ReLCoI** para a abordagem *Policy Gradient*
6. Confecção da monografia

Um cronograma aproximado para a execução dessas atividades está apresentado a seguir.

Atividade	Mês											
	4	5	6	7	8	9	10	11	12			
1	x	x										
2	x	x										
3			x									
4				x	x							
5					x	x	x					
6			x	x	x	x	x	x	x	x	x	x

5 Forma de análise dos resultados

Os resultados da pesquisa serão avaliados em termos dos seguintes produtos ou atuações resultantes como decorrência direta da execução do projeto de pesquisa:

- Disponibilização pública dos códigos desenvolvidos.
- Submissão ou publicação do trabalho em eventos ou outros veículos pertinentes.
- Apresentação dos resultados em congressos e eventos científicos.
- Divulgação dos conhecimentos adquiridos através de atividades no ambiente universitário, como atividades em grupos de extensão ou palestras.

Referências

- Bello, I., Pham, H., Le, Q. V., Norouzi, M., and Bengio, S. (2016). Neural combinatorial optimization with reinforcement learning. *CoRR*, abs/1611.09940.
- Boundy, J. A. and Murty, U. S. R. (2010). *Graph Theory*. Springer.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Hamilton, W. L. (2020). Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.
- Ijaz, A. Z., Ali, R. H., Ali, N., Laique, T., and Ali Khan, T. (2022). Solving graph coloring problem via graph neural network (gnn). In *2022 17th International Conference on Emerging Technologies (ICET)*, pages 178–183.

- Li, W., Li, R., Ma, Y., Chan, S. O., Pan, D., and Yu, B. (2022). Rethinking graph neural networks for the graph coloring problem.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.
- Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, University of Cambridge.
- Watkins, G., Montana, G., and Branke, J. (2023). Generating a graph colouring heuristic with deep q-learning and graph neural networks.
- Wikipedia contributors (2024). Petersen graph — wikipedia, the free encyclopedia. Accessed: 2025-04-20.
- Zhang, Y., Zhang, K., and Wu, N. (2024). Using graph isomorphism network to solve graph coloring problem. In *2024 4th International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 209–215.