Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 12, n. 1, 2026.

Trabalho apresentado no XLIV CNMAC, Fundação Getulio Vargas - Rio de Janeiro - RJ, 2025.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Bipartite Modeling for Solving the University Timetable Problem

Gabrielly de A. da Silva[1]
IC-UFRJ, Rio de Janeiro, RJ

The University Timetabling Problem (UTP) is a combinatorial optimization challenge classified as NP-hard. It involves allocating resources — usually professors, courses and rooms — while meeting multiple institutional constraints. Wren [2] describes it as a specific scheduling case, where the solution space grows exponentially, rendering manual resolution difficult. This complexity is compounded by the unique institutional characteristics [1]. To address the allocation challenges within the Institute of Computing (IC) at UFRJ, this work proposes a bipartite optimization model as a minimum viable product (MVP).

Unlike traditional approaches that solve the assignment of professors, courses, and rooms in a unified decision structure [3], our model separates the problem into two interconnected subproblems: (1) assigning professors to courses and (2) allocating courses to rooms. This separation offers greater flexibility, allowing each model to be used independently or jointly. In the IC case study, the output of the first model feeds into the second to produce a complete, synchronized schedule. The modular structure also facilitates adaptation to other departments, particularly as a centralized tool for improving room utilization and avoiding underuse of shared spaces. The approach emphasizes structured decision-making automation via mathematical optimization rather than computational performance.

In the **course allocation model**, the objective function maximizes expected academic performance by considering professor aptitude and the satisfaction of both students and faculty, while incorporating soft constraints. These include penalties for not meeting the minimum workload requirements for both permanent and substitute professors. The model also enforces hard constraints related to institutional rules—such as workload limits, scheduling conflicts, and qualification requirements—as well as specific conditions for manual allocations and maximum workloads. To enhance flexibility, a fictitious professor named DUMMY is introduced. As DUMMY is exempt from all constraints, the model remains feasible even when no qualified or available professor can be assigned to a given course. The final result is a mixed-integer linear programming model.

Meanwhile, the **room assignment model** uses the output of the previously defined model or the pre-established schedule planning. The main variable represents room allocation for a session. The objective function maximizes the sum of allocations while minimizing the difference between room capacity and session vacancies, with a coefficient prioritizing institutional allocations. The constraints include room usage, exclusive occupation, room type compatibility, and room preferences for first-year students, as well as restrictions on specific resources. The final modeling results in a Mixed Integer Quadratic Program, which characterizes it as a non-linear problem.

The implementation was carried out on a MacBook Air (M3, 8 GB RAM) using the Gurobi solver under an academic license. The solver operated in its default configuration, which automatically selects the most appropriate algorithm based on the problem's characteristics. Often, Gurobi uses a concurrent method, running multiple algorithms in parallel and selecting the one with the best performance — an approach that leverages multi-core processors. Additionally, a presolve phase is applied to simplify the model and enhance efficiency. The optimization runs until

[1]gabriellyas@ic.ufrj.br / gabriellydandrade@gmail.com

a convergence criterion is met, ensuring that the solution is within an acceptable tolerance. This study considered only the best solution identified by the solver.

The optimization process was performed in Python using the GurobiPy library, with separate projects developed for each model. Best software engineering practices ensured modularity, version control, technical documentation, environment variables, virtual environments, and automated unit testing, enhancing maintainability and scalability. Google Sheets was used to manage input data for both models, with access via the Google Sheets API and a local cache for offline use. Data was processed and standardized with pandas before being fed into the solver, and results were exported as CSV files. A Streamlit web interface was developed for interactive visualization of the results.

The course allocation model included 27,509 constraints, 3,725 integer variables (3,680 binary), and 67,182 non-zero coefficients. It reached an optimal solution in 0.03 seconds using a combination of primal and dual simplex, heuristics, and branch and bound. In the final allocation, 80 sessions were distributed across 39 mandatory courses, 40 service courses, and one elective. All assigned professors were qualified, with most teaching two courses; only two had no assignments, indicating opportunities for reallocation to elective or graduate courses.

The room allocation model, formulated as a Mixed Integer Quadratic Program (MIQP), included 19,482 constraints, 22,080 variables (15,180 integer, including 8,740 binary), and 8,740 quadratic terms. Gurobi solved it in 0.05 seconds, employing the concurrent MIP strategy with primal and dual simplex methods, heuristics, branch and bound, and branch and cut techniques. The final solution assigned 76 classes, ensuring that no room had a capacity lower than required. Some rooms, however, had more capacity than needed, indicating that room selection could be further optimized.

Discussion A unified allocation model would not adequately address the needs of the case study, which aims to support multiple institutes beyond the IC. Therefore, dividing the process into two stages provided the necessary clarity and flexibility, making the model more adaptable to diverse institutional contexts. The course allocation model, being linear, is easier to solve, while the room allocation model is quadratic and more computationally demanding. This separation also simplifies debugging and solver execution. Despite the complexity, both models were solved in seconds due to Gurobi's intelligent architecture, which includes internal optimizations and a powerful pre-solver that reduced the problem size by 90%, enabling rapid convergence. However, the two-stage approach brings challenges, such as increased implementation time and the need for careful integration between models, as separating course allocation from room assignment may lead to suboptimal results. As future work, practical testing with professors and administrative staff will contribute to refining the model based on real needs, while full implementation and expanded evaluation—especially for room allocation—will support continued improvements.

Referências

[1] F. Trigos e R. Coronel. "A transdisciplinary approach to course timetabling an optimal comprehensive campus application". Em: **Product Management and Development** 21.1 (2023). DOI: 10.4322/pmd.2023.006.

[2] A. Wren. "Scheduling, timetabling and rostering — A special relationship?" Em: **Practice and Theory of Automated Timetabling**. Springer, 2005, pp. 46–75. ISBN: 978-3-540-70682-3. DOI: 10.1007/3-540-61794-9_51.

[3] F. Zabidee e M. H. M. Adnan. "Optimization in University Student Timetables: A Comprehensive Literature Review". Em: **Journal of Advanced Research in Applied Sciences and Engineering Technology** 41.1 (2024), pp. 14–43. DOI: 10.37934/araset.41.1.1443.