

Um algoritmo para a resolução do problema de designação de disciplinas a professores, em um contexto universitário

Miguel Rodrigues Guimarães de Oliveira¹

Mayron César de Oliveira Moreira²

Andreza Cristina Beezão Moreira³

Resumo: O problema de designação é um clássico na área de Otimização Combinatória. Uma de suas vertentes se trata da designação de professores (TAP, do inglês *Teacher assignment problem*), que consiste em distribuir um conjunto de disciplinas para um determinado grupo de professores, considerando uma série de regras, como o número mínimo de horas semanais que um professor deve lecionar, regras trabalhistas, acordos institucionais/departamentais, entre outras. Além disso, costuma-se também considerar as preferências dos docentes, objetivando que a maior quantidade de professores lecionem matérias de seu interesse.

A designação de professores para disciplinas pode ser modelada como um problema de programação inteira, composto por variáveis binárias [1]. Problemas dessa natureza se enquadram, em geral, na classe dos problemas NP-difíceis, conforme indicam os trabalhos [2], [3], [4]. Isso significa que não se conhece um algoritmo exato e polinomial e, por isso, é importante a proposição de abordagens que visem tanto gerar quanto validar soluções para tal problema.

Por ter um caráter muito combinatório, o TAP pode, também, ser modelado através de um conjunto de restrições. Optamos, portanto, por implementar um algoritmo baseado na técnica de programação por restrição, ou *Constraint Programming (CP)*, em inglês. Esta técnica tem conseguido espaço como algoritmo para solucionar problemas de designação, em geral. Como exemplo disso, [5] usam CP para organizar o agendamento de uma sala de operações. Outro trabalho que aborda um problema e forma de solução semelhantes é o de [7], em que a técnica é utilizada para resolver a alocação de recursos em atividades repetitivas em contexto de construção industrial, como na construção de rodovias, ferrovias e gasoduto. Por fim, citamos o trabalho [6], também sobre ordenação em contexto acadêmico. Os autores usaram *solution-based phase saving* e *hot starts*, os quais fornecem, respectivamente, um direcionamento para o solver da região próxima da melhor solução encontrada e um ponto de partida para o algoritmo que utilizou programação por restrição. Com essas ferramentas foram obtidos resultados semelhantes aos de abordagens focadas em heurísticas, dentre elas a *Simulated Annealing* e *Adaptive Large Neighborhood Search*.

¹Universidade Federal de Lavras,
miguel.oliveira2@estudante.ufla.br

²Universidade Federal de Lavras,
mayron.moreira@ufla.br

³Universidade Federal de Lavras,
andreza.moreira@ufla.br

Foram utilizadas ferramentas da *Google OR-Tools* e, para testar a qualidade do algoritmo proposto, realizamos um experimento computacional utilizando dados de um departamento de uma universidade brasileira. A solução gerada gastou um tempo total de processamento de menos de 10 segundos, com 57.89% das composições de turmas distribuídas para professores que as preferiam. Propomos, ainda, uma interface que permite a interação do usuário com o algoritmo e seus resultados. Como sugestão para trabalhos futuros, propomos a validação desta interface junto aos usuários, fazendo um estudo de *design* para melhor adequá-la às suas necessidades. Por fim, um ponto bastante relevante é a possível adaptação deste método de solução para outros departamentos, com condições similares de atribuição de docentes a disciplinas.

Referências

- [1] R. Karp. *Reducibility Among Combinatorial Problems*, In: Jünger, M., et al. 50 Years of Integer Programming 1958-2008, v. 40, p. 85–103, Springer, Berlin, Heidelberg, 1972. DOI:10.1007/978-3-540-68279-0₈
- [2] S. Sahni, T. Gonzalez. P-complete approximation problems, *J. ACM, Association for Computing Machinery*, 1976. DOI: 10.1145/321958.321975.
- [3] V. A. Bardadym. *Computer-aided school and university timetabling: The new wave*. In: BURKE, E.; ROSS, P. (Ed.). *Practice and Theory of Automated Timetabling*. p. 22-45, Springer, Berlin, Heidelberg, 1996. ISBN 978-3-540-70682-3.
- [4] J. HARTMANIS. Computers and intractability: A guide to the theory of np-completeness (michael r. garey and david s. johnson). v.24, n.1, p. 90–91, *SIAM Review*, 1982. DOI: 10.1137/1024022.
- [5] T. Wang, N. Meskens, D. Duvivier. Scheduling operating theatres: Mixed integer programming vs. constraint programming. v.247, n.2, p. 401–413, *European Journal of Operational Research*, 2015. ISSN 0377-2217.
- [6] Y. Tang. Scheduling optimization of linear schedule with constraint programming. *Computer-Aided Civil and Infrastructure Engineering*, 2018. DOI: 10.1111/mice.12277.
- [7] E. Demirovic, P. J. Stuckey. *Constraint programming for high school timetabling: A scheduling-based model with hot starts*. In: HOEVE, W.-J. van (Ed.). *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. p. 135–152, Springer International Publishing, 2018. ISBN 978-3-319-93031-2.