



Applications of Operations Research in Real-Life Scenarios: A Literature-Based Analytical Study

Mruga Patel , Navreen Kaur , Megh Thakkar, Mohak Agarwal , Mohit Issrani

Topic: University Faculty Workload Allocation

Operations Research Project Report



[https://doi.org/ 10.55041/ijst.v2i10.054](https://doi.org/10.55041/ijst.v2i10.054)

Cite this Article: Patel, M., Kaur, N., Thakkar, M., Agarwal, M. & Issrani, M. (2026). Applications of Operations Research in Real-Life Scenarios: A Literature-Based Analytical Study. *International Journal of Science, Strategic Management and Technology*, 02(10), 1-9.
<https://doi.org/10.55041/ijst.v2i10.054>

License:  This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

1. Introduction to Operations Research

Operations Research (OR) is the discipline of applying mathematical models, optimisation techniques and analytical methods to help organisations make better decisions. It began in military logistics in World War II and is now used in manufacturing, healthcare, transport, finance, energy and education. An OR study typically follows a fixed path: define the real problem, collect data, build a mathematical model, solve it with a suitable technique, test the solution against reality, and implement it.

Most OR models have three building blocks:

- Decision variables: the choices the decision-maker controls (for example, which lecturer teaches which course).
- Objective function: a single measure to maximise or minimise (for example, total satisfaction or the imbalance in workload).
- Constraints: the limits that any acceptable solution must respect (for example, maximum teaching hours or the rule that every course needs a teacher).

Common OR techniques include linear programming (LP), integer programming (IP), transportation and assignment models, network models such as PERT/CPM, queuing, simulation, and heuristics or metaheuristics for problems too

large to solve exactly. This project examines how two of these, the assignment model and integer linear programming, apply to a problem every university faces: deciding who teaches what.

2. The Problem: Faculty Workload Allocation

Before every semester, a department head or timetable coordinator must assign faculty members to the courses and sections being offered. Done by hand, this is slow and error-prone because many things must be balanced at once:

- Every course or section must have a qualified teacher;
- No lecturer should be overloaded, and no lecturer should sit underused;
- Lecturers differ in expertise, seniority, administrative duties and personal preferences;
- Timetable clashes must be avoided;
- The result should feel fair to staff, as complaints about unequal loads are common.

Industry/domain: Higher education (public services and human resource planning).

Type of problem: At its core, this is an assignment problem, a special case of the transportation problem in which each task goes to exactly one resource. Because real universities add extra rules (a lecturer may teach several courses, loads must be capped, preferences matter), the practical version is a binary (0-1) integer linear programming problem, closely related to the generalised assignment problem. A decision is either made (1) or not (0), so ordinary continuous LP is not enough and integer methods are needed.

Why OR is suitable: The problem has a clear objective, clear limits and a large number of possible combinations. Even a small department with 20 lecturers and 20 courses has more possible one-to-one assignments than any person could check by hand. OR models search this space systematically, guarantee that every rule is met, and can find a provably best solution in seconds.

3. Selection of Research Articles

Two journal papers were selected because both deal directly with assigning faculty to courses in a real university department using integer programming, and they differ usefully in focus (workload balance and validation in one, faculty preferences and policy rules in the other).

	Paper 1	Paper 2
Title	Development and validation of an integer linear programming model for the lecturer-to-course assignment problem	Solving Faculty-Course Allocation Problem Using Integer Programming Model
Authors	Mhalapitsa et al.	Torres, M. C., Villegas, K. K. S., and Gavina, M. K. A.
Journal	Journal of Optimization in Industrial Engineering, Issue 34, Spring 2023	Philippine Journal of Science, 150(4), 679-689, August 2021

	Paper 1	Paper 2
Setting	University engineering department	University department (University of the Philippines Los Baños)
Main focus	Balancing workload fairly and validating the model	Honouring faculty preferences and non-teaching duties
Technique	Binary ILP, validated with the Delphi method	Binary integer programming with preference constraints

A third study, from UiTM Seremban in Malaysia, is used only as a supplementary illustration of how such a model is run in Excel Solver, and an earlier zero-one formulation by Mohamad (2006) is used as background on how the classical transportation/assignment model was extended.

4. Paper 1: Mhalapitsa et al. (2023)

4.1 Real-life problem and context

The study addresses the lecturer-to-course assignment problem in a university department that offers engineering programmes. In many departments, lecturers are assigned using only the number of courses each person teaches. That ignores other things that make one course heavier than another, such as the number of contact hours, the number of credits and the size of the class. The result is that some lecturers end up overworked while others are under-used, which creates unfairness and dissatisfaction among lecturers, programme coordinators and students.

4.2 Why OR techniques were suitable

The decision is a yes/no choice for each lecturer-course pair, and there are a number of workload-related constraints that have to be simultaneously satisfied. This can be formulated as a binary integer linear programming problem, which can be optimally solved. It also replaces subjective, possibly biased, human judgments with an objective, transparent algorithmic procedure.

4.3 Key findings

- The model takes into account several factors related to the workload: class size, contact hours of the course, course credits and number of courses per lecturer.
- It balances the load by allocating more courses to under-utilised lecturers and fewer to over-utilised lecturers, while still meeting the other workload constraints.
- The authors have validated the model using the Delphi method, which is a structured procedure in which a panel of experts provides feedback in consecutive rounds until a consensus is reached. This demonstrates that the model has been tested against the opinion of experts, and not only against its own arithmetic.
- The paper concludes that the approach improves fairness, saves time, provides a consistent formal procedure, and enhances understanding of the factors that influence the allocation of courses to lecturers.

5. Paper 2: Torres, Villegas and Gavina (2021)

5.1 Real-life problem and context

Every semester, the department chair has to assign faculty members to courses. The authors state that this is a tedious process due to the many possible faculty/course combinations, institutional policies that have to be taken into account,

and staffing issues such as people retiring, resigning, or new faculty members being hired. This study was conducted on the faculty/course assignments of a university department.

5.2 Why OR techniques were suitable

Because employee satisfaction increasingly influences scheduling decisions, the authors wanted a model that treats faculty preferences as a first-class input rather than an afterthought. A binary integer programming model can combine preference scores with hard institutional rules in a single optimisation, which a manual process cannot do reliably.

5.3 Key findings

- A binary integer programming model was built that incorporates the **preferences of faculty members**.
- The model also includes constraints that allow faculty to carry out **duties other than instruction**, such as research or administration, so that teaching load is not treated in isolation.
- Applied to a real department, the model produced assignments that respected both institutional policies and individual preferences.

6. Mathematical Modelling

Both papers use a binary integer programming structure. The general formulation below is the standard form of the faculty-course assignment model that both build on. Each paper adds its own refinements (listed in Section 6.4), so the exact equations and parameter values should be checked against the full texts when finalising the report.

6.1 Sets, parameters and decision variables

Symbol	Meaning
$i = 1, \dots, m$	Index of faculty members (lecturers)
$j = 1, \dots, n$	Index of courses (or course sections)
$p(i,j)$	Preference or suitability score of lecturer i for course j (higher is better)
$h(j)$	Teaching hours (or credit load) required by course j
$H(i)$	Maximum workload (hours) that lecturer i can take
$N(i)$	Maximum number of courses lecturer i can teach
$x(i,j)$	Decision variable: equals 1 if lecturer i is assigned to course j , and 0 otherwise

6.2 Objective function

Maximise total preference (satisfaction) of the allocation:

Maximise $Z = \text{sum over all } i \text{ and } j \text{ of } p(i,j) * x(i,j)$

Depending on the study, the objective can alternatively minimise the difference between the heaviest and lightest workloads, so as to promote fairness. Paper 1 emphasises balance, and Paper 2 emphasises preferences.

6.3 Constraints

1. **Every course must be taught:** for each course j , the sum over i of $x(i,j) = 1$.
2. **Workload cap:** for each lecturer i , the sum over j of $h(j) * x(i,j) \leq H(i)$.
3. **Course-count cap:** for each lecturer i , the sum over j of $x(i,j) \leq N(i)$.

4. **Binary restriction:** $x(i,j) = 0$ or 1 for all i and j .
5. **Eligibility:** $x(i,j) = 0$ whenever lecturer i is not qualified for course j .
6. **No timetable clash:** a lecturer cannot be assigned two courses that meet at the same time.

Assumptions: the list of courses, their hours and the timetable are known and fixed before allocation; each course or section has one teacher; lecturer capacities and preference scores are known and accurate; all hours of teaching are treated as comparable units of work.

6.4 What each paper adds to the basic model

Feature	Paper 1 (Mhalapitsa et al.)	Paper 2 (Torres et al.)
Workload factors	Class capacity, contact hours, credits, courses per lecturer	Teaching load within institutional policy limits
Personal factors	Not the main focus	Faculty preferences built into the objective
Other duties	Not the main focus	Constraints for non-teaching duties
Fairness	Core aim: shift load from over-utilised to under-utilised lecturers	Satisfaction through preferences

6.5 Small illustrative example

To illustrate the model, suppose there are 3 lecturers and 3 courses, each lecturer teaches exactly one course. The preference score is shown below (note that these numbers are made up for illustration purposes; they are not derived from either paper). Preference score ranges from 1 to 10.

	Course 1	Course 2	Course 3
Lecturer 1	9	6	5
Lecturer 2	7	8	4
Lecturer 3	6	7	9

There are $3! = 6$ possible one-to-one assignments. Their total preference scores are 26, 22, 20, 19, 19 and 16. The best is Lecturer 1 to Course 1, Lecturer 2 to Course 2 and Lecturer 3 to Course 3, with $Z = 9 + 8 + 9 = 26$. The same result is found by the Hungarian method. With real departments, the number of combinations explodes (20 lecturers and 20 courses gives about 2.4×10^{18} possibilities), which is why a solver is needed.

7. Solution Methodology

Technique: Because the variables are binary, the models are integer linear programs. Solvers typically use the Branch-and-Bound method to solve integer linear programs: The problem is first solved as a linear program (ignoring the integrality constraints), which provides an upper bound on the optimal solution value. Then, the solver branches on variables that take on fractional values in the linear programming solution and prunes branches that cannot improve upon the best known solution. For assignment problems, when the model is purely a one-to-one assignment, the Hungarian method can be used to find the optimal solution quickly.

Software/tools: The abstracts of the two papers do not specify the solver used, so it would be necessary to check the full papers for details. However, common solvers for this type of optimisation model include LINGO, CPLEX, Gurobi, MATLAB, Python (with PuLP), and Microsoft Excel Solver. The supplementary UiTM Seremban study provides an

example of the model's practical application by solving a teaching load allocation model using Excel Solver. This model was applied to a real-world dataset involving 19 courses, 92 student groups, and 25 lecturers, with constraints of no more than 4 student groups and 16 lecture hours per lecturer.

Analysis and validation:

- *Paper 1* validated its model with the Delphi method, in which expert panellists review the model and results in repeated rounds, giving confidence that the model reflects real practice.
- *Paper 2* tested its model on a real department's assignment and checked that the resulting schedule satisfied both institutional policy and faculty preferences.
- In both cases, the solution is easy to verify, because a human can check every constraint on the final allocation table.

8. Critical Analysis and Summary

8.1 Summary of findings and impact

Both papers demonstrate that integer programming can transform a stressful, subjective and time-consuming activity into a transparent, reproducible and quick process. The practical implications of such transformations include optimised use of the lecturer's time, more equitable distribution of workloads, assignment of courses in accordance with the policies and significantly reduced coordinator's workload. The two studies together provide a comprehensive solution for workload allocation by addressing both the quantitative and qualitative aspects of the problem. Paper 1 offers a more complete definition of the workload and implements an expert-level validation. Paper 2 addresses the issue of preferences and non-teaching duties.

8.2 Strengths of the models

- Guarantee that every hard rule is satisfied, which manual allocation often fails to do.
- Reach an optimal or near-optimal allocation quickly, even for realistic department sizes.
- Flexible: new rules (eligibility, caps, clash avoidance) can be added as extra constraints.
- Objective measures such as total preference or load imbalance make the fairness of the result visible.
- Paper 1's expert validation adds credibility beyond mathematical correctness.

8.3 Limitations and assumptions

- Results are dependent on the quality of the input data, which includes subjective preference scores and workload weights.
- Both models are used on a single department, and so results may not be generalizable to another university with different rules.
- Teaching hours are fixed, and the timetable is taken as given, not accounting for possible last-minute clashes or changes.
- Hours or credits may not be an appropriate measure of the work involved in a course, particularly for differing numbers of students and preparation and marking times, or the difference between familiar and new courses.
- A mathematically optimal solution may not always be appealing from a social point of view, and so there is a need for human judgment in such cases.
- An exact integer program may be unduly slow for large institutions

8.4 Possibilities for extension and improvement

- Combine allocation with timetabling and classroom assignment in a single model

- Use multi-objective or goal programming to balance fairness, preferences and cost
- Include research, supervision and administrative load in a unified workload score
- Use heuristics or metaheuristics (genetic algorithms, etc.) for huge instances
- Build a simple user-friendly tool (Excel or web-based) that doesn't require OR training to use each semester
- Re-optimize automatically when staff/course changes necessitate a replanning

9. Conclusion

Faculty workload allocation is one of those problems that makes the real-world application of the operations research model different from what is described in textbooks. Both studies demonstrate that the binary integer programming formulation can effectively resolve the limitations associated with the workload allocation process and consider multiple factors. The study can be improved by ensuring the accuracy of data and expanding the pool of departments that can participate in the study. The operations research application can be used to achieve a more efficient allocation with less time spent on the process compared to the manual approach.

References

- Mhalapitsa, B., et al. (2023). Development and validation of an integer linear programming model for the lecturer-to-course assignment problem. *Journal of Optimisation in Industrial Engineering*, Issue 34, Spring 2023.
- Torres, M. C., Villegas, K. K. S., & Gavina, M. K. A. (2021). Solving faculty-course allocation problem using integer programming model. *Philippine Journal of Science*, 150(4), 679-689.
- Afizan, N. F. I., & Abdul Aziz, N. L. (n.d.). An integer linear programming approach to teaching load allocation. Universiti Teknologi MARA, Negeri Sembilan.
- Mohamad, N. (2006). A zero-one integer programming approach to the assignments of lecturers to courses. *MATEMATIKA*, 22, 31-40.