

# Resolução de um Problema de Alocação de Turmas às Salas de Aula

Maria L. T. Santos<sup>1</sup>; Eduardo D. Bernardes<sup>2</sup>

Departamento de Ciências Exatas e Tecnológicas, DCET, UESC, Ilhéus, BA

O problema de alocação de salas de aula é comum em diversas instituições de ensino, principalmente nas de Ensino Superior, conforme observado na literatura relacionada. Problemas desse tipo fazem parte da categoria *NP-Hard* [9] e podem ser resolvidos através de métodos exatos ou métodos heurísticos. Além disso, pode apresentar diferentes características próprias dependendo do caso de estudo, já que as universidades apresentam edificações e organizações distintas.

Para a resolução de um problema de alocação de salas, é importante considerar aspectos e informações relevante, tais como a capacidade dos locais a serem utilizados, os horários disponíveis, a demanda solicitada e geralmente não permitindo choques de grupos em um local no mesmo horário. Na maioria dos casos, busca-se maximizar o número de alocações de turmas para um melhor aproveitamento do espaço físico disponível, mas outros objetivos também podem ser considerados, como a redução do consumo de energia [1] e redução do deslocamento dos grupos [8].

De forma geral, a resolução desse problema é obtida de forma manual, com funcionários sendo responsáveis por analisar todas as turmas, com seus respectivos horários e demandas, a serem alocadas para as salas de aula disponíveis, cada uma com sua capacidade máxima específica. Esse processo de resolução é extremamente exaustivo e demanda uma quantidade de tempo que pode ser reduzida com o uso de estratégia para resolução de problemas de otimização, possibilitando a obtenção de melhores soluções.

O propósito deste trabalho é tratar o problema de alocação de turmas às salas de aula do Departamento de Ciências Exatas e Tecnológicas (DCET) da Universidade Estadual de Santa Cruz (UESC), mostrando como a otimização e a tecnologia podem auxiliar a tomada de decisões, além de oferecer uma alternativa mais eficaz que o processo manual atualmente utilizado.

Para resolução do problema proposto, dois modelos de Programação Linear Inteira foram desenvolvidos, além de uma heurística construtiva. O Modelo 1 busca realizar a alocação das turmas às salas de aula de maneira que salas diferentes possam ser estabelecidas à turma, quando esta apresentar aulas em mais de um dia da semana. O Modelo 2 busca atribuir apenas uma sala de aula para a turma independentemente da distribuição das aulas ao decorrer da semana. A heurística construtiva, baseada em [3], foi desenvolvida considerando o caso do Modelo 2, onde apenas uma sala é utilizada por turma. Nesta estratégia, a alocação é realizada com base em uma lista de turmas ordenada de acordo com o número de aulas.

Para implementação das estratégias propostas, foram utilizadas a linguagem de programação Python 3.9 (2020), a aplicação web Jupyter Notebook (2020), IDE PyCharm (2020) e biblioteca PuLP (2020) com os *solvers* *CBC* (*COIN-OR branch and cut*) (2019), um *solver* gratuito, e *Gurobi* (2021), um *solver* comercial, ambos em suas configurações na forma padrão. Testes computacionais foram realizados considerando uma instância obtida a partir dos dados do primeiro semestre de 2019, coletados por relatórios disponibilizados por setores da própria universidade. A instância

---

<sup>1</sup>teixeirasantosml@gmail.com

<sup>2</sup>edbernardes@uesc.br

contém 310 turmas a serem alocadas nas 16 salas teóricas do pavilhão destinado às aulas do departamento.

Utilizando os *solvers* e a instância considerada, em ambos os modelos, foram encontradas soluções ótimas distintas com um total de 251 turmas alocadas, representando uma melhoria de 51,20% em relação à solução manual utilizada pela universidade, com o número de 166 turmas alocadas. Vale ressaltar que o tempo de solução foi consideravelmente menor utilizando o Gurobi. Ao resolver a mesma instância com a heurística construtiva, foi obtida uma solução com o número de 237 turmas alocadas. Desta forma, a solução obtida representa uma melhoria de 42,77% em relação à solução manual.

A análise dos resultados obtidos pelas estratégias propostas mostra que a solução utilizada pelo departamento, apesar de viável, pode ser aperfeiçoada com a utilização de técnicas de otimização e de ferramentas tecnológicas que tornam o processo mais eficiente tanto no tempo de execução quanto na qualidade das alocações, contribuindo para um maior aproveitamento do espaço físico disponível.

## Agradecimentos

À FAPESB pelo apoio financeiro.

## Referências

- [1] Raphael Medeiros. Alves, Anand. Subramanian e Alisson Vasconcelos de Brito. “Minimizando os custos energéticos de alocação de aulas a salas: o caso de uma instituição federal de ensino”. Em: **Anais do LI SBPO - Simpósio Brasileiro de Pesquisa Operacional**. Galoá. 2019.
- [2] Python Software Foundation. **Python Language Reference**. Versão 3.9. 2020. URL: <https://www.python.org>.
- [3] Junot. Freire e Rafael Augusto de Melo. “Formulações, heurísticas e um limite combinatório para o problema de alocação de salas de aula com demandas flexíveis”. Em: **Anais do XLVIII SBPO - Simpósio Brasileiro de Pesquisa Operacional**. Galoá. 2016, pp. 722–729.
- [4] LLC Gurobi Optimization. **Gurobi**. Versão 9. 2021. URL: <https://www.gurobi.com>.
- [5] JetBrains. **PyCharm**. Versão 2021.3. 2020. URL: <https://www.jetbrains.com/pt-br/pycharm/>.
- [6] Forrest John. e Lougee-Heimer Robin. **CBC (COIN-OR branch and cut)**. Versão 2.10. 2019. URL: <https://projects.coin-or.org/Cbc>.
- [7] Project Jupyter. **Jupyter Notebook**. Versão 6.1.11. 2020. URL: <https://jupyter.org/index.html>.
- [8] Rosana Maria Luvezute Kripka, Moacir. Kripka e Marilene Caproski da Silva. “Formulação para o problema de alocação de salas de aula com minimização de deslocamentos”. Em: **Anais do XLIII SBPO - Simpósio Brasileiro de Pesquisa Operacional**. 2011, pp. 1941–1951.
- [9] R. Lewis e J. Thompson. “Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem”. Em: **European Journal of Operational Research** 240.3 (2015), pp. 637–648.
- [10] Mitchell Stuart. et al. **PuLP**. Versão 2.4. 2020. URL: <https://coin-or.github.io/pulp/>.