Aalto-yliopisto
Perustieteiden korkeakoulu

Integer programming formulations for exam hall allocation

(Presentation of results)

Reetta Leinonen

31.10.2025

Advisor: *Topias Terho*

Supervisor: *Philine Schiewe*

Aalto-yliopisto
Perustieteiden korkeakoulu

Background

- During exam weeks, a large number of students must be assigned to seats in exam halls
- Poor allocation increases cost (hall reservations, exam supervisors)
- Manual allocation can be an administrative burden
- Currently, the allocation is done manually → opportunity for optimization

Objectives

- Develop an integer programming (IP) model for exam hall allocation
- Implement the IP formulation using Python and a suitable optimization solver
- Assign each student in a course to some exam hall while minimizing hall reservation and supervision costs
- Test the implementation on anonymized datasets from past exam weeks

Assumptions

- Data and setup
  - Course sizes, hall capacities, and costs are known and fixed
  - Allocation is made for a single exam time slot that has a standard duration (3 + 1 hours)
- Course allocation
  - Each student must be allocated to a hall
  - A row can host only one course
  - Adjacent rows cannot be assigned to the same course
  - Courses can be split across multiple halls and rows if necessary
- Hall configuration
  - Each hall can operate in one of two seating modes (1 or 2 seats between students)

Parameters & decision variables

Sets

- $C = \{1, 2, \ldots, n_C\}$: courses
- $H = \{1, \ldots, n_H\}$: halls
- $R_h = \{1, \ldots, n_{R,h}\}$: rows in hall h
- $M = \{1, 2\}$: seating modes

Parameters

- size_c: participants in course c
- $\text{cap}_{m,h,r}$: capacity of row r in hall h when using seating mode m
- cost_h: reservation cost of hall h
- $\varepsilon_1, \varepsilon_2$: lower/upper bounds for the ε-constrained method

Decision variables

- $x_{m,c,h,r} \in \{0, 1\}$: row assignment
- $y_{m,h} \in \{0, 1\}$: hall usage by mode
- $z_{c,h} \in \{0, 1\}$: course presence in hall

Constraints

The capacities of the rows assigned to course add up to at least the size of the course

$$\sum_{h\in H}\sum_{r\in R_h}\sum_{m\in M} \mathrm{cap}_{m,h,r} x_{m,c,h,r} \geq \mathrm{size}_c \qquad \forall c \in C$$

A row can have at most one course, and using a row in a hall activates the hall

$$\sum_{c\in C} x_{m,c,h,r} \leq y_{m,h} \qquad \forall h \in H, r \in R_h, m \in M$$

A hall is used by the course if at least one row in the hall is allocated to it

$$x_{m,c,h,r} \leq z_{c,h} \qquad \forall c \in C, h \in H, r \in R_h, m \in M$$

A hall is used by the course only if at least one row is allocated to the course

$$\sum_{m\in M}\sum_{r\in R_h} x_{m,c,h,r} \geq z_{c,h} \qquad \forall c \in C, h \in H$$

Same course cannot have two adjacent rows

$$x_{m,c,h,r} + x_{m,c,h,r+1} \leq 1 \qquad \forall c \in C, h \in H, m \in M, r < n_{R,h} - 1$$

A hall can have only one seating mode

$$\sum_{m\in M} y_{m,h} \leq 1 \qquad \forall h \in H$$

Objective function

$$\min \quad \sum_{h \in H} \sum_{m \in M} \mathrm{cost}_h \cdot y_{m,h}$$

Data

- Anonymized student registeration data from mathematics exams (periods 1 & 2, 2024) in Excel format
- Hall capacities, layout data (seats in each row) and pricing in Excel format

					Period 1					
Date	14.10.2024	14.10.2024	15.10.2024	16.10.2024	17.10.2024					**Total**
Time	9:00-12:00	13:00-16:00	16:30-19:30	16:30-19:30	9:00-12:00					
Total registrations	322	609	633	1139	621					**3324**
Unique course codes	5	2	2	5	3					**17**
					Period 2					
Date	02.12.2024	03.12.2024	03.12.2024	05.12.2024	09.12.2024	10.12.2024	11.12.2024	12.12.2024	12.12.2024	**Total**
Time	9:00-12:00	9:00-12:00	16:30-19:30	9:00-12:00	9:00-12:00	9:00-12:00	9:00-12:00	9:00-12:00	16:30-19:30	
Total registrations	984	100	1180	938	292	208	288	406	149	**4545**
Unique course codes	7	3	5	7	7	2	2	5	2	**40**

Implementation & tools

- Python and PyOptInterface package for writing the model
- Gurobi as the underlying solver
- Pandas for reading and preprocessing Excel files

Example allocation with the optimization model (16.10.2024 16:30-19:30)

Optimized hall reservation costs: €4458.0

Course A0101 (participants 108):

→ B, rows: 14
→ C, rows: 2 8
→ Jetti (A208d), rows: 6 8 10
→ T1, rows: 10
→ U2, rows: 18 20

Course A0103 (participants 588):

→ A (Aalto), rows: 2 4 6 8 10 12 14 16
→ B, rows: 1 3 5 7 9 11 13
→ C, rows: 1 3 5 7 9 11 13
→ Jetti (A208d), rows: 1 3 5 7 9 11
→ T1, rows: 1 3 5 7 9 11
→ U2, rows: 1 3 5 7 9 11 13 15 17 19

Course A0109 (participants 121):

→ A (Aalto), rows: 3 13
→ B, rows: 12
→ C, rows: 4 12
→ Jetti (A208d), rows: 2 4
→ T1, rows: 8
→ U2, rows: 16

Course A0111 (participants 263):

→ A (Aalto), rows: 1 7 9 11 15
→ B, rows: 4 6 8
→ C, rows: 14
→ T1, rows: 2 4 6
→ U2, rows: 2 4 6 8 10 12 14

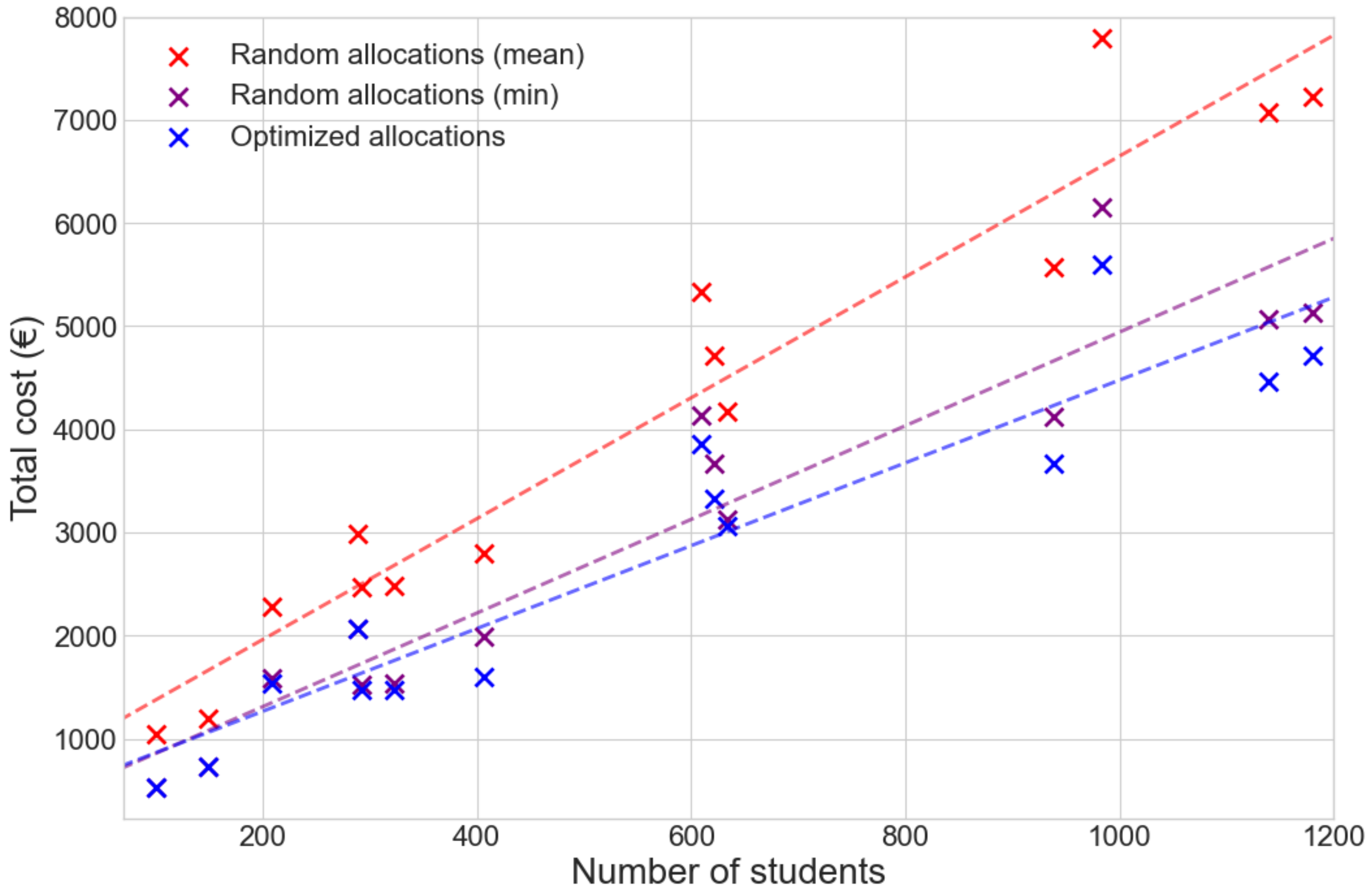
Course E1651 (participants 59):

→ A (Aalto), rows: 5
→ B, rows: 2 10
→ C, rows: 6 10

Comparison to a random allocation (1/2)

- Implemented a random allocation algorithm that mimics the current manual process
- Courses and halls are randomly shuffled at the start of each run
- Courses processed sequentially in the shuffled order
- For each course:
  - Students are assigned row by row until all are seated
  - Only even or odd rows are filled at a time to simulate spacing
  - When a hall is full, the algorithm moves to the next random hall
- Tracks progress for even/odd rows separately
  - The next course uses the row type (even or odd) that has fewer filled rows so far
- The simulation is repeated 1000 times with different random seeds, minimum and average cost were used for the comparison on the next slide

Comparison to a random allocation (2/2)



Multi-objective model (1/3)

- Objectives:
  - Minimize total cost
  - Maximize spacing between students
  - Minimize course splits across halls
- Approach: epsilon-constrained method
  - Optimizes one primary objective while treating others as constraints with specified limits (ε-values)
  - Captures also non-convex Pareto fronts unlike weighted-sum methods
  - Avoids the sensitivity inherent in the weighted-sum method

Multi-objective model (2/3)

- Two additional constraints:

Lower bound for students seated with the sparse seating

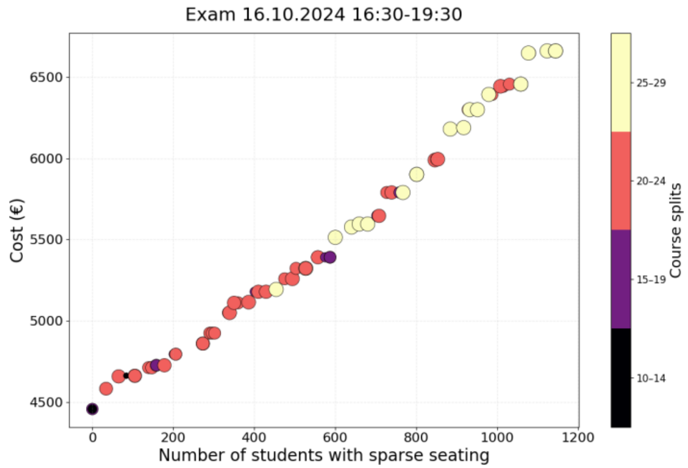
$$\sum_{c \in C} \sum_{h \in H} \sum_{r \in R_h} \text{cap}_{2,h,r} \, x_{2,c,h,r} \geq \varepsilon_1$$

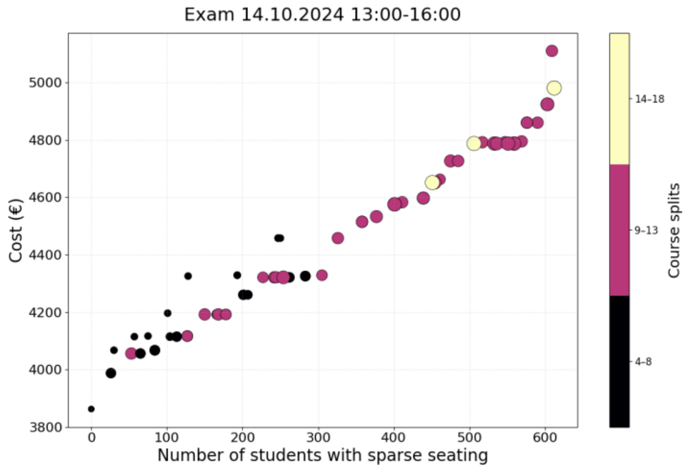
Upper bound for the total course splits

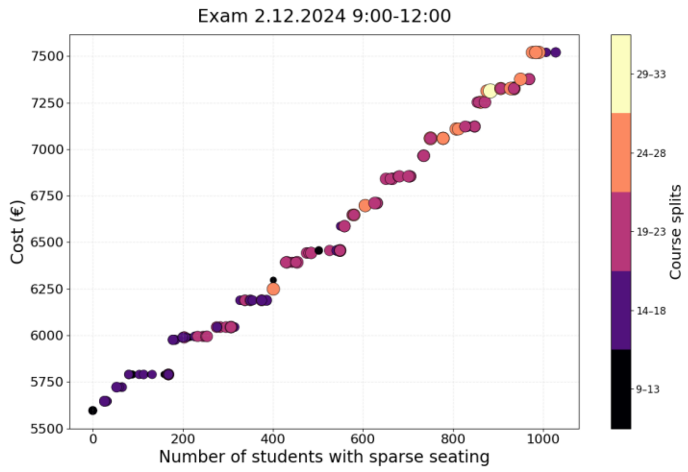
$$\sum_{c \in C} \Big(\sum_{h \in H} z_{c,h} - 1 \Big) \leq \varepsilon_2$$

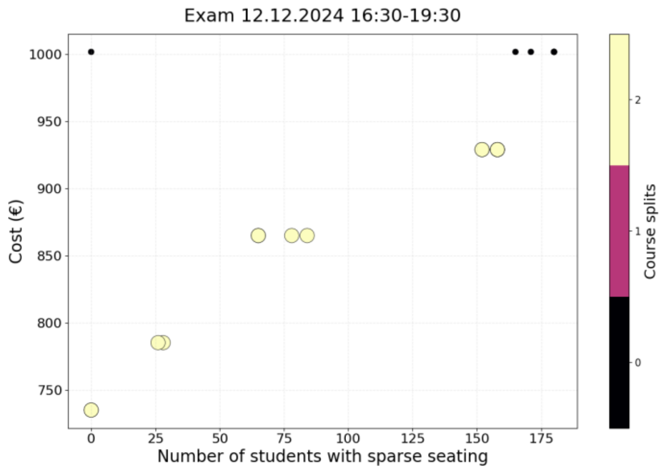
- Experimented with a grid of values for epsilons:
  - Number of students with sparser seating:
    - 0, 25, 50, … , total number of students
  - Total number of course splits:
    - 0, 5, 10, …

Multi-objective model (3/3)









Aalto-yliopisto
Perustieteiden
korkeakoulu

Conclusions

- The proposed optimization model successfully generates feasible exam hall allocation plans
- Compared to a random allocation, the optimized allocation reduces total cost 35% when compared to the average and 6% when compared to the minimum
- The epsilon-constrained approach manages multiple objectives, such as cost, spacing, and minimizing splits
- Results indicate that increasing student spacing leads to a proportional rise in total allocation cost
- The model can serve as a decision-support tool for exam hall allocation and, with further development, could automate the allocation process

References

- Conforti, M., Cornuéjols, G., & Zambelli, G. (2014). *Integer Programming*. Cham: Springer International Publishing.
- Mesquita-Cunha, M., Figueira, J. R., & Barbosa-Póvoa, A. P. (2023). New ϵ− constraint methods for multi-objective integer linear programming: A Pareto front representation approach. *European Journal of Operational Research*, *306*(1), 286-307.