

LEVEL 4 ASSESSMENT



Module Name: *Analysing & Solving Problems*



Module code: *ACCA4019*



Student name: *Thomas Pereira*



Student ID: *2442400*



Group: *1*



Lecturer's Name: *Fahad Ahmed*



Submission: 22 April 2026

Contents

1. Problem Deconstruction	3
1.1 Overview	3
1.2 Key Variables	3
1.3 MoSCoW Analysis.....	4
1.4 Constraints.....	4
2. Proposed Solution and Explanation	5
2.1 Flowchart Overview.....	5
2.2 Step-by-Step Explanation.....	6
2.3 Key Decision Points	7
2.4 Assumptions.....	7
3. Reflection and Justification	7
3.1 Problem-Solving Techniques Used.....	7
3.2 Limitations and Assumptions	7
4. References	9

Introduction

This report analyses the classroom allocation problem faced by a local college and proposes a structured, logical solution to optimise room usage and timetabling efficiency.

1. Problem Deconstruction

1.1 Overview

The problem that is currently facing the college can be categorized under a resource management problem whereby classrooms have not been allocated appropriately for the required classes. The effects of such a scenario are many whereby there are some classrooms that are overcrowded, while other classrooms are underutilized. For one to develop a realistic solution to this issue, it is crucial to dissect the problem and define the factors involved in addition to conducting a requirements analysis [1].

Systems thinking involves analyzing problems from a systems approach, whereby one treats the entire problem as a whole and analyzes all the factors involved in the situation [2]. The problem with classroom allocation at the college does not emanate from a single point of failure, but it comes from several aspects working together.

1.2 Key Variables

These are some of the main variables in our problem, which should be considered when coming up with the solution:

- Room capacity – the total number of people that can be accommodated in a classroom
- Class enrollment – the number of people attending a particular lesson
- Time slots – specific time allocated for each lesson during the college schedule
- Location of rooms – physical location of rooms in relation to each other
- Lessons requirements – whether the class needs a specialized room (like a computer room or art room)
- Time slots and room availability – time constraints in terms of both staff and student availability

1.3 MoSCoW Analysis

Moscow technique for requirement prioritization in systems analysis and project management divides the requirements into four categories: must-have, should-have, could-have, and won't have [3]. When the same technique is applied to the problem of college allocation, we get the following classification:

Priority Category	Requirement
Must Have	Every class is assigned a room with sufficient capacity to accommodate all enrolled students
Must Have	No two classes occupy the same room simultaneously
Must Have	The timetable is fully conflict-free across all time slots
Should Have	Back-to-back classes for the same student group are allocated to rooms in close proximity
Should Have	Specialist rooms (e.g. IT labs, science rooms) are reserved for classes that require them
Could Have	Rooms in less-used building areas are consolidated to reduce energy consumption
Won't Have	Real-time dynamic reallocation of rooms during the college day (out of scope)

This analysis makes clear that conflict prevention and capacity matching are non-negotiable requirements, while movement minimisation is a secondary but important objective.

1.4 Constraints

The most important constraints to consider are the fixed quantity of rooms that are available, the fixed duration of time periods in each day at the college, and the inability to use specialist rooms for other purposes. The availability of teachers as well as existing timetables under contract restrict how freely lessons can be rearranged.

The problem has been broken down into its component parts in order to clearly identify that the most critical issue is that of conflict-free timetabling.

2. Proposed Solution and Explanation

2.1 Flowchart Overview

The diagram that follows shows the logic used to assign classrooms effectively. This diagram has been developed using the technique of sequential decision making, starting from the inputting of data up to the final production of the timetable output.

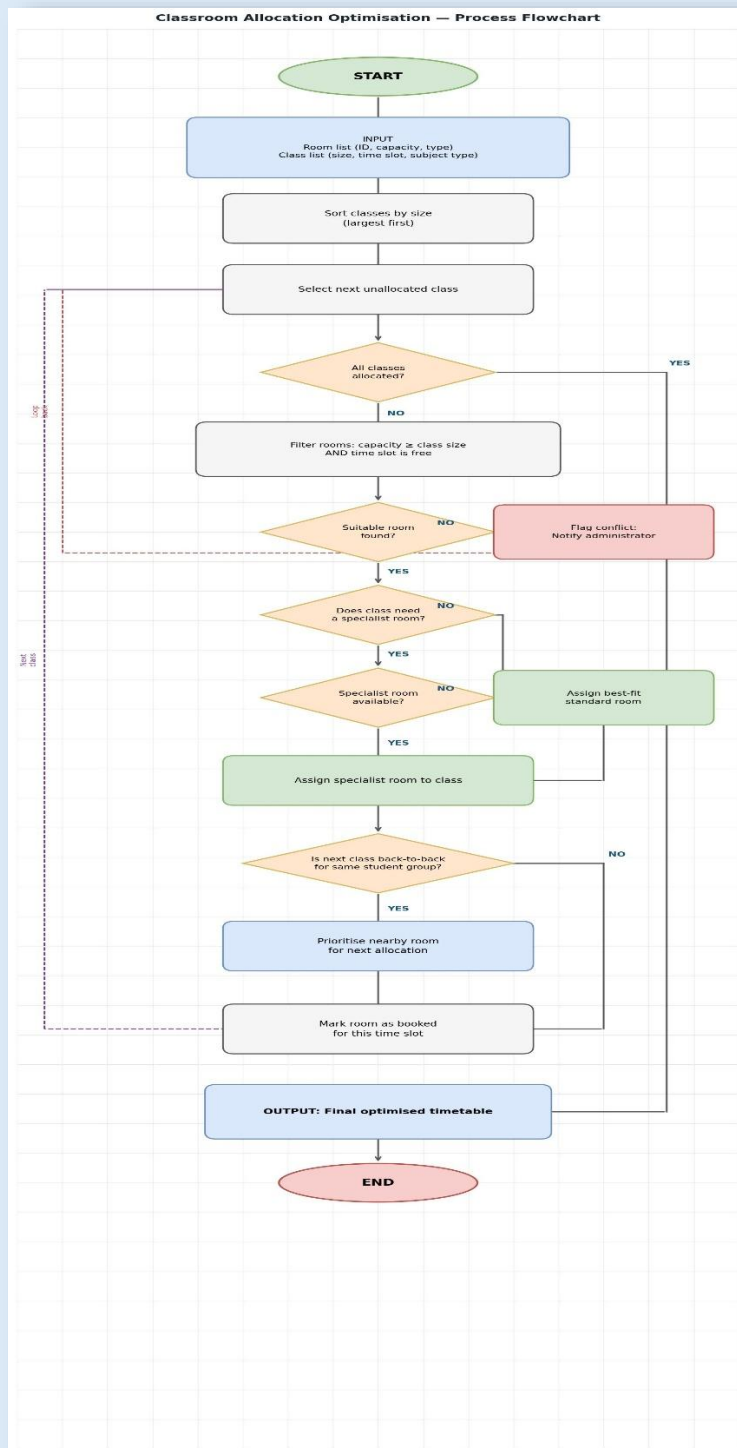


Fig 1: Classroom Allocation Optimisation Process Flowchart

2.2 Step-by-Step Explanation

Step 1 — Data Input: This process starts by gathering all the inputs needed: a comprehensive list of all the rooms that have their respective capacity and type, whether standard or specialty room, and a comprehensive list of all the classes, which includes the size of each class, the time slot, and the subject matter type.

Step 2 — Sort by Class Size: Classes are ordered from largest to smallest prior to scheduling. This guarantees that the largest classes, which have fewer potential room selections available, are scheduled first. This is a popular scheduling technique for optimising class scheduling, commonly known as the “largest-first” approach, which minimises the chance of scheduling conflicts for the larger classes [4].

Step 3 — Filter Available Rooms: In regard to every particular class, the process involves sorting out the available rooms based on two parameters; firstly, that the room capacity is not less than the number of students enrolled in the particular class and secondly, the room is not already booked for that period.

Step 4 — Specialist Room Check: At this stage, the system will determine if the class requires a specialist classroom. If the answer is affirmative, the system will then check if there is any suitable specialist classroom. In case there is, it will be allocated to the class; otherwise, the system will revert to allocating the best suitable standard classroom.

Step 5 — Conflict Handling: However, if an appropriate classroom is not available, a conflict will be detected, and the system will inform the administrator. The algorithm will continue to the next class instead of stopping, which allows more time for the maximum number of allocations possible.

Step 6 — Movement Optimisation: Once a classroom assignment is made temporarily, a second check follows to see if the same group of students is scheduled for another consecutive class period right after that. In such a case, the process gives preference to classrooms located near each other, thereby reducing the time taken by students to travel to another location.

Step 7 — Mark as Booked and Loop: When the room has been assigned to a particular class, it will be noted as occupied in the system during the time period. This process will repeat itself for the remaining classes.

Step 8 — Output: The final output is a complete, optimised timetable showing every class assigned to a specific room and time slot, free of conflicts and matched appropriately to room capacity.

2.3 Key Decision Points

There are four essential decisions made in the flowchart. These include determining whether all the classes have been allocated as part of the loop control; checking whether a room suitable for allocating the particular class exists; determining whether the class requires a specialized room and whether such rooms are available; and finally, whether or not movement minimization logic needs to be applied in the case of classes that happen consecutively. These four decisions enable the solution to dynamically consider the particularities of every class, instead of making the same decision regardless of the situation at hand. They form the foundation of the logic that underpins the entire allocation process. Without them, the solution would be incapable of distinguishing conflicting cases, identifying the need for specialized rooms, or dealing with the need to minimize movement.

2.4 Assumptions

There are several premises on which the proposed solution is based. The first premise is that the data entered into the system is correct and complete before any computations take place; otherwise, the results of the computation will be erroneous. The second assumption is that the time slots are fixed and not adjustable. The third assumption is that the building layout is known and that the proximity of rooms is identifiable. The last assumption is that there is an administrative officer to handle conflict situations.

3. Reflection and Justification

3.1 Problem-Solving Techniques Used

In the context of the present case, decomposition was the main problem-solving approach employed by decomposing a big issue into smaller parts which can be solved one after another [1]. Instead of viewing the whole problem of classroom scheduling as an extremely challenging task, it was broken down into its key parts – class size matching, conflict avoidance, and minimizing movements, which were considered separately in the algorithm.

The use of abstract ideas proved effective when developing the solution since it did not imply specifying a particular institution but rather focused on designing a universal algorithm which works properly irrespective of the amount of classrooms, lessons, and periods involved in the scheduling. Thus, the solution may be applied to any institution [2]. This approach is consistent with soft systems methodology, which emphasises designing flexible, adaptable solutions for complex organisational problems [5].

Prioritization of objectives based on the MoSCoW method was essential in making design decisions to ensure that the flowchart solves most critical issues first, without focusing on secondary problems or neglecting them [3].

3.2 Limitations and Assumptions

There are several shortcomings of the solution. First of all, the algorithm works with the use of predefined input data, and it cannot take into account unexpected factors like unavailability of the room due to some maintenance or increased enrollment of students for the classes. Another limitation is that there is no computation of the time necessary to get from one classroom to another because it considers only the case of back-to-back classes and their location on the campus [4].

Also, the solution works under the assumption that the conflicts will be sorted out quickly by an administrator. In real life, this process takes time and there is always room for human mistakes. It would be better to develop a solution which suggests rescheduling automatically.

Although the solution has its shortcomings, it offers a good logical basis for developing a more efficient system for classroom allocation. The approach of breaking the task down into parts, modeling it using abstractions and prioritizing requirements helps in solving important problems efficiently. [6].

Conclusion

This solution is highly effective in dealing with classroom allocation using a systematic breakdown of the problem, MoSCoW priority setting, and flowcharts, providing a reliable foundation for conflict-free timetable.

References

- [1] S. Baase and A. V. Gelder, "Computer algorithms : introduction to design and analysis," Addison-Wesley Longman, 2000. [Online]. Available: <https://search.worldcat.org/title/40813345>.
- [2] D. H. Meadows, "Thinking in systems : a primer," Chelsea Green Publishing, 2008. [Online]. Available: <https://search.worldcat.org/title/225871309>.
- [3] D. Clegg and R. Barker, "CASE method fast-track," Addison-Wesley, Reading, 1994. [Online]. Available: <https://search.worldcat.org/title/59952336>.
- [4] M. W. C. a. G. Laporte, "Recent developments in practical examination timetabling," Springer Nature, 1996. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-61794-9_49.
- [5] P. Checkland, Systems Thinking, Systems Practice, Chichester: Wiley, 1981.
- [6] E. Burke, K. Jackson, J. H. Kingston and R. Weare, "Automated university timetabling: The state of the art," *The Computer Journal*, vol. 40, no. 9, pp. 565-571, 1997.