Problema de Dominação Romana

Ieremies Vieira da Fonseca Romero

Resumo

A Dominação Romana é um problema proposto por Stewart [14] em que desejamos defender o império romano dispondo de um certo número de legiões. Para isso, cada cidade deve ser assegurada de forma que uma legião alocada ou seja vizinha de outra que possua duas legiões. Assim, deseja-se minimizar a quantidade de legiões distribuídas sem abdicar da segurança do império.

Neste projeto, utilizaremos a técnica de Programação Linear Inteira (PLI) para modelar o problema. Objetivamos propor novos modelos de PLI para dominação romana e suas variantes, explorando técnicas como branch-cut, branch-price e ferramental moderno de PLI.

1 Introdução

Durante a Segunda Guerra Mundial, General Douglas MacArthur propôs uma estratégia de movimentação que consistia em avançar suas tropas de uma ilha para outra apenas quando ele poderia deixar para trás um número suficiente de tropas ([14]). Ele não foi o primeiro a utilizar dessa estratégia: segundo Stewart [14], referências históricas apontam que o Imperador Constantino, no quarto século A.C., aplicou estratégia similar para defender o Império Romano de invasões dos povos ditos "bárbaros".

Para exemplificar o seu uso, considere o mapa do Império Romano simplificado na Figura 1. Nesse exemplo, o imperador possui 4 legiões para serem distribuídas pelo território e ele deseja fazê-lo de forma que todas as cidades sejam consideradas seguras. Uma região é dita segura, ou coberta, se há uma legião em seu território ou se está conectada a outra região com duas legiões.

./mapa.jpg

Figura 1: Representação do mapa do Império Romano usada como ilustração do problem, inspirada de Stewart [14].

As aplicações vão além do universo militar: ao alocar estações de serviços de emergência, uma lógica similar à "segurança", acima descrita, é muito útil [7]. Além disso, redes sem fio "ad hoc", onde um conjunto de usuários com conexões sem fio formam uma rede temporária, sem necessidade de autoridade central ou infraestrutura permanente, constituem um outro bom exemplo [15].

1.1 Modelo matemático

Para um grafo $G = (V, E)$, dizemos que a **vizinhança aberta** $N(v)$ de um vértice v é definida como o conjunto de vértices adjacentes a v em G, ou seja, $N(v) = \{u|(u, v) \in E\}$. Similarmente, dizemos que a **vizinhança fechada** $N[v]$ de um vértice v é a vizinhança aberta incluindo o próprio v, ou seja, $N[v] = N(v) \cup \{v\}$. Para um conjunto de vértices s, a vizinhança aberta desse conjunto é a união das vizinhanças abertas de cada um dos seus vértices (o respectivo pode ser dito para a vizinhança fechada). Um **conjunto dominante** de um grafo G é um subconjunto de vértices D tal que a vizinhança fechada de D é o próprio conjunto V . Por sua vez, o **número de dominação** de um grafo G, dito $\gamma(G)$, é a cardinalidade do menor conjunto dominante do grafo G.

O problema de dominação romana é definido em um grafo $G = (V, E)$ simples, finito e não-direcionado, no qual cada vértice representa uma cidade ou região do império e as arestas são as conexões entre elas [2]. Diremos que uma **função de dominação romana** é uma função $f : V \rightarrow z0, 1, 2\}$ na qual $f(v)$ indica a quantidade de legiões naquela região,de forma que, para qualquer v tal que $f(v) = 0$, deve existir u vizinho a v cujo $f(u) = 2$. Definimos o **número de dominação romana** total de um grafo G como o menor valor $f(v), \forall v \in V$, tal que f é uma função de dominação romana do grafo G.

1.2 Revisão bibliográfica

Após a descrição inicial do problema, ReVelle e Rosing [12] apresentaram o desenvolvimento inicial em teoria de grafos. Além disso, Cockayne et al. [2] apresentaram alguns resultados importantes de teoria de grafos sobre o problema, como limitantes e propriedades da função de dominação romana, os quais foram estendidos e aprimorados por Xing, Chen e Chen [16], Favaron et al. [6], Mobaraky e Sheikholeslami [11]. Algumas classes especiais de grafos podem ser resolvidas em tempo linear, mas, no caso geral, o problema é NP-difícil ([5, 10, 13]).

Ivanovic [8] utilizaram *Variable Neightborhood Search* (VNS) no mesmo problema, obtendo resultados interessantes para as instâncias propostas por Currò [3]. Essa meta-heurística parte da ideia de que soluções ótimas são encontradas "próximas"de boas soluções. Assim, é possível utilizar busca local e eventuais técnicas de perturbação, para escapar de mínimos locais.

Já Khandelwal, Srivastava e Saran [9] utilizaram algoritmos genéticos no problema de dominação romana, uma ideia que toma de inspiração da evolução das espécies observadas na natureza. Partindo de um conjunto de soluções, realizamos "cruzamentos"das melhores para produzir novas gerações. A cada uma, induzimos "mutações"aleatórias que alteram certos pontos das soluções, espelhando a realidade e tentando evitar cair em mínimos locais.

2 Metodologia

Programação Linear é uma técnica de otimização de problemas a partir da modelagem dos mesmos em **programas lineares**. Nestes, definimos uma função objetivo, a qual queremos maximizar ou minimizar com suas variáveis sujeitas a um conjunto de restrições lineares (equações ou inequações lineares) [1] . Todo programa linear pode ser escrito em sua forma canônica:

$$\begin{aligned} &\text{maximize } cx \\ &\text{sujeito a} \quad Ax \leq b \\ &\qquad\qquad\quad\; x \in \mathbb{R}_{+} \end{aligned}$$

Perceba que maximizar uma função é o mesmo que minimizar a mesma com sinal oposto.

Para encontrar soluções viáveis com valores ótimos, conhecemos o algoritmo **simplex** que, apesar de ter complexidade exponencial, no caso médio possui comportamento polinomial.

Para alguns problemas, como o de dominação romana, não faz sentido falar em soluções fracionárias, afinal, não conseguimos "alocar meia legião". Para isso, restringimos as variáveis aos inteiros, fazendo assim um **Programa Linear Inteiro**. Caso apenas um subconjunto das variáveis possuam a restrição de integralidade, chamamos esse programa de linear misto.

O que a princípio pode parecer uma pequena alteração, torna o problema computacionalmente ainda mais complexo. Para encontramos boas soluções viáveis para esse tipo de programa, algoritmos como o simplex não são o suficiente. Para isso, utilizamos técnicas como **branch-and-bound**, que consiste em dividir o problema em subproblemas menores e, durante o processo, encontrar limitantes que permitam diminuir o espaço de busca.

Variações como **branch-and-cut**, na qual, ao atingir soluções não inteiras na relaxação linear usando o simplex, utilizamos algoritmos de plano de cortes para adicionar mais restrições até a solução fornecida pelo simplex na RL for inteira. Já para **branch-and-price**, essa técnica advém da observação que, para grandes problemas, grande parte das variáveis permanecem nulas entre as interações do *branch-and-bound*. Assim podemos inseri-las conforme progredimos na nossa busca utilizando técnicas de geração de colunas. fonte?

Ivanovic [8] propôs duas formulações, como a descrita a baixo, para o problema de dominação romana. Nesta, para cada vértice i, existem duas variáveis associadas: x_i real não-negativa e y_i binária.

- Se $x_i < 1$ e $y_i = 0$, então $f(i) = 0$.
- Se $x_i \geq 1$, então $f(i) = 1$.
- Se $y_i = 1$, então $f(i) = 2$.

$$
\begin{aligned}
\text{minimize} \; & \sum_{i \in V} x_i + 2 \sum_{i \in V} y_i \\
\text{sujeito a} \quad & x_i + y_i + \sum_{j \in N_i} y_j \geq 1 && i \in V \\
& x_i \in \mathbb{R}_+ && i \in V \\
& y_i \in \{0, 1\} && i \in V
\end{aligned}
$$

Perceba que soluções onde $x_i > 1$ e $y_i = 1$ são viáveis, mas Ivanovic [8] mostra que elas não serão ótimas.

3 Objetivos

Neste projeto, objetivamos propor novos modelos de PLI para dominação romana e suas variantes explorando técnicas como **branch-and-cut** e **branch-and-price**. Além disso, estudaremos a possibilidade de novos cortes e limitantes para as formulações.

Os algoritmos e modelos propostos serão comparados com as instâncias presentes na literatura, como em Currò [3] e, se necessário novas instâncias poderão ser geradas.

Os resultados dos experimentos computacionais serão comparados utilizando técnicas como **Performance Profile** demonstrado por Dolan e Moré [4].

Referências

[1] Vasek Chvatal, Vaclav Chvatal e Vasek Chv?atal. *Linear Programming*. W. H. Freeman, 15 de set. de 1983. 500 pp. ISBN: 978-0-7167-1587-0. Google Books: `DN20\_tW\_BV0C`.

[2] Ernie J Cockayne et al. "Roman domination in graphs". Em: *Discrete Mathematics* 278.1-3 (mar. de 2004), pp. 11–22. ISSN: 0012-365X. DOI: `10.1016/j.disc.2003.06.004`. (Acedido em 15/07/2022).

[3] Vincenzo Currò. "The Roman Domination Problem on Grid Graphs". Em: (2014), p. 96.

[4] Elizabeth D. Dolan e Jorge J. Moré. "Benchmarking optimization software with performance profiles". Em: *Mathematical Programming* 91.2 (1 de jan. de 2002), pp. 201–213. ISSN: 0025-5610, 1436-4646. DOI: `10.1007/s101070100263`. (Acedido em 10/06/2022).

[5] Paul Andrew Dreyer. "Applications and variations of domination in graphs". Tese de doutoramento. Ann Arbor, United States, 2000. 178 pp. ISBN: 9780599993846. URL: `https://www.proquest.com/docview/304623075/abstract/392C1D0079F84EE6PQ/1` (acedido em 15/07/2022).

[6] O. Favaron et al. "On the Roman domination number of a graph". Em: *Discrete Mathematics* 309.10 (mai. de 2009), pp. 3447–3451. ISSN: 0012-365X. DOI: 10.1016/j.disc.2008.09.043. (Acedido em 15/07/2022).

[7] Alireza Ghaffari-Hadigheh. "Roman domination problem with uncertain positioning and deployment costs". Em: *Soft Computing* 24.4 (1 de fev. de 2020), pp. 2637–2645. ISSN: 1432-7643, 1433-7479. DOI: 10.1007/s00500-019-03811-z. (Acedido em 13/09/2022).

[8] Marija Ivanovic. "Improved mixed integer linear programing formulations for roman domination problem". Em: *Publications de l'Institut Mathematique* 99.113 (2016), pp. 51–58. ISSN: 0350-1302, 1820-7405. DOI: 10.2298/pim1613051i. (Acedido em 15/07/2022).

[9] Aditi Khandelwal, Kamal Srivastava e Gur Saran. "On Roman Domination of Graphs Using a Genetic Algorithm". Em: *Computational Methods and Data Engineering*. Ed. por Vijendra Singh et al. Advances in Intelligent Systems and Computing. Singapore: Springer, 2021, pp. 133–147. ISBN: 9789811568763. DOI: 10.1007/978-981-15-6876-3_11.

[10] Aneta Klobucar e Ivona Puljic. "Some results for Roman domination number on Cardinal product of paths and cycles". Em: *Kragujevac Journal of Mathematics* 38.1 (2014), pp. 83–94. ISSN: 1450-9628. DOI: 10.5937/kgjmath1401083k. pmid: 1450-96281401083K. (Acedido em 15/07/2022).

[11] B.P. Mobaraky e Seyed Sheikholeslami. "Bounds on Roman Domination Numbers of Graphs". Em: *Matematichki Vesnik* 60 (1 de jan. de 2008).

[12] Charles S. ReVelle e Kenneth E. Rosing. "Defendens Imperium Romanum: A Classical Problem in Military Strategy". Em: *The American Mathematical Monthly* 107.7 (1 de ago. de 2000), pp. 585–594. ISSN: 0002-9890, 1930-0972. DOI: 10.1080/00029890.2000.12005243. (Acedido em 15/07/2022).

[13] Weiping Shang e Xiaodong Hu. "The Roman Domination Problem in Unit Disk Graphs". Em: *Computational {{Science}} – {{ICCS}} 2007*. Ed. por Yong Shi et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007, pp. 305–312. ISBN: 978-3-540-72588-6. DOI: 10.1007/978-3-540-72588-6_51.

[14] Ian Stewart. "Defend the Roman Empire!" Em: *Scientific American* 281.6 (dez. de 1999), pp. 136–138. ISSN: 0036-8733. DOI: 10.1038/scientificamerican1299-136. URL: https://www.scientificamerican.com/article/defend-the-roman-empire (acedido em 15/07/2022).

[15] Jie Wu e Hailan Li. "Domination and its applications in ad hoc wireless networks with unidirectional links". Em: *Proceedings 2000 International Conference on Parallel Processing*. Proceedings 2000 International Conference on Parallel Processing. IEEE Comput. Soc, ago. de 2000, pp. 189–197. DOI: 10.1109/icpp.2000.876117.

[16] Hua-Ming Xing, Xin Chen e Xue-Gang Chen. “A note on Roman domination in graphs”. Em: *Discrete Mathematics* 306.24 (28 de dez. de 2006), pp. 3338–3340. ISSN: 0012-365X. DOI: 10.1016/j.disc.2006.06.018. (Acedido em 15/07/2022).