FACULTY OF ECONOMICS AND
BUSINESS ADMINISTRATION

Robust university timetable recovery: Evaluating the impact of recovery flexibility under disruption

Leonard Vanryckeghem
Student number: 02004749

Supervisors: Prof. Broos Maenhout, Samuel Bakker
Counsellor: Prof. Beatriz Oliveira (University of Porto)

Master's dissertation submitted in order to obtain the degree of Master of Science in Business Engineering (Double Degree) - Operations Management

Academic year 2024-2025

GHENT
UNIVERSITY

Confidentiality agreement

”The authors and supervisors give permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In the case of any other use, the limitations of the copyright have to be respected, in particular with regard to the obligation to state expressly the source when quoting results from this master dissertation.”

I declare that the research was conducted in accordance with the rules governing scientific and academic integrity. I have read, and acted in accordance with, the Code of Ethics of the Faculty.

Leonard Vanryckeghem, June 2025

Abstract

English version

University course timetables are often disrupted by events such as room unavailability or curriculum changes, requiring recovery actions to restore feasibility and quality before the schedule is put into operation. While most research focuses on generating robust initial schedules, the recovery process has received comparatively less attention, especially in realistic settings where planners cannot freely adjust both time and space assignments.

This thesis investigates the effectiveness of three practically motivated recovery strategies for disrupted university course timetables: adjusting only room assignments while keeping time slots fixed (room-only recovery), adjusting only time slots while keeping rooms fixed (period-only recovery) and restricting recovery to a limited subset of room-period combinations (restricted solution space). These strategies reflect realistic scheduling conditions in which full flexibility is often infeasible and localised changes are preferred. Building on the ITC-2007 curriculum-based timetabling model, a bi-objective optimisation framework is developed and used to evaluate the trade-off between timetable quality and the number of reassignments compared to the original schedule under four disruption types affecting room, time and course structure. In addition, a logistic regression model is used to predict whether recovery is likely to succeed based on solution space availability and schedule load factor.

The results highlight how both the type of disruption and the available recovery flexibility influence rescheduling outcomes, providing guidance on which forms of limited recovery are most effective in different scenarios. These insights can support timetable planners in managing real-world disruptions more effectively.

Portuguese version

Horários de cursos universitários são frequentemente afetados por perturbações, como indisponibilidade de salas ou alterações curriculares, exigindo ações de recuperação para restaurar a viabilidade e a qualidade antes que o horário entre em operação. Embora a maioria das pesquisas se concentre na geração de horários iniciais robustos, o processo de recuperação tem recebido comparativamente menos atenção, especialmente em contextos realistas onde os planejadores não podem ajustar livremente tanto os horários quanto as alocações de salas.

Esta dissertação investiga a eficácia de três estratégias de recuperação motivadas pela prática para horários universitários perturbados: ajustar apenas as alocações de salas mantendo os horários fixos (recuperação apenas de sala), ajustar apenas os horários mantendo as salas fixas (recuperação apenas de período) e restringir a recuperação a um subconjunto limitado de combinações de sala-período (espaço de solução restrito). Essas estratégias refletem condições realistas de agendamento, nas quais a flexibilidade total muitas vezes é inviável e mudanças localizadas são preferidas. Com base no modelo de agendamento de cursos do ITC-2007, é desenvolvido um modelo de otimização biobjetivo para avaliar o compromisso entre a qualidade do horário e o número de reatribuições em comparação com o horário original, considerando quatro tipos de perturbação que afetam a sala, o tempo e a estrutura dos cursos. Além disso, é utilizado um modelo de regressão logística para prever a probabilidade de sucesso da recuperação com base na disponibilidade de espaço de solução e no fator de carga do calendário.

Os resultados destacam como tanto o tipo de perturbação quanto a flexibilidade disponível para recuperação influenciam os resultados do reagendamento, fornecendo orientações sobre quais formas de recuperação limitada são mais eficazes em diferentes cenários. Esses insights podem apoiar os planejadores de horários na gestão mais eficaz de perturbações no mundo real.

Acknowledgements

This thesis is the final work of my Master studies in *Business Engineering (Double Degree) – Operations Management* at *Ghent University*. It marks the end of an intense but fulfilling academic journey.

I chose this topic because of my interest in scheduling problems and the practical relevance of timetabling. University scheduling, in particular, felt tangible and directly connected to the academic environment I've experienced throughout my studies.

This work would not have been possible without the help and support of several individuals, whom I would like to acknowledge here.

First and foremost, I would like to thank my promotor, Broos Maenhout, for his continuous guidance and constructive feedback throughout this process. His insights and critical questions helped shape the direction and quality of this thesis. I am also very grateful to my co-promotor, Samuel Bakker, for his support throughout the process. His clear explanations, practical advice and technical expertise were of great value to this thesis. I would also like to thank Beatriz Oliveira, my promotor at the University of Porto, for her warm welcome during my stay in Portugal and for her valuable reflections and critical insights..

Finally, I want to express my heartfelt thanks to my parents, my girlfriend Saar and my friends. Their ongoing support, encouragement and patience have meant a great deal to me and helped carry me through the final months of this dissertation.

Leonard Vanryckeghem, June 2025

Contents

List of Figures

List of Tables

Chapter 1

Introduction

University course timetables play a critical role in organising when and where lectures take place. Once finalised and published, these schedules serve as the foundation for semester planning. However, disruptions are an unavoidable part of real-world operations. A lecturer may become unavailable, or a room might be reassigned for another purpose. Even if such events are infrequent, they can render parts of the timetable infeasible and require intervention.

Rather than rebuilding the entire timetable from scratch, universities typically prefer to repair the existing schedule with as few changes as possible. This gives rise to the *minimum perturbation problem*, where the goal is to restore feasibility while minimising disruption to students and staff.

Yet feasibility alone is not sufficient. The quality of a timetable, measured through soft constraints such as room stability, lecture compactness, or even distribution of lectures across the week, remains essential for usability. A minimally changed solution may still lead to a dramatic deterioration in quality. Therefore, effective timetable recovery must strike a balance between minimising changes and maintaining or restoring soft constraint satisfaction.

This thesis builds upon the recovery-orientated framework which models the trade-off between recovery effort and solution quality using a bi-objective optimisation approach (Lindahl et al., 2019). While their work focused on full flexibility in rescheduling, this study extends the analysis by systematically evaluating how recovery performance varies under different types of disruption and constrained flexibility. Methodologically, the thesis follows a structured experimental design centred on comparative model adaptation. We begin by constructing a baseline recovery scenario in which disruptions are resolved with complete rescheduling flexibility, allowing unrestricted changes to both room and period assignments. From this baseline, we systematically introduce constraints that limit the recovery process. This stepwise approach enables a controlled comparison of how different recovery limitations affect performance by providing a consistent framework for analysing robustness across disruption types and flexibility conditions, ultimately offering practical insights into when and how timetable repair is most effective.

This thesis is organised as follows: In Chapter 2, the relevant literature is reviewed. In Chapter 3, the problem context and model are introduced. In Chapter 4, the methodology is described. In Chapter 5, the computational experiments and results are presented. In Chapter 6, the main conclusions are drawn and directions for future research are discussed.

Chapter 2

Literature Review

University course timetabling is a well-studied combinatorial optimisation problem with numerous practical applications. This chapter reviews the academic literature relevant to the development of effective scheduling and rescheduling methods. It begins by addressing established approaches to constructing high-quality university timetables and then examines re-optimisation techniques developed to restore feasibility and quality after disruptions.

2.1 University course timetabling and the ITC-2007 benchmark

University course timetabling involves assigning a set of lectures to specific time periods and rooms while satisfying institutional constraints. These typically include hard constraints (e.g., avoiding conflicts between lectures in the same curriculum, ensuring that no teacher or room is double-booked) and soft constraints (e.g., reducing room overcapacity and minimising isolated lectures). From a computational standpoint, university course timetabling is a well-known NP-hard problem, as it can be reduced to a graph colouring problem (Lindahl et al., 2019). This implies that, while verifying the feasibility of a proposed timetable is computationally tractable, finding an optimal or even high-quality feasible solution becomes increasingly difficult as the problem size grows. The complexity stems from the combinatorial explosion in the number of possible course–room–period assignments under various institutional constraints.

This thesis adopts the *Curriculum-Based Course Timetabling* (CB-CTT) formulation in the context of the Second International Timetabling Competition (ITC-2007) (Di Gaspero et al., 2007). In CB-CTT, each curriculum represents a fixed group of students attending a common set of courses. Feasible timetables must satisfy all hard constraints and aim to minimise violations of soft constraints, including room capacity (minimising student overflow in assigned rooms), room stability (using the same room for all lectures of a course), curriculum compactness (avoiding isolated lectures) and minimum working days (spreading a course across the week). The ITC-2007 benchmark comprises 21 such instances derived from real-world data provided by the University of Udine. These instances vary in size and structure and have become a de facto standard in the timetabling literature, enabling reproducibility and cross-study comparison. All recovery experiments in this thesis are conducted on these benchmark instances to ensure consistency with prior research. These instances and their characteristics will be further discussed in Chapter 4.

The curriculum-based course timetabling problem has been tackled using a wide variety of optimisation techniques. High-performing metaheuristics such as threshold accepting (Geiger, 2010) and hybrid genetic-local search algorithms (Badoni and Gupta, 2015) have demonstrated their effectiveness in generating high-quality solutions for large-scale instances. In addition, matheuristics such as the fix-and-optimise strategy by Lindahl et al. (2018b) and decomposition-based MIP formulations by Lach and Lübbecke (2012) have enabled exact or near-exact solution approaches, particularly valuable for benchmarking and lower bound analysis (Hao and Benlic, 2011). Among these, the decomposition-based method by Lach and Lübbecke (2012) stands out for its sequential treatment of period and room assignment, which significantly reduces model complexity and supports modular solver design. While their work does not explicitly address recovery, the structural separation of time and room dimensions provides useful motivation for the restricted recovery strategies introduced in this thesis. Specifically, the room-only and period-only recovery mechanisms explored here align with this decomposition by isolating rescheduling flexibility to a single dimension.

University course timetabling can be viewed as a multi-level decision-making process, comprising strategic, tactical and operational planning layers (Lindahl et al., 2018a). At the strategic level, high-level decisions are made regarding the academic calendar, curriculum structure and long-term resource allocation. Tactical timetabling focuses on assigning teachers to courses and defining base room and time allocations. Finally, operational timetabling refers to the creation and day-to-day adaptation of detailed timetables in response to real-world disturbances such as last-minute room unavailability, unexpected enrolment changes, or teacher absences. This thesis bridges the gap between the tactical and operational layer, where disruptions may arise after a feasible timetable has already been generated, but before the schedule goes into operation. In this context, the objective is not to construct a new timetable from scratch, but rather to recover feasibility and quality while making as few changes as possible to the original schedule. The recovery framework developed in this thesis explicitly models this challenge, emphasising the practical importance of constrained flexibility and minimal perturbation in real-world scheduling environments.

2.2 Handling disruptions: from perturbation to recovery

While much research in university course timetabling has focused on constructing high-quality initial schedules, real-world timetables are often subject to disruptions that arise after a feasible schedule has been generated. These disruptions may include room unavailability, unexpected enrolment changes, or last-minute scheduling conflicts due to staff absences. They typically occur in a critical, yet often overlooked, phase: after a timetable has been finalised and published, but before it is actually put into use (Lindahl et al., 2019).

Reoptimising the entire timetable in response to such disruptions is generally undesirable. Although a complete reconstruction could yield a high-quality new solution, it often requires widespread changes across the schedule impacting many lecturers, students and administrative constraints. Instead, institutions typically aim to perform *recovery*: restoring feasibility by selectively modifying only the affected parts of the timetable while minimising disruption to the rest.

A common formalisation of this process is the *Minimal Perturbation Problem* (MPP), which seeks a feasible timetable that differs as little as possible from the original. The number of changes, counted as course–period–room reassignments, is minimised to preserve continuity. Müller et al. (2005) were among the first to practically apply this principle to university timetabling, emphasising that institutional acceptance often depends more on preserving structural stability than on fully optimising soft constraints.

However, a strict focus on minimal change can come at the expense of timetable quality: recovered solutions may exhibit decreased room stability, more isolated lectures, or an uneven spread of courses throughout the week, even when all hard constraints are satisfied (Lindahl et al., 2019). This motivates the use of bi-objective models, which jointly minimise perturbation effort and solution quality degradation. A notable example is the recovery framework of Lindahl et al. (2019), who formalise these competing priorities in a curriculum-based course timetabling setting.

Beyond university timetabling, alternative recovery strategies have also been developed in other domains. For example, El Sakkout et al. (1998) propose a hybrid constraint and linear programming approach for recovery in dynamic scheduling in a working schedule while Zivan et al. (2011) use a local search method for recovery in dynamic constraint satisfaction problems. Similarly, Maenhout and Vanhoucke (2011) study minimal-change recovery in the context of nurse rerostering, balancing fairness and preference satisfaction with minimal deviation from the original roster. Within university course timetabling, Phillips et al. (2017) propose an integer programming approach using expanding neighbourhoods to recover feasibility while limiting the number of changes. Together, these works reinforce the broader relevance of limited-change recovery frameworks in practice, in a field where recovery has received comparatively less attention than initial timetable generation.

Chapter 3

Problem description and Formulation

As discussed in Chapter 1, university timetables are often finalised before the semester starts, but remain vulnerable to late-stage disruptions such as lecturer unavailability or room conflicts. These disruptions may invalidate part of the published schedule and require controlled recovery.

Rather than constructing a new timetable from scratch, this thesis focuses on repairing robustly the existing schedule through minimal changes while aiming to preserve overall quality. The remainder of this chapter formalises the quality recovery problem, building on the Curriculum-Based Course Timetabling (CB-CTT) formulation.

3.1 Curriculum-Based Course Timetabling (CB-CTT)

The CB-CTT problem, as formalised in the ITC2007 competition by Di Gaspero et al. (2007), involves assigning a set of lectures to periods and rooms over a fixed period. Each period is a day-timeslot combination. Courses consist of several lectures taught by a single teacher and shared among students. Rooms differ in capacity and can host one lecture per period. Curricula are defined as groups of courses with overlapping student enrolments. The aim is to build a feasible timetable satisfying these constraints while optimising soft criteria like room stability, compactness and lecture spread.

3.1.1 Static problem formulation

The static Curriculum-Based Course Timetabling (CBCTT) problem is formulated as a Mixed-Integer Programming (MIP) model (Model 1). The goal is to assign a fixed number of lectures for each course $c \in C$ to specific periods $p \in P$ and rooms $r \in R$, while satisfying a set of hard and soft constraints. Each lecture must be assigned exactly once to a valid (period, room) combination.

$$
\begin{aligned}
\min \quad f_{\text{qual}} = & \sum_{c\in C, p\in P, r\in R} obj(c,r)\cdot x_{c,p,r} && (3.1)\\
& + \sum_{c\in C, r\in R} 1\cdot y_{c,r} - |C| && (3.2)\\
& + \sum_{c\in C} 5\cdot w_c && (3.3)\\
& + \sum_{cu\in CU, p\in P} 2\cdot v_{cu,p} && (3.4)
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{p\in P, r\in R} x_{c,p,r} = L(c) && \forall c\in C && (3.5)\\
& \sum_{c\in C} x_{c,p,r} \leq 1 && \forall p\in P, r\in R && (3.6)\\
& \sum_{c\in C(t), r\in R} x_{c,p,r} \leq 1 && \forall t\in T, p\in P && (3.7)\\
& \sum_{c\in C(cu), r\in R} x_{c,p,r} \leq 1 && \forall cu\in CU, p\in P && (3.8)\\
& \sum_{p\in P} x_{c,p,r} - |P|\cdot y_{c,r} \leq 0 && \forall c\in C, r\in R && (3.9)\\
& \sum_{p\in d, r\in R} x_{c,p,r} - z_{c,d} \geq 0 && \forall c\in C, d\in D && (3.10)\\
& \sum_{d\in D} z_{c,d} + w_c \geq mnd(c) && \forall c\in C && (3.11)\\
& \sum_{c\in C(cu), r\in R} x_{c,p,r} - r_{cu,p} = 0 && \forall cu\in CU, p\in P && (3.12)\\
& -r_{cu,p-1} + r_{cu,p} - r_{cu,p+1} - v_{cu,p} \leq 0 && \forall cu\in CU, p\in P && (3.13)
\end{aligned}
$$

$$
\begin{aligned}
& x_{c,p,r} \in B && \forall c\in C, p\in P, r\in R && (3.14)\\
& y_{c,r} \in B && \forall c\in C, r\in R && (3.15)\\
& w_c \in Z_+ && \forall c\in C && (3.16)\\
& z_{c,d} \in B && \forall c\in C, d\in D && (3.17)\\
& v_{cu,p} \in B && \forall cu\in CU, p\in P && (3.18)\\
& r_{cu,p} \in B && \forall cu\in CU, p\in P && (3.19)
\end{aligned}
$$

Model 1: Static MIP model for curriculum-based course timetabling.

To model this, the binary decision variable $x_{c,p,r}$ is introduced, defined as:

$$x_{c,p,r} = \begin{cases} 1 & \text{if course } c \in C \text{ is planned at period } p \in P \text{ and in room } r \in R \\ 0 & \text{otherwise} \end{cases}$$

Hard constraints and feasibility requirements

Equation (3.5) ensures that all lectures are scheduled exactly once, which enforces that each course $c \in C$ is assigned to precisely $L(c)$ course-period-room combinations. Constraint (3.6) ensures room occupancy limits: no room $r \in R$ can be used for more than one lecture in the same period $p \in P$, maintaining feasibility in the physical use of space. Teacher conflicts are prevented through constraint (3.7), which ensures that each teacher $t \in T$ is involved in at most one course at any given period. Similarly, curriculum conflicts are avoided using constraint (3.8), making sure that for any curriculum $cu \in CU$, students are not scheduled to attend more than one course simultaneously.

Soft constraints and objective function

The quality of a timetable is determined by the extent to which soft constraints are violated. Four types of soft constraints are considered, each contributing penalty points when violated. The objective function (3.1)–(3.4) minimises the total penalty across these dimensions.

Room Capacity. To discourage over-allocation of students to rooms with insufficient capacity, the model assigns a penalty of 1 for each student exceeding a room's maximum capacity. Each room $r \in R$ has an associated capacity cap(r) and each course $c \in C$ has a known demand $s(c)$ reflecting the number of enrolled students. For each course-room assignment, the cost function is defined as obj$(c, r) = \max\{0,\ s(c) - \text{cap}(r)\}$, which yields a penalty equal to the number of students exceeding the room's capacity. This penalty is integrated directly into the objective function (3.1) as a weighted sum over all course-room-period combinations.

Room Stability. This constraint aims to minimise the number of distinct rooms used per course. The binary variable $y_{c,r}$ is activated whenever course c is scheduled in room r (3.9). The objective then penalises each additional room used beyond the first (3.2), encouraging consistent room assignments for individual courses. A penalty of 1 is given for each extra room used.

Minimum Working Days. To encourage temporal spread in course schedules, each course is required to be taught on a minimum number of distinct days. Let D be the set of days and let $p \in d$

denote all periods within day $d \in D$. The binary variable $z_{c,d}$ is introduced to indicate whether course c is scheduled on day d (3.10). The total number of days a course is scheduled is then compared against its required minimum $mnd(c)$. If this threshold is not met, a slack variable w_c accounts for the shortfall (3.11). The objective function penalises each such violation with a weight of 5 (3.3).

Curriculum Compactness. To enhance the daily experience of students, the model discourages isolated lectures (i.e., lectures scheduled without adjacent sessions) from appearing in a curriculum's timetable. The binary variable $r_{cu,p}$ signals whether any course belonging to curriculum cu is scheduled during period p (3.12). The auxiliary variable $v_{cu,p}$ identifies isolated lectures. $v_{cu,p}$ becomes active when a period p contains a lecture while the adjacent periods $p-1$ and $p+1$ remain empty for the same curriculum (3.13). This sliding-window detection mechanism effectively captures isolated periods within the daily schedule. Each isolated lecture is penalised with two objective points (3.4).

The model and definitions presented in this chapter provide the foundation for constructing and evaluating recovery strategies. In the next chapter, we outline how these formulations are embedded within a broader solution methodology, detailing how different recovery mechanisms are implemented and compared across disruption scenarios.

Chapter 4

Solution Methodology

This chapter presents the solution methodology developed to address timetable recovery after disruptions. It outlines the underlying modelling and algorithmic approach and introduces the different recovery strategies that are evaluated in the remainder of this thesis.

4.1 Quality recovery mechanism

We begin by formalising the core components of the recovery model. This includes the perturbation function that captures the number of assignment changes and the integration of this measure into a bi-objective optimisation formulation.

4.1.1 Perturbation function

When a disruption renders the original timetable infeasible, the recovery process involves modifying the initial solution by reassigning lectures to new room-period combinations. The *perturbation function* quantifies this effort by counting the number of such assignment changes relative to the original solution.

Let $x_{cpr}^{\text{init}} \in \{0,1\}$ denote the initial assignment for course c in period p and room r and let $x_{cpr} \in \{0,1\}$ be the recovered assignment. The perturbation function is defined as:

$$f_{\Delta}(x, x^{\text{init}}) = \sum_{c \in \mathcal{C}} \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} \left| x_{cpr} - x_{cpr}^{\text{init}} \right| \tag{4.1}$$

This function measures the total number of changes in course-period-room assignments. Each modification, the removal of an existing assignment or the addition of a new one, counts as a change. This formulation ensures that the recovery process aims to restore feasibility with as little deviation from the original timetable as possible.

4.1.2 Bi-objective model formulation

The full *Quality Recovery Problem* is formulated as a bi-objective Mixed Integer Programming (MIP) model (Model 2) that integrates both solution quality and perturbation minimisation. This formulation was first introduced by Lindahl et al. (2019), who developed a structured framework to evaluate the trade-off between preserving solution quality and minimising deviations from the original timetable.

Building upon the static Curriculum-Based Course Timetabling (CB-CTT) model outlined in Chapter 3 (Model 1), this recovery formulation extends it by incorporating a second objective function $f_{\Delta}(x, x^{\text{init}})$, which measures the effort (i.e., the number of reassignments) and an additional constraint that inserts the disruption.

$$
\begin{aligned}
\min \quad & f_{\text{qual}}(x) && \text{(defined in (3.1)–(3.4))} \\
\min \quad & f_{\Delta}(x, x^{\text{init}}) && \text{(defined in (4.1))} \\
\text{s.t.} \quad & \text{Static constraints from (3.5)–(3.13)} \\
& \text{Disruption-specific constraints (4.2)–(4.5)}
\end{aligned}
$$

Model 2: Bi-objective MIP model for timetable recovery.

This bi-objective setup allows the generation of Pareto-optimal recovery solutions by balancing the degree of change with the restored timetable quality. While Lindahl et al. (2019) introduced the initial formulation, our work extends its application by systematically evaluating how this trade-off behaves under varying recovery mechanisms and flexibility constraints. The constraints (4.2)-(4.5) that capture the disruptions are discussed in section 4.1.3.

Quality recovery algorithm

To solve the quality recovery problem formulated in the bi-objective MIP Model, both the timetable quality f_{qual} and the perturbation effort f_{Δ} must be taken into account simultaneously. These objectives are inherently conflicting: minimising the number of changes limits the flexibility to improve solution quality and vice versa. This results in a bi-objective optimisation problem, where the goal is to explore the trade-off curve between both objectives.

To generate the set of Pareto-optimal solutions, where one objective cannot be improved without worsening the other, we adopt the ϵ-constraint method (Haimes et al., 1971). This approach was used in the context of university timetabling by Lindahl et al. (2019) and forms the basis of our recovery strategy.

The method begins with an initial perturbation bound of $\Delta = 1$, incrementally increasing it in each iteration to progressively explore the Pareto frontier between solution quality and perturbation effort. For each value of Δ, which corresponds to ϵ in the classical ϵ-constraint method, a bi-objective model is solved by minimising the timetable quality f_{qual}, subject to the constraint that the perturbation distance f_{Δ} remains within Δ. This iterative process continues until a stopping criterion is met. The approach follows an ϵ-constraint scheme and is formally described in Algorithm 1.

Algorithm 1: Quality recovery using the ϵ-constraint method.

1. $\Delta \leftarrow 1$ *// Initial perturbation bound*
2. $f_{\text{qual}}(x) \leftarrow \infty$
3. **repeat**
4. Update ϵ-constraint: $f_{\Delta}(x, x^{\text{init}}) \leq \Delta$
5. $S_{\text{QR}}^{\epsilon} \leftarrow \min f_{\text{qual}}(x)$ *// Find Pareto-solution*
6. $\Delta \leftarrow \Delta + 1$
7. **until** Stopping criterion met

4.1.3 Disruption types

The four types of disruptions used by Lindahl et al. (2019) reflect realistic challenges in university timetabling. Each disruption type introduces specific constraints that alter the feasible solution space and trigger the recovery process.

Invalid Assignment. This disruption removes a specific course-room-period assignment $(\hat{c}, \hat{p}, \hat{r})$ from the original timetable. This course needs to be assigned to a different room or to a different timeslot. This only affects a single assignment and is therefore the least invasive in terms of scope. The following constraint is added:

$$x_{\hat{c}\hat{p}\hat{r}} = 0 \tag{4.2}$$

Insert Curriculum. This disruption introduces four new courses $\hat{\mathcal{C}}$, which are placed into a new curriculum. These courses must be scheduled without overlapping existing lectures. The addition of this new curriculum introduces multiple conflict constraints that indirectly reduce scheduling flexibility across the timetable. As a result, this disruption typically impacts multiple assignments.

This is enforced through the constraint:

$$\sum_{c\in\hat{\mathcal{C}},\ r\in R} x_{cpr} \leq 1 \quad \forall p \in P \tag{4.3}$$

Removed Room. This disruption makes a specific room $\hat{r} \in R$ unavailable for an entire day $\hat{d} \in D$. The recovery model prohibits all assignments to this room during any period on the affected day:

$$x_{cpr} = 0 \quad \forall c \in C,\ \forall p \in \hat{d},\ r = \hat{r} \tag{4.4}$$

For ease of reference, this disruption will be referred to as *Removed Room* throughout the remainder of this thesis, although it strictly concerns the unavailability of a room for a certain day. The number of affected assignments depends on the original utilisation of the room and may range from none to several course lectures.

Removed Period. This disruption makes a specific period $\hat{p} \in P$ unavailable for scheduling. Depending on the density of the original timetable, this disruption may simultaneously invalidate multiple assignments across different courses and rooms, making it one of the more severe disruption types in terms of immediate feasibility impact. All course-room assignments at this period are prohibited:

$$x_{cpr} = 0 \quad \forall c \in C,\ r \in R,\ p = \hat{p} \tag{4.5}$$

4.2 Proposed recovery strategies

Each disruption introduces feasibility constraints to the recovery model, forcing the solver to adapt while minimising the number of changes and preserving solution quality. This interplay between disruption type and recovery flexibility forms the core focus of this thesis, where we systematically explore how different recovery mechanisms perform under varied constraint scenarios.

4.2.1 Full recovery flexibility

This baseline strategy allows complete flexibility in reassigning course–period–room combinations during recovery. Specifically, the model is free to change both the period and the room of any course without restriction. It can make any necessary adjustments to recover from the disruption, subject only to the epsilon budget and the bi-objective formulation. This recovery mechanism is identical to the one proposed in Lindahl et al. (2019) and it serves as our reference baseline throughout this thesis against which alternative recovery strategies are evaluated. The underlying formulation is provided in Model 2.

4.2.2 Restricted recovery mechanisms

Real-world recovery from timetable disruptions does not necessarily permit unrestricted changes to both time and room allocations. Operational constraints (e.g., instructor availability, fixed lecture sequences or pre-booked facilities) often make certain types of changes more feasible than others. To reflect this, we examine restricted recovery mechanisms that limit the flexibility of the model to either room reassignments or period shifts. These scenarios simulate common institutional policies, where, for example, time changes are avoided to preserve coordination or where room changes are infeasible due to specialised equipment or fixed capacity allocations.

In addition to these two restricted recovery strategies, we explore a third restricted recovery regime in which only a fraction of the original solution space is available. Instead of allowing full access to all period–room combinations, we deactivate the furthest combinations from the original schedule. These are the ones that involve significant spatial or temporal deviation. This simulates environments where rescheduling flexibility is limited to only the most similar or convenient alternatives. By prioritising the closest options, we assess whether recovery quality can be preserved under localised flexibility and whether such strategies can provide actionable guidelines for planners facing constrained operational settings.

Finally, each of these recovery regimes is evaluated across all disruption types introduced in Section 4.1.3, allowing us to systematically assess how different disruptions respond to varying flexibility constraints. This enables a comprehensive understanding of where robustness is lost and which recovery strategies prove more resilient under pressure.

Room-only recovery

This recovery regime investigates a restrictive setting in which only room assignments may be changed, while period assignments are fixed to their values in the original solution. This constraint reflects realistic institutional scenarios where altering scheduled time slots is infeasible or undesirable (e.g., fixed course sequences, coordination requirements, lecturer availability constraints). In such cases, reallocating a lecture to a different room is often the more practical option, making room-only recovery a relevant mechanism in operational timetabling.

Let $\mathcal{A}^{\text{init}} \subseteq C \times P \times R$ denote the set of original assignments, where each tuple $(c', p', r') \in \mathcal{A}^{\text{init}}$ corresponds to a course c' assigned to period p' and room r' in the original non-disrupted timetable. From $\mathcal{A}^{\text{init}}$, we derive the set of originally used course-period pairs, denoted as $\mathcal{T}^{\text{init}}$, capturing all period assignments originally associated with each course, independent of their room:

$$\mathcal{T}^{\text{init}} = \{(c', p') \mid \exists\, r' \text{ such that } (c', p', r') \in \mathcal{A}^{\text{init}}\}.$$

With these definitions in place, three recovery-specific constraints are added to the baseline model (Model 2) to restrict the allowable assignments and define which variables may change relative to the original solution:

For each original assignment $(c', p', r') \in \mathcal{A}^{\text{init}}$, no change $\delta_{c'p'r'}$ is counted if the original room is retained:

$$\delta_{c'p'r'} = 0 \quad \forall (c', p', r') \in \mathcal{A}^{\text{init}} \tag{4.6}$$

For the same course-period pair $(c', p') \in \mathcal{T}^{\text{init}}$, any reassignment to a different room $r \neq r'$ is explicitly counted as a change:

$$\delta_{c'p'r} = x_{c'p'r} \quad \forall (c', p', r') \in \mathcal{A}^{\text{init}}, \quad \forall r \in R \setminus \{r'\} \tag{4.7}$$

Finally, for all course-period pairs $(c, p) \notin \mathcal{T}^{\text{init}}$, no assignment is permitted in any room:

$$x_{cpr} = 0 \quad \forall (c, p) \notin \mathcal{T}^{\text{init}},\ \forall r \in R \tag{4.8}$$

The resulting bi-objective MIP formulation is given in Model 3.

$\min$	$f_{\text{qual}}(x)$	(defined in (3.1)–(3.4))
$\min$	$f_{\Delta}(x, x^{\text{init}})$	(defined in (4.1))
s.t.	Static constraints from (3.5)–(3.13)	
	Disruption-specific constraints (4.2)–(4.5)	
	Recovery constraints (4.6)–(4.8)	

Model 3: Room-only recovery model: bi-objective MIP formulation with restricted recovery constraints.

Period-only recovery

This recovery regime investigates a restrictive setting in which only period assignments may be changed, while room assignments are fixed to their values in the original solution. This constraint reflects real-world scenarios where spatial flexibility is limited (e.g., courses require room-specific equipment, departments have exclusive room rights, or building logistics prevent reallocation). In such contexts, rescheduling a lecture to a different time slot may be more feasible than relocating it to another room, making period-only recovery a relevant mechanism in practice.

Let $\mathcal{A}^{\text{init}} \subseteq C \times P \times R$ denote the set of original assignments, where each tuple $(c', p', r') \in \mathcal{A}^{\text{init}}$

corresponds to a course c' assigned to period p' and room r' in the original non-disrupted timetable. From $\mathcal{A}^{\text{init}}$, we derive the set of originally used course-room pairs, denoted as $\mathcal{S}^{\text{init}}$, capturing all room assignments originally associated with each course, independent of their period:

$$\mathcal{S}^{\text{init}} = \{(c', r') \mid \exists\, p' \text{ such that } (c', p', r') \in \mathcal{A}^{\text{init}}\}.$$

With these definitions in place, three recovery-specific constraints are added to the baseline model (Model 2) to restrict the allowable assignments and define which variables may change relative to the original solution:

For each original assignment $(c', p', r') \in \mathcal{A}^{\text{init}}$, no change $\delta_{c'p'r'}$ is counted if the original period is retained:

$$\delta_{c'p'r'} = 0 \quad \forall (c', p', r') \in \mathcal{A}^{\text{init}} \tag{4.9}$$

For the same course-room pair $(c', r') \in \mathcal{S}^{\text{init}}$, any reassignment to a different period $p \neq p'$ is explicitly counted as a change:

$$\delta_{c'pr'} = x_{c'pr'} \quad \forall (c', p', r') \in \mathcal{A}^{\text{init}}, \quad \forall p \in P \setminus \{p'\} \tag{4.10}$$

Finally, for all course-room pairs $(c, r) \notin \mathcal{S}^{\text{init}}$, no assignment is permitted in any period:

$$x_{cpr} = 0 \quad \forall (c, r) \notin \mathcal{S}^{\text{init}},\ \forall p \in P \tag{4.11}$$

The resulting bi-objective MIP formulation is given in Model 4.

$$\begin{aligned}
\min \quad & f_{\text{qual}}(x) && \text{(defined in (3.1)–(3.4))} \\
\min \quad & f_{\Delta}(x, x^{\text{init}}) && \text{(defined in (4.1))} \\
\text{s.t.} \quad & \text{Static constraints from (3.5)–(3.13)} \\
& \text{Disruption-specific constraints (4.2)–(4.5)} \\
& \text{Recovery constraints (4.9)–(4.11)}
\end{aligned}$$

Model 4: Period-only recovery model: bi-objective MIP formulation with restricted recovery constraints.

Restricted solution space recovery

This recovery mechanism investigates a scenario in which the timetabling flexibility is constrained by limiting available room-period assignment options. Instead of permitting unrestricted rescheduling across all possible room-period combinations, the model considers only a dynamically selected subset of room-period pairs that are closest to the disruption, based on a specified percentage of the full solution space.

To define the restricted solution space, we introduce $\mathcal{U} \subseteq R \times P$, representing the allowed room-period combinations selected based on their closeness to the disrupted assignment(s). Closeness is quantified using the Manhattan distance:

$$D_{\text{Manhattan}}((r_1, p_1), (r_2, p_2)) = |\text{rank}(r_1) - \text{rank}(r_2)| + |p_1 - p_2|$$

where rank(r) denotes the index of room r in a predefined ordering and p represents the period. The predefined ordering refers to the sequence in which periods and rooms appear in the input files. For periods, this ordering reflects a logical progression over time across days and slots. For rooms, the ordering in the input file does not carry an explicit interpretation but is used as a proxy for proximity. This relies on the implicit assumption that rooms listed close together in the input are also similar or nearby in practice.

When selecting the closest $x\%$ of room–period combinations based on this distance metric, ties may occur between multiple combinations with identical distance scores. In such cases, no specific rule is used to break ties. The selection simply follows the order in which room–period pairs are processed in the code. This tie-breaking procedure is therefore arbitrary and depends on how the data is structured, rather than on optimisation criteria.

While more refined selection rules could theoretically lead to better quality solutions, identifying the optimal subset would require evaluating all possible tie-breaking permutations, which is computationally demanding given the already significant runtime demands of the recovery process. The current approach strikes a practical balance between proximity-based filtering and computational tractability.

Assignments to room-period combinations outside the allowed subset $\mathcal{U}$ are explicitly forbidden:

$$x_{cpr} = 0 \quad \forall c \in C, \quad \forall (r, p) \notin \mathcal{U} \tag{4.12}$$

The set $\mathcal{U}$ is defined dynamically for each disruption type (invalid assignment, curriculum insertion, room removal or period removal) by selecting the closest $x\%$ of room-period combinations to the disrupted assignment(s).

The remainder of the recovery model remains as described in the baseline model (Model 2). The complete bi-objective MIP formulation under this restricted flexibility is summarised in Model 5.

$$
\begin{array}{lll}
\min & f_{\mathsf{qual}}(x) & \text{(defined in (3.1)–(3.4))} \\
\min & f_{\Delta}(x, x^{\mathsf{init}}) & \text{(defined in (4.1))} \\
\text{s.t.} & \text{Static constraints from (3.5)–(3.13)} & \\
& \text{Disruption-specific constraints (4.2)–(4.5)} & \\
& \text{Restricted solution space recovery constraint (4.12)} &
\end{array}
$$

Model 5: Restricted solution space recovery model: bi-objective MIP formulation with dynamically constrained assignment availability.

This structured approach enables systematic evaluation of model robustness under progressively restricted recovery conditions.

4.3 Computational setup

We evaluate our recovery framework on the 21 benchmark instances from the Second International Timetabling Competition (ITC-2007). The characteristics of each instance are stated in Table 4.1 (Lindahl et al., 2019). The set of disruptions, which are randomly generated for each instance, is described in detail in Section 4.1.3.

Table 4.1: Characteristics of the 21 ITC-2007 instances.

Instance	Courses	Rooms	Timeslots	Curricula
comp01	30	6	30	14
comp02	82	16	25	70
comp03	72	16	25	68
comp04	79	18	25	57
comp05	54	9	36	139
comp06	108	18	25	70
comp07	131	20	25	77
comp08	86	18	25	61
comp09	76	18	25	75
comp10	115	18	25	67
comp11	30	5	45	13
comp12	88	11	36	150
comp13	82	19	25	66
comp14	85	17	25	60
comp15	72	16	25	68
comp16	108	20	25	71
comp17	99	17	25	70
comp18	47	9	36	52
comp19	74	16	25	66
comp20	121	19	25	78
comp21	94	18	25	78

All experiments were conducted on a high-performance computing (HPC) cluster running 64-bit Linux (kernel 5.14.0) on an AMD EPYC 7532 2.4 GHz processor with 16 cores and 62 GB of RAM. The models were implemented in Python 3.11.3 and solved using Gurobi Optimiser version 11.0.0 with default solver settings.

To construct the original timetable for each instance (Model 1), prior to any disruption, an initial optimisation was performed without recovery constraints, using a 24-hour time limit per instance. These solutions serve as the high-quality reference timetables against which recovery performance is evaluated.

A time limit of 3 hours (10.800 seconds) was imposed for each run of the Full Flexibility (Model 2), Room-Only (Model 3) and Period-Only (Model 4) recovery experiments. For the Restricted Solution Space recovery experiments (Model 5), a tighter time limit of 1 hour (3.600 seconds) was enforced, as the evaluation of multiple proximity thresholds per instance per disruption type proved computationally intensive. This choice balances tractability with comparability, ensuring that results remain meaningful despite the reduced runtime per instance. In all experiments, the solver was allowed to run until either the initial solution quality was fully recovered or the time limit was reached. If the initial solution quality could not be recovered within the time limit, the best solution found up to that point was retained and used for further analysis. As a consequence, the reported results do not necessarily reflect globally optimal solutions, but provide high-quality approximations that are sufficient for robust comparative analysis.

Chapter 5

Computational Results

This chapter presents the results of the computational experiments designed to evaluate the robustness and effectiveness of the proposed timetable recovery framework. We assess how different recovery mechanisms perform across various disruption types, using a comprehensive set of performance indicators including feasibility, solution quality, perturbation effort and runtime.

The chapter is structured as follows. Section 5.1 examines recovery performance under full perturbation flexibility and establishes a baseline. Sections 5.2 and 5.3 explore the impact of constrained mechanisms: room-only and period-only recovery. Section 5.4 provides a comparative summary of these two strategies across all disruption types. Section 5.5 further investigates robustness under progressively reduced solution space availability, simulating increasingly restrictive operational environments. Together, these sections highlight the trade-offs between recovery performance and flexibility across disruption scenarios.

5.1 Recovery under full perturbation flexibility

To interpret results in this chapter, the columns in the reported tables are clarified. *Instance* identifies the specific ITC-2007 test case, while *Disruption* describes the type of modification applied to the original schedule. f^{init} refers to the objective value of the original, undisrupted timetable (from Model 1). The remaining results are based on solving Model 2 for each instance under the corresponding disruption scenario. The column f_{Δ} indicates the number of changes (ϵ) made to restore feasibility. The column f_{qual} reflects the objective value (i.e., quality) of the recovered timetable after the disruption. The *Quality Recovered* column shows whether the original objective value was fully restored. Columns $f_{\text{qual}}^{\max}$ and $f_{\Delta}^{\max}$ report the best obtained solution quality and its corresponding number of allowed changes (ϵ). Finally, *Total Solve Time* and *Median Solve Time* refer to the total computation time across all iterations and the median time per iteration. One iteration corresponds to solving the model for one instance for a fixed value of ϵ. To support interpretation across instances, each table also includes a final row reporting the average values for all numeric columns. These averages provide a high-level summary of performance trends, but are primarily discussed and interpreted in the summary sections that follow each group of tables.

5.1.1 Feasibility, solution quality, perturbation effort and runtime

Table 5.1 for Invalid Assignment disruptions shows that all 21 instances successfully recovered a feasible solution under full flexibility. Moreover, every instance was able to return to its initial solution quality with a low amount of changes (f_{Δ}^{max}). This suggests that invalidating a single course assignment has limited impact on the feasibility and quality of the timetable when full flexibility in room and period reassignment is allowed. Most recoveries were achieved within a few seconds, although some harder instances (e.g., `comp16`) required significantly longer solve times.

Table 5.1: Detailed recovery performance per instance for Invalid Assignment (full flexibility).

Instance	Disruption (assignment invalid)	f^{init}	f_{Δ}	f_{qual}	Quality Recovered	f_{Δ}^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Course c0016, Period 14, Room B204	5	1	41	Yes	2	5	0.56	0.20
comp02	Course c0127, Period 15, Room B006	0	1	1	Yes	4	0	26.24	3.53
comp03	Course TeoRetEn, Period 23, Room B006	20	1	21	Yes	2	20	6.49	2.13
comp04	Course c0409, Period 5, Room L	8	1	9	Yes	3	8	17.91	4.10
comp05	Course StoVicOriAnt, Period 18, Room 14	176	1	177	Yes	3	176	8.14	2.34
comp06	Course c0896, Period 8, Room G	10	1	11	Yes	4	10	119.02	5.39
comp07	Course c0257, Period 18, Room G	2	1	15	Yes	8	2	2041.11	118.14
comp08	Course c0409, Period 23, Room 38	4	1	4	Yes	1	4	0.73	0.73
comp09	Course c0741, Period 5, Room E	12	1	13	Yes	9	12	571.65	31.69
comp10	Course c0808, Period 15, Room 51	0	1	1	Yes	3	0	22.79	4.70
comp11	Course c0045, Period 17, Room LUF2	0	1	4	Yes	2	0	1.06	0.45
comp12	Course PalLat, Period 22, Room DTM	182	1	183	Yes	4	182	563.72	62.47
comp13	Course c0441, Period 14, Room 27	10	1	11	Yes	8	10	486.24	34.84
comp14	Course c1031, Period 0, Room D	28	1	31	Yes	12	28	505.19	21.96
comp15	Course MecVibMn, Period 5, Room r52	20	1	21	Yes	2	20	5.93	1.98
comp16	Course c0519, Period 0, Room r50	2	1	8	Yes	10	2	5000.43	61.07
comp17	Course c1024, Period 20, Room rL	8	1	11	Yes	16	8	1510.83	18.08
comp18	Course LET-CLE-Glo, Period 13, Room rL	33	1	36	Yes	14	33	1144.99	19.69
comp19	Course c0053, Period 21, Room rB	6	1	7	Yes	3	6	20.80	3.46
comp20	Course c1077, Period 3, Room r31	0	1	1	Yes	4	0	85.85	12.03
comp21	Course c0176, Period 24, Room rH	2	1	5	Yes	6	2	203.83	14.83
Average		**25.14**	**1.00**	**29.10**		**5.71**	**25.14**	**587.78**	**20.18**

For Insert Curriculum disruptions, where a new curriculum consisting of four courses is added, all instances also recovered a feasible solution (Table 5.2). However, solution quality could be recovered in 19 out of 21 instances. `Comp05` and `comp16` failed to match the original quality. Solve times were generally higher compared to Invalid Assignments disruptions, reflecting the added complexity of inserting multiple linked courses.

Table 5.2: Detailed recovery performance per instance for Insert Curriculum (full flexibility).

Instance	Disruption (courses forming a curriculum)	f^{init}	f_Δ	f_{qual}	Quality Recovered	f_Δ^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	c0015, c0001, c0063, c0030	5	7	60	Yes	23	5	8628.89	1.71
comp02	c0297, c0559, c0058, c0201	0	3	11	Yes	12	0	528.86	11.65
comp03	RetCal2Gns, ComMecMn, TecCosAn, IdrAppCv	20	4	29	Yes	7	20	20.02	1.33
comp04	c0036, c0060, c0978, c0114	8	2	8	Yes	2	8	6.01	1.92
comp05	FilTesItaCS, StoArtMed1, MetRicArc, LinLetGre1	176	2	207	No	11	178	10595.40	205.98
comp06	c0484, c0212, c0046, c0422	10	2	22	Yes	16	10	10108.03	162.79
comp07	c0047, c1074, c1064, c0411	2	4	7	Yes	12	2	8221.47	4.51
comp08	c0882, c0249, c0953, c0409	4	5	6	Yes	11	4	327.76	3.85
comp09	c0953, c0534, c1007, c0149	12	1	14	Yes	2	12	5.33	1.58
comp10	c0072, c0230, c0132, c015a	0	2	1	Yes	3	0	27.36	4.35
comp11	c0007, c0045, c0008, c0109	0	3	31	Yes	6	0	4.21	0.43
comp12	StoCollstMus, ArcCla2, StoArcCon, StoRes	182	1	182	Yes	1	182	4.90	4.90
comp13	c0901, c0947, c0035, c0420	10	5	17	Yes	6	10	9.75	0.74
comp14	c1027, c0247, c0452, c0823	28	3	42	Yes	10	28	172.10	1.95
comp15	RetCal1G1n, StaAns, MecCom1Cv, StaPenAns	20	2	27	Yes	5	20	21.43	2.05
comp16	c0965, c0533, c0085, c0949	2	5	13	No	14	3	9362.38	12.83
comp17	c0051, c0199, c1027, c0495	8	1	56	Yes	5	8	230.67	22.56
comp18	CBC-LIB-InfDoc, CBC-ART-StoRes, LET-CST-StoRom, CBC-ART-MatCarCosResEdiSto	33	1	33	Yes	1	33	1.17	1.17
comp19	c0624, c0036, c0395, c0094	6	2	10	Yes	10	6	168.83	8.57
comp20	c0835, c0056, c0808, c0047	0	2	13	Yes	6	0	220.90	6.74
comp21	c0452, c0463, c0793, c0533	2	1	5	Yes	6	2	197.37	13.92
Average		**25.14**	**2.76**	**37.81**		**8.05**	**25.29**	**2326.80**	**22.65**

Removed Room disruptions (Table 5.3), where an entire room is made unavailable for a full day, resulted in even lower recovery quality. While all instances could still be repaired feasibly, 18 out of 21 achieved their original quality. Most instances require higher perturbation effort (f_Δ^{max}) compared to previous disruption types. Some instances, like `comp01` and `comp07`, illustrate the difficulty of absorbing room losses without considerable effort or degraded quality. Solve times were generally higher compared to Invalid Assignments disruptions, reflecting the added complexity of removing a room for a full day.

Table 5.3: Detailed recovery performance per instance for Removed Room (full flexibility).

Instance	Disruption (removed room for a day)	f^{init}	f_Δ	f_{qual}	Quality Recovered	f_Δ^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Removed Room C (index 1) for Day 4	5	6	183	No	20	83	11094.17	0.70
comp02	Removed Room 37 (index 1) for Day 0	0	3	3	Yes	8	0	133.10	7.14
comp03	Removed Room E (index 7) for Day 4	20	3	22	Yes	6	20	32.11	1.98
comp04	Removed Room B (index 7) for Day 2	8	5	17	Yes	20	8	1965.45	36.56
comp05	Room L (index 4) removed for Day 5	176	0	176	Yes	0	176	0.29	0.29
comp06	Room 37 (index 2) removed for Day 3	10	3	13	Yes	12	10	1424.05	43.49
comp07	Room G (index 10) removed for Day 1	2	5	107	No	16	5	10073.77	21.78
comp08	Room B (index 7) removed for Day 2	4	4	10	Yes	19	4	6126.96	40.91
comp09	Room 50 (index 14) removed for Day 4	12	5	15	Yes	9	12	84.87	2.41
comp10	Room 36 (index 1) removed for Day 3	0	3	2	Yes	7	0	124.25	7.06
comp11	Room O (index 3) removed for Day 4	0	3	1	Yes	5	0	3.01	0.29
comp12	Room O (index 7) removed for Day 2	182	3	187	Yes	10	182	4947.58	48.39
comp13	Room D (index 8) removed for Day 2	10	3	13	Yes	13	10	1050.01	16.64
comp14	Room 38 (index 3) removed for Day 4	28	2	30	Yes	5	28	30.11	3.19
comp15	Room rD (index 6) removed for Day 0	20	1	21	Yes	4	20	33.62	4.00
comp16	Room rA (index 11) removed for Day 3	2	3	4	Yes	13	2	2389.72	32.36
comp17	Room rDS2 (index 16) removed for Day 3	8	5	12	Yes	17	8	589.84	15.74
comp18	Room rC1 (index 2) removed for Day 0	33	3	36	Yes	10	33	148.14	5.27
comp19	Room r37 (index 1) removed for Day 4	6	1	6	Yes	1	6	0.50	0.25
comp20	Room r31 (index 4) removed for Day 2	0	3	4	No	6	1	8852.08	68.29
comp21	Room r27 (index 5) removed for Day 3	2	4	5	Yes	13	2	1194.21	23.72
Average		**25.14**	**3.24**	**41.29**		**10.19**	**29.05**	**2395.14**	**18.12**

Removed period disruptions (Table 5.4), where a random period is made unavailable, resulted in the lowest recovery quality. Although 20 out of 21 instances still yielded feasible solutions, only

2 managed to return to their original quality. Quality degradation is high and solve times were substantially higher compared to all other disruption types discussed above. This confirms that removing an entire period constitutes a particularly severe disruption. This is consistent with its broad impact on multiple assignments across the timetable, which increases both model complexity and computational effort.

Table 5.4: Detailed recovery performance per instance for Removed Period (full flexibility).

Instance	Disruption (unavailable period)	f^{init}	f_Δ	f_{qual}	Quality Recovered	f_Δ^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Period 4	5	6	104	No	28	6	10846.19	0.32
comp02	Period 1	0	10	87	No	30	15	11382.40	8.19
comp03	Period 7	20	11	85	No	34	37	11075.08	70.66
comp04	Period 7	8	12	52	No	38	9	10950.29	21.12
comp05	Period 16	176	NaN	NaN	No	NaN	NaN	11283.19	4.12
comp06	Period 18	10	16	70	No	29	36	10300.22	2.27
comp07	Period 10	2	19	113	No	34	36	7778.16	2.84
comp08	Period 7	4	13	42	No	41	5	11073.30	7.58
comp09	Period 14	12	12	91	No	40	34	11308.76	44.18
comp10	Period 18	0	15	119	No	34	31	10345.72	15.06
comp11	Period 28	0	3	29	Yes	21	0	35.42	0.90
comp12	Period 30	182	9	396	No	23	218	10840.85	9.82
comp13	Period 8	10	14	27	No	25	14	11029.77	8.47
comp14	Period 3	28	10	63	No	27	29	10121.00	13.81
comp15	Period 6	20	12	59	No	37	26	11108.47	12.35
comp16	Period 11	2	14	71	No	33	25	10709.61	24.60
comp17	Period 16	8	14	136	No	33	30	10849.12	10.89
comp18	Period 11	33	0.5	33	Yes	0.5	33	0.26	0.26
comp19	Period 21	6	15	60	No	48	20	11107.86	63.03
comp20	Period 23	0	14	249	No	29	20	11189.84	2.53
comp21	Period 5	2	15	168	No	37	45	10446.12	12.25
Average		**25.14**	**11.73**	**102.70**		**31.08**	**33.45**	**9703.89**	**15.96**

5.1.2 Summary of baseline recovery performance

Table 5.5 provides a comparative overview of baseline recovery performance across the four disruption types, under full perturbation flexibility. The values in the table are calculated based on aggregated results across all test instances $i \in I$, where $|I| = 21$. The feasibility rate (% Feasible Instances) is computed as the proportion of instances for which a feasible solution was found, given by $\frac{|\{i \in I | \text{feasible solution found}\}|}{|I|} \times 100$.

For the remaining metrics, only the subset of instances for which a feasible solution was found is considered. We denote this subset as $I' \subseteq I$. The quality recovery rate (% Quality recovered) then represents the percentage of feasible instances in which the recovered solution reaches the initial objective quality value, formally $\frac{|\{i \in I' | f_i^{\text{best}} = f_i^{\text{init}}\}|}{|I'|} \times 100$, where f_i^{init} and f_i^{best} denote the initial and recovered objective values.

To quantify the degradation in solution quality due to the disruption and recovery process, we compute the average quality decrease as $\frac{100}{|I'|} \sum_{i \in I'} \left(\frac{f_{\text{qual},i}^{\text{max}} - f_i^{\text{init}}}{f_i^{\text{init}}} \right)$. The average perturbation effort is calculated as $\frac{1}{|I'|} \sum_{i \in I'} f_{\Delta,i}^{\text{max}}$, where $f_{\Delta,i}^{\text{max}}$ denotes the number of changes needed to get the best recovered solution for instance i.

Finally, the median solve time per disruption type is computed as the average median runtime required to solve the model for a single value of ϵ, where ϵ denotes the number of allowed changes from the initial schedule. This reflects the typical computational effort needed for an instance when the model is solved under a fixed epsilon budget. In addition, the total solve time aggregates the cumulative runtime across all ϵ-levels and then averages this over all instances within a disruption type, providing a measure of the overall computational burden associated with each recovery scenario.

While this feasibility-based filtering ensures that quality and effort metrics reflect meaningful outcomes, it also means that metrics are calculated only over I', the subset of instances with feasible solutions. This may introduce a degree of skewness, as difficult disruption scenarios where recovery failed, are excluded from the analysis.

Table 5.5: Comparison of baseline recovery performance across disruption types under full perturbation flexibility.

Disruption Type	% Feasible Instances	% Quality Recovered	Avg. Quality Decrease (%)	Avg. Perturbation Effort (ϵ)	Total solve time (s)	Median Solve Time (s)
Invalid Assignment	100.0%	100.0%	0.0%	5.71	587.78	20.18
Insert Curriculum	100.0%	90.5%	0.6%	8.05	2326.80	22.64
Removed Room	100.0%	85.7%	15.5%	10.19	2395.14	18.12
Removed Period	95.2%	10.0%	90.1%	31.08	9703.89	15.96

A feasible solution was found for nearly all instances across disruption types, with full feasibility achieved in three out of four cases. This indicates that the model consistently succeeds in restoring feasibility when both room and period assignments can be modified. The proportion of instances with fully recovered quality drops from 100% for Invalid Assignment to 90.5% for Insert Curriculum, 85.7% for Removed Room and just 10.0% for Removed Period. This decline is mirrored in the average quality decrease, which rises from 0% for Invalid Assignment to 90.1% for Removed Period. A similar pattern is observed in the required perturbation effort: an Invalid Assignment disruption needed only 5.71 changes on average to recover quality, whereas a Removed Period disruption required more than 31. Total solve time captures the cumulative computational burden of solving across multiple perturbation budgets. Removed Period disruptions incurred the highest total runtime (9703.89 seconds), while Invalid Assignment disruptions required the least (587.78 seconds). Despite these differences in disruption severity, median solve times remained relatively low and stable, with no clear trend across types.

Taken together, these results show that the ability to recover is sensitive to the type of disruption. Recovery feasibility decreases steeply for more severe disruptions and the quality of recovered solutions suffers disproportionately. The most impactful disruptions (removing a period) trigger not only lower feasibility rates but also require significantly more changes and yield higher degradation in quality. Conversely, localised disruptions (invalidating a single assignment) result in much higher recovery success rates and require substantially fewer changes. These patterns emphasise the importance of preserving scheduling flexibility, especially when anticipating high-impact disruptions.

The results from this section highlight the limits of the model's robustness and serve as a baseline reference point for evaluating alternative recovery strategies in the subsequent experiments.

5.2 Room-only recovery

The previous section demonstrated that, under full perturbation flexibility, the recovery model is generally effective at restoring both feasibility and quality across a wide range of disruptions. However, this assumes that both room and period assignments can be freely adjusted. This section presents the results for the room-only recovery scenario, in which period assignments remain fixed and only room reassignments are permitted. The model is restricted accordingly and we evaluate its ability to restore feasibility and quality under this limited flexibility regime (Model 3).

5.2.1 Feasibility, solution quality, perturbation effort and runtime

Table 5.6 presents the detailed recovery performance for Invalid Assignment disruptions under the room-only recovery setting. Despite the constraint that only room assignments may be modified, all 21 instances succeeded in finding a feasible solution. Moreover, 19 out of 21 instances fully recovered their original objective quality. Compared to the full flexibility baseline (Table 5.1), where 100% of instances also achieved full recovery, the drop in solution quality recovery is minimal. This underlines the robustness of the model against isolated assignment disruptions, even under recovery conditions restricted to spatial adjustments.

Table 5.6: Detailed recovery performance per instance for Invalid Assignment (room-only recovery).

Instance	Disruption (assignment invalid)	f^{init}	f_Δ	f_{qual}	Quality Recovered	$f_\Delta^{\max}$	$f_{\text{qual}}^{\max}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Course c0016, Period 14, Room B	5	2	7	No	2	7	3096.53	0.05
comp02	Course c0127, Period 15, Room F	0	1	1	Yes	6	0	2.32	0.40
comp03	Course TeoRetEn, Period 23, Room G	20	1	21	Yes	6	20	2.88	0.44
comp04	Course c0409, Period 5, Room L	8	1	9	Yes	3	8	1.10	0.36
comp05	Course StoVicOriAnt, Period 18, Room 14	176	1	177	Yes	3	176	0.56	0.18
comp06	Course c0896, Period 8, Room G	10	1	71	Yes	39	10	45.23	0.91
comp07	Course c0257, Period 18, Room G	2	1	85	Yes	42	2	216.70	4.39
comp08	Course c0409, Period 23, Room 38	4	1	5	Yes	6	4	2.94	0.50
comp09	Course c0741, Period 5, Room E	12	1	13	Yes	14	12	5.47	0.39
comp10	Course c0808, Period 15, Room 51	0	1	1	Yes	3	0	0.98	0.34
comp11	Course c0045, Period 17, Room LUF2	0	1	4	No	1	4	2847.69	0.04
comp12	Course PalLat, Period 22, Room DTM	182	1	183	Yes	5	182	1.66	0.34
comp13	Course c0441, Period 14, Room 27	10	1	11	Yes	8	10	3.59	0.44
comp14	Course c1031, Period 0, Room D	28	1	49	Yes	27	28	15.06	0.42
comp15	Course MecVibMn, Period 5, Room r52	20	1	21	Yes	11	20	3.56	0.32
comp16	Course c0519, Period 0, Room r50	2	1	36	Yes	39	2	37.97	0.85
comp17	Course c1024, Period 20, Room rL	8	1	99	Yes	37	8	28.03	0.61
comp18	Course LET-CLE-Glo, Period 13, Room rL	33	2	35	Yes	22	33	3.27	0.15
comp19	Course c0053, Period 21, Room rB	6	1	7	Yes	15	6	6.15	0.41
comp20	Course c1077, Period 3, Room r31	0	1	1	Yes	36	0	69.66	1.79
comp21	Course c0176, Period 24, Room rH	2	1	46	Yes	17	2	9.61	0.59
Average		**25.14**	**1.10**	**42.00**		**16.29**	**25.43**	**304.81**	**0.66**

Table 5.7 presents the recovery outcomes for Insert Curriculum disruptions when only room reassignments are permitted. In stark contrast to the full flexibility setting (Table 5.2), where all 21 instances successfully recovered feasibility, only a single instance (`comp02`) yields a feasible solution under this restricted scenario. This extreme drop in recoverability is not coincidental. Insert Curriculum disruptions require the model to accommodate four new, interrelated courses within the existing timetable. Under the room-only restriction, period assignments are fixed, meaning any initial overlap in time between the inserted courses renders the disruption irrecoverable. This constraint leads to a fundamental infeasibility in most cases. The only exception, `comp02`, owes its feasibility to the fact that the four inserted courses were originally assigned to non-overlapping periods and compatible rooms. As such, a feasible solution could be constructed without triggering temporal conflicts. These results underscore the structural complexity of insertion-type disruptions and the critical importance of temporal flexibility.

Table 5.7: Detailed recovery performance per instance for Insert Curriculum (room-only recovery).

Instance	Disruption (courses forming a curriculum)	f^{init}	f_{Δ}	f_{qual}	Quality Recovered	f_{Δ}^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	c0067, c0001, c0030, c0004	5	NaN	nan	No	NaN	NaN	6098.92	0.03
comp02	c0113, c0201, c0066, c0298	0	0	0	Yes	0	0	0.43	0.43
comp03	DisComEsG2n, GeoAn, PriIngAns, FonChiAn	20	NaN	NaN	No	NaN	NaN	8087.22	0.15
comp04	c0527, c0224, c0884, c0447	8	NaN	NaN	No	NaN	NaN	7903.52	0.17
comp05	LetCriAnt, MetRicStoArt, StoArtMod1, IcoIcoB	176	NaN	NaN	No	NaN	NaN	8779.97	0.16
comp06	c0050, c1028, c0085, c0185	10	NaN	NaN	No	NaN	NaN	8696.15	0.37
comp07	c0058, c0127, c0009, c0095	2	NaN	NaN	No	NaN	NaN	7377.28	0.24
comp08	c0441, c0738, c0177, c0181	4	NaN	NaN	No	NaN	NaN	8184.37	0.21
comp09	c0972, c0378, c0923, c0149	12	NaN	NaN	No	NaN	NaN	8038.27	0.18
comp10	c0193, c0804, c0980, c0069	0	NaN	NaN	No	NaN	NaN	8988.55	0.38
comp11	c0042, c0045, c0034, c0079	0	NaN	NaN	No	NaN	NaN	6484.90	0.04
comp12	EtrAntIta, LinGreA, InfDoc, MetRicStoArt	182	NaN	NaN	No	NaN	NaN	9210.02	0.35
comp13	c0163, c0177, c0233, c0866	10	NaN	NaN	No	NaN	NaN	7999.93	0.18
comp14	c0051, c0452, c0463, c0474	28	NaN	NaN	No	NaN	NaN	8079.30	0.18
comp15	Mat1Cn, DisComEsG2n, TerAppMv, CosMacMn	20	NaN	NaN	No	NaN	NaN	7213.69	0.10
comp16	c1068, c0826, c0484, c0109	2	NaN	NaN	No	NaN	NaN	7301.62	0.20
comp17	c1042, c0006, c0051, c0935	8	NaN	NaN	No	NaN	NaN	8351.19	0.24
comp18	CBC-ART-StoCriArt, CBC-ART-StoMus, CBC-LIB-PalLat, LET-STO-StoMed2	33	NaN	NaN	No	NaN	NaN	7313.57	0.07
comp19	c0130, c0053, c0116, c0309	6	NaN	NaN	No	NaN	NaN	8213.49	0.17
comp20	c0537, c0118, c0020, c0195	0	NaN	NaN	No	NaN	NaN	9040.81	0.44
comp21	c0600, c0049, c0011, c0433	2	NaN	NaN	No	NaN	NaN	8074.38	0.24
Average		**25.14**	**0.00**	**0.00**		**0.00**	**0.00**	**7592.27**	**0.22**

Table 5.8 summarises the recovery performance for the scenario in which a room is made unavailable for a full day, evaluated under room-only recovery. Compared to the full flexibility setting (Table 5.3), the room-only constraint leads to a small drop in feasibility. While all 21 instances were solvable under full flexibility, 19 out of 21 recover to a feasible solution when period reassignments are disallowed. The two infeasible cases (`comp01` and `comp12`) highlight scenarios where spatial reassignments alone are insufficient to overcome the disruption.

Among the 19 feasible instances, 12 successfully recover their initial objective value. This indicates that although room-only recovery restricts the solution space, it often retains enough flexibility to even fully restore quality. However, the remaining 7 feasible instances show some level of degradation, reflecting the limitations of spatial adjustments in absorbing capacity constraints caused by room-for-a-day removals. Instances like `comp04` and `comp13` experience some quality degradation despite large epsilon budgets (`comp04:` $\epsilon = 69$, `comp13:` $\epsilon = 69$).

Overall, the findings confirm that room-only recovery can be a viable strategy in many - removed room for a period of time - scenarios. While some quality loss is observed in harder cases, the ability to restore feasibility and even recover full quality in some instances highlights the robustness of the recovery model under spatial constraints.

Table 5.8: Detailed recovery performance per instance for Removed Room for a day (room-only recovery).

Instance	Disruption (room removed for a day)	f^{init}	f_Δ	f_{qual}	Quality Recovered	f_Δ^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Room C (index 1) removed for Day 4	5	NaN	NaN	No	NaN	NaN	6771.64	0.04
comp02	Room 37 (index 1) removed for Day 0	0	3	3	Yes	11	0	4.00	0.35
comp03	Room E (index 7) removed for Day 4	20	3	22	Yes	12	20	2.89	0.26
comp04	Room B (index 7) removed for Day 2	8	5	31	No	69	12	8998.67	0.26
comp05	Room L (index 4) removed for Day 5	176	1	176	Yes	1	176	0.27	0.27
comp06	Room 37 (index 2) removed for Day 3	10	3	13	Yes	15	10	44.11	1.58
comp07	Room G (index 10) removed for Day 1	2	5	207	No	21	10	10277.74	6.12
comp08	Room B (index 7) removed for Day 2	4	4	68	No	20.5	7	9241.04	0.33
comp09	Room 50 (index 14) removed for Day 4	12	5	15	Yes	11	12	12.12	0.54
comp10	Room 36 (index 1) removed for Day 3	0	3	3	Yes	11.5	0	30.19	1.52
comp11	Room O (index 3) removed for Day 4	0	3	3	No	6	2	9449.05	0.11
comp12	Room O (index 7) removed for Day 2	182	NaN	NaN	No	NaN	NaN	9149.29	0.31
comp13	Room D (index 8) removed for Day 2	10	3	53	No	69	30	9105.89	0.30
comp14	Room 38 (index 3) removed for Day 4	28	2	30	Yes	3	28	2.37	0.39
comp15	Room rD (index 6) removed for Day 0	20	1	21	Yes	3	20	1.92	0.33
comp16	Room rA (index 11) removed for Day 3	2	3	5	Yes	35.5	2	170.55	2.38
comp17	Room rDS2 (index 16) removed for Day 3	8	5	21	Yes	36	8	151.34	1.35
comp18	Room rC1 (index 2) removed for Day 0	33	3	36	No	3.5	34	7288.97	0.07
comp19	Room r37 (index 1) removed for Day 4	6	1	7	Yes	1.5	6	1.00	0.36
comp20	Room r31 (index 4) removed for Day 2	0	3	13	No	23.5	1	10527.85	1.35
comp21	Room r27 (index 5) removed for Day 3	2	4	5	Yes	8.5	2	9.40	0.53
Average		**25.14**	**3.16**	**38.53**		**29.53**	**20.00**	**3868.59**	**0.89**

Room-only recovery is largely infeasible for *Removed Period* disruptions. Since the disrupted time slot is explicitly invalidated and the recovery mechanism prohibits any changes in period assignments, it is structurally impossible to reassign courses that were originally scheduled in that period. As a result, no feasible solution can be found in nearly all instances. The only exception is `comp18`, where the removed period did not contain any originally scheduled courses. This highlights that recovery only remains possible when the disrupted time slot does not intersect with the initial non-disrupted timetable.

Table 5.9: Detailed recovery performance per instance for Removed Period (room-only recovery).

Instance	Disruption (unavailable period)	f^{init}	f_Δ	f_{qual}	Quality Recovered	f_Δ^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Period 4	5	NaN	NaN	No	NaN	NaN	3465.08	0.01
comp02	Period 1	0	NaN	NaN	No	NaN	NaN	7024.69	0.11
comp03	Period 7	20	NaN	NaN	No	NaN	NaN	6325.87	0.08
comp04	Period 7	8	NaN	NaN	No	NaN	NaN	6580.59	0.10
comp05	Period 16	176	NaN	NaN	No	NaN	NaN	7909.56	0.12
comp06	Period 18	10	NaN	NaN	No	NaN	NaN	7386.30	0.18
comp07	Period 10	2	NaN	NaN	No	NaN	NaN	7704.37	0.26
comp08	Period 7	4	NaN	NaN	No	NaN	NaN	6659.84	0.11
comp09	Period 14	12	NaN	NaN	No	NaN	NaN	6896.82	0.11
comp10	Period 18	0	NaN	NaN	No	NaN	NaN	7777.58	0.23
comp11	Period 28	0	NaN	NaN	No	NaN	NaN	3818.60	0.02
comp12	Period 30	182	NaN	NaN	No	NaN	NaN	8913.28	0.31
comp13	Period 8	10	NaN	NaN	No	NaN	NaN	6896.54	0.13
comp14	Period 3	28	NaN	NaN	No	NaN	NaN	6784.86	0.11
comp15	Period 6	20	NaN	NaN	No	NaN	NaN	6985.56	0.10
comp16	Period 11	2	NaN	NaN	No	NaN	NaN	7693.20	0.22
comp17	Period 16	8	NaN	NaN	No	NaN	NaN	7041.77	0.14
comp18	Period 11	33	0	33	Yes	0	33	0.22	0.22
comp19	Period 21	6	NaN	NaN	No	NaN	NaN	6685.84	0.09
comp20	Period 23	0	NaN	NaN	No	NaN	NaN	7513.26	0.23
comp21	Period 5	2	NaN	NaN	No	NaN	NaN	7264.05	0.15
Average		**25.14**	**0.00**	**33.00**		**0.00**	**33.00**	**6539.42**	**0.14**

5.2.2 Summary and key observations

A disruption-specific overview of recovery performance under room-only recovery is provided in Table 5.10. The same evaluation metrics are used as in the full flexibility summary section 5.1.2. This structure allows for a detailed comparison with the full flexibility baseline results in Table 5.5, highlighting the impact of removing temporal flexibility across different disruption types.

Table 5.10: Comparison of recovery performance across disruption types under room-only recovery.

Disruption Type	% Feasible Instances	% Quality Recovered	Avg. Quality Decrease (%)	Avg. Perturbation Effort (ϵ)	Total solve time (s)	Median Solve Time (s)
Invalid Assignment	100.0%	90.5%	1.1%	16.29	304.81	0.66
Insert Curriculum	4.8%	100.0%	0.0%	0.00	7592.27	0.22
Removed Room	90.5%	63.2%	11.4%	29.53	3868.59	0.89
Removed Period	4.8%	100.0%	0.0%	0.00	6539.42	0.14

Invalid Assignment disruptions are handled well under room-only recovery. Feasibility remains at 100% and the quality recovery rate remains high at 90.5%. The average quality decrease is minimal (1.1%) and average perturbation effort triples from 5.71 to 16.29. Total solve time stays low (304.81s), showing that the solver is well-guided under room-only constraints. Median iteration time drops significantly (20.18s to 0.66s), likely due to quick convergence.

In contrast, *Insert Curriculum* disruptions are nearly impossible to recover under room-only recovery. Feasibility drops from 100% to 4.8%. Total solve time is very high (7592.27s) and the median solve time is low (0.22s), due to many rapid infeasible runs. Insert Curriculum disruptions require integrating four new, often interdependent courses into the timetable. Under room-only constraints, period assignments are fixed and any temporal overlap between newly inserted courses results in fundamental infeasibility. These findings confirm that time flexibility is indispensable when inserting new curricula.

Removed Room disruptions are partially robust under room-only recovery. Feasibility drops slightly from 100% to 90.5% and quality recovery declines from 85.7% to 63.2%. However, among recovered instances, solution quality is fairly well preserved (11.4%). Perturbation effort almost triples (10.21 to 29.53), indicating the need for room reshuffling. Total solve time slightly rises (3868.59s) and the median iteration time remains low (0.89s).

Removed Period disruptions are almost completely unrecoverable under room-only constraints. Feasibility plummets from 95.2% to 4.8%. The only instance that could be recovered (`comp18`) corresponds to a situation where the removed period did not contain any assignments in the initial solution. This illustrates that feasibility is only possible when the disruption does not intersect with the original schedule. Time flexibility proves essential to restore feasibility after period-level disruptions. Total solve time is high (6539.42s) and median iteration time is extremely low (0.14s), reflecting many early-terminated runs.

Overall, these results emphasise that room-only recovery mechanisms are well suitable for *Invalid Assignment* disruptions. However, for disruptions like *Insert Curriculum* and *Removed Period*, temporal flexibility is critical to ensure feasibility and maintain solution quality.

5.3 Period-only recovery

This section investigates recovery performance under a restrictive setting in which only period assignments may be changed, while room assignments are fixed to their values in the initial solution (Model 4). This constraint reflects scenarios where spatial flexibility is limited.

5.3.1 Feasibility, solution quality, perturbation effort and runtime

Table 5.11 presents the detailed recovery performance for Invalid Assignment disruptions under the period-only recovery setting. While the model recovers a feasible schedule in 13 out of 21 instances, it fails to restore the initial objective quality in nearly all of these cases. Only instance `comp01` recovers to its initial quality. This consistent drop in quality shows that *period flexibility* alone is insufficient to completely neutralise the effects of invalidated assignments. Although the disruption affects only a single assignment, the limited flexibility of period-only recovery often leads to considerable quality degradation.

Table 5.11: Detailed recovery performance per instance for Invalid Assignment (period-only recovery).

Instance	Disruption (assignment invalid)	f^{init}	f_{Δ}	f_{qual}	Quality Recovered	f_{Δ}^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Course c0016, Period 14, Room B204	5	6	19	Yes	9	5	0.71	0.05
comp02	Course c0127, Period 15, Room B006	0	4	13	No	35	5	7867.70	0.13
comp03	Course TeoRetEn, Period 23, Room B006	20	8	53	No	15	23	11286.11	21.27
comp04	Course c0409, Period 5, Room L	8	7	34	No	19	13	11280.96	9.29
comp05	Course StoVicOriAnt, Period 18, Room 14	176	1	186	No	3	179	9758.24	130.48
comp06	Course c0896, Period 8, Room G	10	NaN	NaN	No	NaN	NaN	10608.14	1.24
comp07	Course c0257, Period 18, Room G	2	NaN	NaN	No	NaN	NaN	10546.66	1.57
comp08	Course c0409, Period 23, Room 38	4	24	101	No	103	13	11326.07	15.66
comp09	Course c0741, Period 5, Room E	12	20	98	No	72	34	10840.35	10.08
comp10	Course c0808, Period 15, Room 51	0	NaN	NaN	No	NaN	NaN	10466.31	1.18
comp11	Course c0045, Period 17, Room LUF2	0	NaN	NaN	No	NaN	NaN	8229.97	0.07
comp12	Course PalLat, Period 22, Room DTM	182	1	199	No	11	194	11011.75	157.46
comp13	Course c0441, Period 14, Room 27	10	NaN	NaN	No	NaN	NaN	9589.43	0.40
comp14	Course c1031, Period 0, Room D	28	23	142	No	59	49	10789.25	9.58
comp15	Course MecVibMn, Period 5, Room r52	20	13	86	No	59	26	11144.65	20.27
comp16	Course c0519, Period 0, Room r50	2	33	128	No	74	29	10316.69	4.89
comp17	Course c1024, Period 20, Room rL	8	24	112	No	60	32	10966.69	6.49
comp18	Course LET-CLE-Glo, Period 13, Room rL	33	2	44	No	8	35	11200.89	30.27
comp19	Course c0053, Period 21, Room rB	6	NaN	NaN	No	NaN	NaN	10270.10	0.56
comp20	Course c1077, Period 3, Room r31	0	NaN	NaN	No	NaN	NaN	10716.57	1.72
comp21	Course c0176, Period 24, Room rH	2	NaN	NaN	No	NaN	NaN	10253.76	0.75
Average		**25.14**	**12.77**	**93.46**		**40.54**	**49.00**	**9927.19**	**20.16**

Table 5.12 presents the recovery performance for Insert Curriculum disruptions under the period-only recovery mechanism. Despite the complexity of introducing multiple new courses, all 21 instances succeeded in finding a feasible solution. Moreover, 18 out of 21 instances fully recovered their original objective value. This indicates that period flexibility alone is sufficient to absorb the scheduling pressure introduced by curriculum insertions, as long as room assignments remain available to accommodate the new courses.

Table 5.12: Detailed recovery performance per instance for Insert Curriculum (period-only recovery).

Instance	Disruption (courses forming a curriculum)	f^{init}	f_{Δ}	f_{qual}	Quality Recovered	f_{Δ}^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	c0067, c0066, c0068, c0015	5	10	29	Yes	16	5	3.77	0.06
comp02	c0119, c0323, c0020, c0316	0	2	2	Yes	3	0	1.54	0.24
comp03	ChiGenAn, CalProG2n, Idr2Cn, ChiStaEn	20	1	24	Yes	20	20	1611.67	22.61
comp04	c0825, c0972, c0148, c0218	8	9	24	Yes	15	8	35.35	0.26
comp05	StoArtCon1, FilSem, StoCriChi, LinLetLat1	176	2	185	No	14	179	10913.65	161.79
comp06	c0185, c1068, c0935, c0026	10	2	40	Yes	15	10	480.00	10.52
comp07	c0221, c0132, c0023, c0253	2	5	12	Yes	18	2	1976.05	44.59
comp08	c0884, c0447, c0409, c0851	4	8	12	Yes	11	4	8.09	0.34
comp09	c1007, c0179, c0214, c0465	12	6	51	Yes	18	12	85.56	1.01
comp10	c0201, c0602, c0108, c0723	0	2	11	Yes	5	0	8.50	0.46
comp11	c0027, c0017, c0109, c0115	0	3	0	Yes	3	0	0.37	0.06
comp12	LegBenCul2, Esteti, Sociol1, OrientCS	182	2	201	No	7	186	10246.27	131.87
comp13	c0491, c0745, c0098, c0214	10	2	38	Yes	10	10	37.68	1.01
comp14	c0105, c0935, c0439, c0462	28	2	43	Yes	34	28	5163.56	40.50
comp15	LabEle2Ar, FonChiAn, IngConAn, TeoStr1Cv	20	2	34	Yes	6	20	2.61	0.21
comp16	c0228, c0846, c1042, c0896	2	2	6	Yes	5	2	9.62	0.35
comp17	c0097, c0231, c1058, c0422	8	5	36	Yes	22	8	688.02	8.20
comp18	CBC-LIB-InfDoc, CBC-ARC-Palantro, LET-STO-StoMed2, LET-CLE-FilCla	33	1	35	Yes	2	33	0.55	0.15
comp19	c0331, c0395, c0160, c0344	6	15	67	No	48	10	11328.53	45.47
comp20	c0496, c0478, c0024, c0936	0	2	36	Yes	20	0	1753.96	39.57
comp21	c0935, c0830, c0792, c0433	2	3	14	Yes	38	2	7788.39	63.02
Average		**25.14**	**4.10**	**42.86**		**15.71**	**25.67**	**2483.04**	**27.25**

Table 5.13 presents the recovery performance for Removed Room disruptions under the period-only recovery mechanism. A feasible solution is found in 16 out of 21 instances, indicating that temporal

flexibility alone is often sufficient to restore feasibility. 10 of these feasible instances succeed in recovering the initial objective quality. This shows that while adjusting periods allows for basic schedule repair, restoring the initial quality remains challenging when an entire room is removed for a full day, particularly in instances where room capacity or suitability was already a limiting factor.

Table 5.13: Detailed recovery performance per instance for Removed Room (period-only recovery).

Instance	Disruption (room removed for a day)	f^{init}	f_{Δ}	f_{qual}	Quality Recovered	f_{Δ}^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Room C (index 1) for Day 4	5	NaN	NaN	No	NaN	NaN	7550.05	0.05
comp02	Room 37 (index 1) for Day 0	0	NaN	NaN	No	NaN	NaN	10623.22	0.94
comp03	Room E (index 7) for Day 4	20	4	24	Yes	6	20	2.32	0.18
comp04	Room B (index 7) for Day 2	8	NaN	NaN	No	NaN	NaN	9396.09	0.33
comp05	Room L (index 4) for Day 5	176	0.5	176	Yes	0.5	176	0.24	0.24
comp06	Room 37 (index 2) for Day 3	10	3	31	No	38	15	11361.95	13.21
comp07	Room G (index 10) for Day 1	2	NaN	NaN	No	NaN	NaN	10333.66	1.09
comp08	Room B (index 7) for Day 2	4	5	29	No	29	6	11335.00	12.20
comp09	Room 50 (index 14) for Day 4	12	8	25	No	21	14	11155.49	28.08
comp10	Room 36 (index 1) for Day 3	0	4	20	Yes	29	0	3061.91	40.56
comp11	Room O (index 3) for Day 4	0	3	2	Yes	5	0	1.43	0.08
comp12	Room O (index 7) for Day 2	182	7	261	No	23	205	9316.21	37.75
comp13	Room D (index 8) for Day 2	10	4	31	No	15	15	11319.86	57.53
comp14	Room 38 (index 3) for Day 4	28	2	41	Yes	12	28	99.65	1.25
comp15	Room rD (index 6) for Day 0	20	1	28	Yes	28	20	1913.91	19.48
comp16	Room rA (index 11) for Day 3	2	4	14	Yes	19	2	185.84	3.56
comp17	Room rDS2 (index 16) for Day 3	8	NaN	NaN	No	NaN	NaN	9726.07	0.48
comp18	Room rC1 (index 2) for Day 0	33	3	39	Yes	17	33	185.54	3.95
comp19	Room r37 (index 1) for Day 4	6	1	6	Yes	1	6	0.35	0.17
comp20	Room r31 (index 4) for Day 2	0	4	22	Yes	32	0	6773.29	68.25
comp21	Room r27 (index 5) for Day 3	2	7	53	No	44	20	11327.88	58.06
Average		**25.14**	**3.78**	**50.13**		**19.97**	**35.00**	**5984.29**	**16.54**

Table 5.14 displays the recovery results for Removed Period disruptions under the period-only recovery setting. In this setting, the disruption removes a specific period from the timetable, while the recovery process is restricted to reassigning courses to other periods. Despite the hard constraint of disallowing room reassignments and the severe disruption of removing a period, 17 out of 21 instances still manage to reach a feasible solution. However, only 3 of these recover their original objective value, highlighting the limited effectiveness of period-only adjustments in the face of a severe disruption.

Table 5.14: Detailed recovery performance per instance for Removed Period (period-only recovery).

Instance	Disruption (unavailable period)	f^{init}	f_{Δ}	f_{qual}	Quality Recovered	f_{Δ}^{max}	$f_{\text{qual}}^{\text{max}}$	Total Solve Time (s)	Median Solve Time (s)
comp01	Period 4	5	NaN	NaN	No	NaN	NaN	8865.51	0.08
comp02	Period 1	0	NaN	NaN	No	NaN	NaN	10498.93	0.77
comp03	Period 7	20	13	135	No	50	32	11120.12	29.97
comp04	Period 7	8	15	56	Yes	52	8	1933.79	6.66
comp05	Period 16	176	NaN	NaN	No	NaN	NaN	11267.09	3.82
comp06	Period 18	10	NaN	NaN	No	NaN	NaN	10539.80	1.23
comp07	Period 10	2	29	121	No	62	24	10747.60	15.60
comp08	Period 7	4	13	42	No	41	5	11347.20	7.23
comp09	Period 14	12	12	91	No	38	36	11138.90	37.41
comp10	Period 18	0	15	119	No	34	31	10255.90	12.69
comp11	Period 28	0	3	29	Yes	21	0	36.59	0.67
comp12	Period 30	182	9	396	No	23	218	10911.60	12.82
comp13	Period 8	10	14	27	No	25	14	11213.50	9.42
comp14	Period 3	28	10	63	No	27	29	9806.08	17.08
comp15	Period 6	20	12	59	No	37	26	11033.90	11.44
comp16	Period 11	2	14	71	No	33	25	10580.30	26.81
comp17	Period 16	8	14	136	No	32	31	10573.80	7.81
comp18	Period 11	33	0	33	Yes	0.5	33	0.29	0.29
comp19	Period 21	6	15	60	No	48	20	11229.70	52.28
comp20	Period 23	0	14	249	No	28	21	10531.00	2.38
comp21	Period 5	2	15	168	No	37	45	10420.90	14.78
Average		**25.14**	**12.76**	**109.12**		**34.18**	**35.18**	**9240.59**	**12.92**

5.3.2 Summary and key observations

A disruption-specific overview of recovery performance under period-only recovery is provided in Table 5.15. The same evaluation metrics are used as in the full flexibility summary section (Section 5.1.2). This consistent structure allows for a direct comparison with the baseline results in Table 5.5, highlighting the impact of removing spatial flexibility across different disruption types.

Table 5.15: Comparison of recovery performance across disruption types under period-only recovery.

Disruption Type	% Feasible Instances	% Quality Recovered	Avg. Quality Decrease (%)	Avg. Perturbation Effort (ϵ)	Total solve time (s)	Median Solve Time (s)
Invalid Assignment	61.9%	7.7%	27.9%	40.54	9927.19	20.16
Insert Curriculum	100.0%	85.7%	2.0%	15.71	2483.04	27.25
Removed Room	76.2%	62.5%	10.9%	19.97	5984.29	16.54
Removed Period	81.0%	17.6%	43.6%	34.18	9240.59	12.92

Recovery performance deteriorates under period-only recovery for *Invalid Assignment* disruptions. Feasibility drops from 100% to 61.9%, indicating that spatial flexibility is sometimes required even for these localised disruptions. The percentage of instances that fully recover the initial solution quality collapses from 100% to just 7.7%. For the instances that recover, average quality objective degradation rises (0.0% to 27.9%) and average perturbation effort increases (from 5.71 to 40.54). Total solve time increases sharply (from 587.78s to 9927.19s) and median solve time remains nearly unchanged (20.18s vs. 20.16s). These findings underscore that invalid assignment disruptions are not always trivial to repair under period-only constraints, as both feasibility and quality can suffer considerably in the absence of room flexibility.

In contrast, *Insert Curriculum* disruptions remain highly recoverable under period-only constraints. Feasibility stays at 100% and the quality recovery rate only slightly declines (90.5% to 85.7%). The average quality decrease remains low (0.6% vs. 2.0%) and average perturbation effort increases moderately (from 8.05 to 15.71). Total and median solve times are also comparable to the baseline (2326.80s to 2483.04s and 22.64s to 27.25s). These results state that temporal flexibility is typically sufficient to absorb newly inserted curricula, provided the original spatial configuration is maintained.

Period-only recovery offers partial robustness for *Removed Room* disruptions. Feasibility drops from 100% to 76.2%, while quality recovery falls from 85.7% to 62.5%. Among feasible cases, solution quality is largely maintained (10.9%), indicating limited degradation relative to the baseline. However, average perturbation effort almost doubled (10.21 to 19.97). Total Solve time increases (2395.14s to 5984.29s) and median solve time remains comparable (18.12s to 16.54s). These findings show that while some room removals can be handled through time adjustments, room flexibility is sometimes required as well in handling this disruption.

Removed Period disruptions can be handled under period-only recovery. Feasibility declines only slightly from 95.2% to 81.0% and quality recovery stays low (17.6%). However, the average quality decrease remains stable (43.6%). Average perturbation effort rises only slightly (31.08 to 34.18) and total solve time remains high (9240.59s) with a low median iteration time (12.92s). Despite the severity of *Removed Period* disruptions, period-only recovery performance remains surprisingly close to the baseline with only moderate drops in feasibility and manageable increases in average perturbation effort and quality degradation.

Overall, these comparisons reaffirm the importance of allowing both spatial and temporal adjustments. While period-only recovery performs closest to the full flexibility baseline in cases such as *Insert Curriculum* and, to a lesser extent, *Removed Period*, it proves less effective for disruptions like *Invalid Assignment*, where spatial reassignment is more critical.

5.4 Comparative summary

Comparing period-only versus room-only recovery reveals how each type of recovery (temporal or spatial) contributes to disruption management across different scenarios.

Invalid Assignment disruptions are more effectively handled through room-only recovery. Feasibility and quality recovery remain high when only room changes are permitted, whereas period-only recovery suffers substantial declines in both. This suggests that spatial flexibility is generally sufficient (and sometimes essential) for resolving localised assignment failures.

Insert Curriculum disruptions show the opposite pattern. Period-only recovery retains strong performance, while room-only recovery is nearly always infeasible. This highlights that temporal flexibility is indispensable when integrating multiple interdependent courses into the timetable.

For *Removed Room* disruptions, both room-only and period-only recovery provide partial robustness. While feasibility and quality recovery differ moderately across the two settings, neither mechanism alone consistently guarantees recovery. This implies that both spatial and temporal adjustments play a complementary role in addressing this type of disruption.

Removed Period disruptions benefit clearly from period-only recovery. Room-only recovery is nearly always infeasible, only being feasible when the removed period did not intersect with the initial schedule. Although quality recovery remains limited even with period flexibility, it remains the viable option when temporal disruptions occur.

Overall, these comparisons reaffirm that the most effective recovery mechanism is disruption-dependent. While room-only recovery suffices for localised assignment changes, period-only flexibility is essential for broader temporally driven disruptions such as curriculum insertions or period removals.

5.5 Recovery under restricted solution space

This section presents the results for recovery under restricted solution space, where only a subset of the closest room-period combinations is available for rescheduling. The model is constrained accordingly and we assess how reduced flexibility affects feasibility, quality and perturbation effort across disruption types (Model 5). We structure this section as follows: First, we clearly outline the specific performance indicators used for assessing robustness in Section 5.5.1. Next, we present detailed visual analyses of the results in Section 5.5.2, followed by a comparative summary in tabular form in Section 5.5.3.

5.5.1 Performance indicators

To evaluate and compare the robustness of the timetabling model under restricted solution space conditions across different disruption types, we report three key performance indicators.

Recovery at $s\%$ solution space $\boldsymbol{R_s}$ quantifies the proportion of instances that recovered a feasible solution when the solution space was restricted to the $s\%$ of room–period combinations closest to the disrupted assignment(s). This metric is computed as:

$$R_s = 100 \times (\frac{Q_s}{N})$$

where Q_s denotes the number of instances for which a feasible solution was recovered at solution space level $s\%$ and N is the total number of tested instances (here, $N = 21$). This ratio serves as a direct indicator of robustness under a highly constrained environment.

Average quality decrease compared to baseline $\mathbf{Q}_s$ reflects the mean relative degradation in objective value due to reduced flexibility at solution space level $s < 100\%$. The quality decrease is computed as:

$$\mathsf{Q}_s = \frac{100}{|I'|} \sum_{i \in I'} \left(\frac{f_{i,s}^{\text{best}} - f_{i,100\%}^{\text{best}}}{f_{i,100\%}^{\text{best}}} \right)$$

where $f_{i,s}^{\text{best}}$ denotes the objective value of the best-quality solution found at solution space level s within the one-hour time limit. This formulation only averages over the subset $I' \subseteq I$ of instances that successfully recovered a feasible solution at level s. The value $f_{i,100\%}^{\text{best}}$ represents the best solution quality obtained at full solution space (i.e., 100% flexibility), corresponding to $f_{\text{qual}}^{\text{max}}$ as defined in Section 5.1.

Average perturbation effort $\boldsymbol{E_s}$ measures the required effort when reducing the solution space from 100% to $s\%$. The metric is calculated as:

$$E_s = \frac{1}{|I'|} \sum_{i \in I'} \epsilon_{i,s}^{\text{best}}$$

where $\epsilon_{i,s}^{\text{best}}$ denotes, for each instance $i \in I'$, the smallest perturbation bound ϵ at solution space level s for which the best-quality solution was found within the given time limit. Again, this formulation averages only over the subset $I' \subseteq I$ of instances that successfully recovered a feasible solution at level s. Note that this value does not reflect the minimum number of changes required to restore feasibility, but rather the ϵ-level associated with the best-quality solution identified during the recovery process. As this thesis focuses on robustness rather than feasibility alone, the analysis prioritizes the effort needed to recover high-quality solutions, rather than the minimal effort required to achieve feasibility. The value $\epsilon_{i,100\%}^{\text{best}}$ corresponds to the epsilon bound at full solution space (i.e., 100% flexibility) and matches the f_{Δ}^{max} value reported in Section 5.1 for each instance and disruption type.

In addition, we report the *load factor* of each instance as a contextual variable. While not a performance metric itself, it provides insight into the density of the original timetable and helps interpret variations in recovery performance across instances.

The *load factor* captures the density of the original (non-disrupted) timetable by measuring the proportion of room-period combinations that are used in the original solution:

$$\text{Load Factor} = \frac{|\mathcal{V}^{\text{init}}|}{|R| \cdot |P|}$$

where $\mathcal{V}^{\text{init}} \subseteq R \times P$ is the set of unique room-period pairs used in the original schedule:

$$\mathcal{V}^{\text{init}} = \{(r', p') \mid \exists\, c' \text{ such that } (c', p', r') \in \mathcal{A}^{\text{init}}\}$$

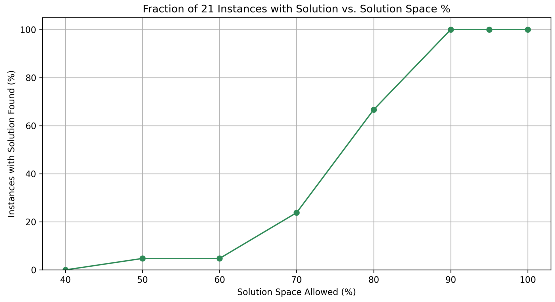
A higher load factor indicates that the timetable is densely packed, leaving less flexibility for recovery after disruption. These indicators will be mostly evaluated and discussed after presenting the detailed experimental outputs across disruption types.

5.5.2 Discussion of experimental results

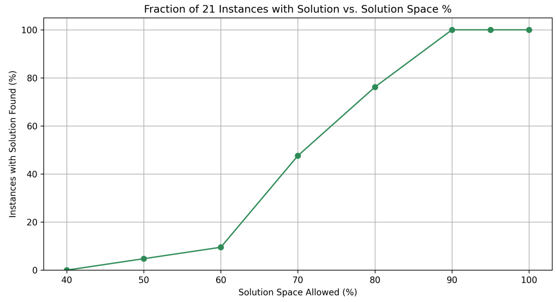
Unlike the full flexibility and previous restricted recovery strategies, which yield for each instance a single solution per disruption, the solution space recovery setup generates a series of solutions for each instance. This is because the recovery process is repeated across multiple levels of allowed flexibility (i.e., different percentages of the solution space). As a result, the output consists of a sequence of solutions that trace the trade-off between solution quality and available flexibility, allowing for a more detailed robustness analysis.

Feasibility across disruption types

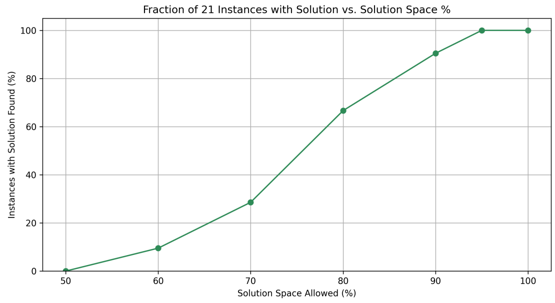
The evolution of feasibility as a function of solution space reduction is shown in Figure 5.1, separately for each disruption type. Each plot shows the evolution of recovery feasibility as a function of solution space reduction, separately for each disruption type. The x-axis represents the percentage of available solution space relative to the baseline, while the y-axis indicates the proportion of instances that successfully recover a feasible solution within the one-hour time limit. For each solution space level, the plotted value reflects the fraction of instances for which a feasible solution was found, providing a direct visualisation of how robustness deteriorates as flexibility decreases. The analysis includes solution space levels from 100% down to 40%, evaluated in steps of 10 percentage points. This step size offers a good balance between computational effort and granularity of insight. To assess whether even minimal restrictions already affect recovery performance, a 95% level was also included. Levels below 40% were excluded, as feasibility dropped to zero across all disruption types, making further reductions uninformative for comparative analysis.



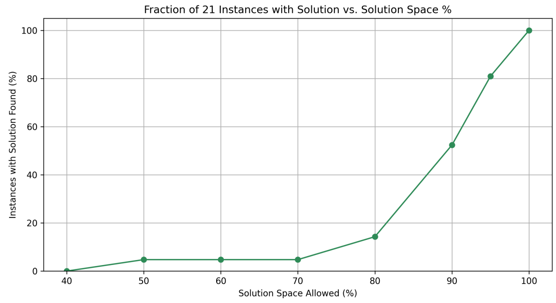
(a) Invalid Assignment



(b) Insert Curriculum



(c) Removed Room

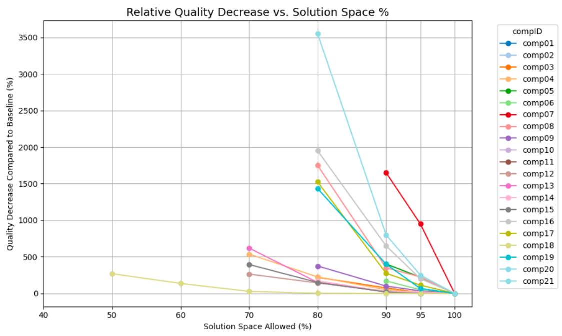


(d) Removed Period

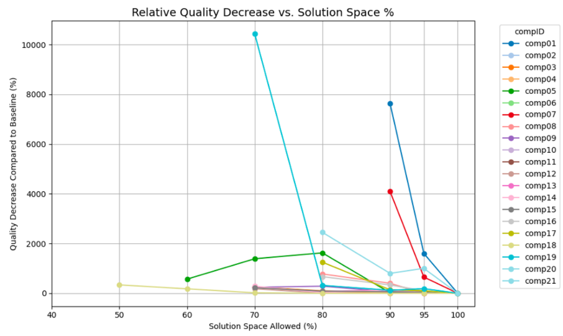
Figure 5.1: Fraction of instances with feasible solution as a function of solution space for each disruption type.

As the allowed solution space decreases, feasibility remains high for most disruption types down to the 90% level, with all instances recovering successfully for *Insert Curriculum*, *Invalid Assignment* and over 90% for *Removed Room* disruptions. However, once the solution space is reduced beyond 90%, a sharp decline in feasibility is observed across all disruption types. The most severe drop occurs for *Removed Period* disruptions, which are already sensitive to previous flexibility constraints and show limited robustness even at higher space levels. These patterns indicate that most disruptions can be effectively managed under modest flexibility reductions, but robustness deteriorates rapidly when the solution space becomes too constrained.

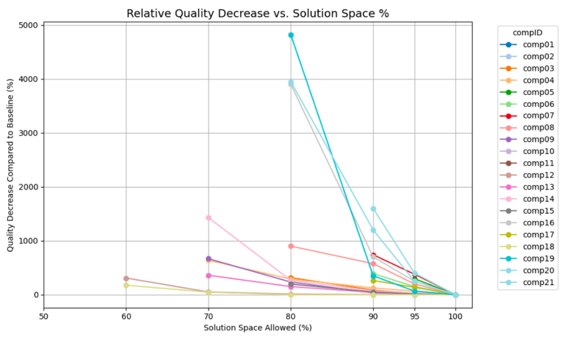
Each plot in Figure 5.2 shows the evolution of recovered solution quality for each instance as a function of available solution space, separately for each disruption type. The x-axis represents the percentage of available solution space, while the y-axis displays the relative decrease in the quality (= increase in objective function value) compared to the baseline solution without space restriction. The curves thus illustrate how solution quality progressively degrades as scheduling flexibility diminishes.



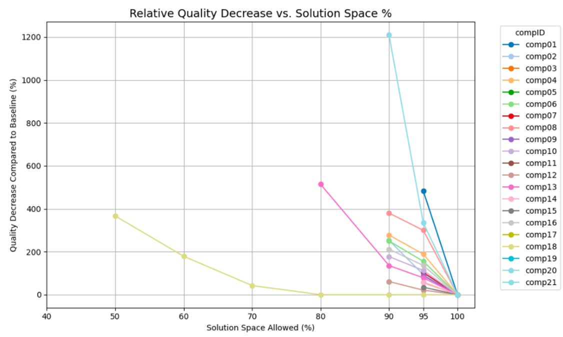
(a) Invalid Assignment



(b) Insert Curriculum



(c) Removed Room



(d) Removed Period

Figure 5.2: Relative quality degradation as a function of solution space for each disruption type.

For each disruption type, the relative objective value compared to the baseline progressively worsens as fewer rescheduling options are available, conditional on a feasible solution being found. The plots show that solution quality remains relatively stable when at least 95% of the original solution space is retained, making this level a natural reference point for comparison. Moreover, at this 95% threshold, all instances still recover a feasible solution for *Invalid Assignment* (Table 5.16), *Insert Curriculum* (Table 5.17) and *Removed Room* (Table 5.18). Even for *Removed Period* (Table 5.19), where feasibility is most difficult to regain, 17 instances still recover. This makes 95% the most balanced and representative point for cross-disruption analysis. While infeasibility introduces selection bias, since average values are only computed over recovered instances, this effect is minimal at 95%. *Insert Curriculum* shows the steepest increase in recovered solution cost at this level, with an average quality decrease Q_s of 224.01% (Table 5.17), highlighting severe quality degradation. *Removed Period* follows with an average decrease of 140.36% (Table 5.19), while *Removed Room* exhibits a similar impact (126.53%) (Table 5.18). In contrast, *Invalid Assignment* shows the smallest decrease (127.03%) (Table 5.16), indicating that this disruption type tends to cause the least degradation in solution quality.

Below 80% solution space, the number of recovered instances drops sharply across all disruptions, particularly for *Removed Period*, where only 3 out of 21 instances recover at 80% and just 1 at 70%, making those averages highly sensitive to individual outliers and less suitable for robust interpretation. Even at 95%, the results for *Removed Period* remain uncertain due to the nonrecovery of four specific instances: `comp05`, `comp17`, `comp19` and `comp21`. Their absence introduces bias, especially considering that these instances exhibit some of the highest solution costs under other disruptions. For example, `comp05` shows an average quality degradation of 566.30% under *Insert Curriculum*, 297.20% under *Removed Room* and 220.50% under *Invalid Assignment*. This suggests that if recovery had been achieved under *Removed Period*, the corresponding averages would have been substantially higher.

Table 5.16: Average quality decrease and recovery counts for Invalid Assignment.

Solution Space s (%)	Q_s (%)	Recovered Instances
40	0.00	0
50	269.70	1
60	136.36	1
70	368.70	5
80	895.86	14
90	296.19	21
95	127.03	21
100	0.00	21

Table 5.17: Average quality decrease and recovery counts for Insert Curriculum.

Solution Space s (%)	Q_s (%)	Recovered Instances
40	0.00	0
50	336.40	1
60	371.05	2
70	1348.68	10
80	549.68	16
90	843.31	21
95	224.01	21
100	0.00	21

Table 5.18: Average quality decrease and recovery counts for Removed Room.

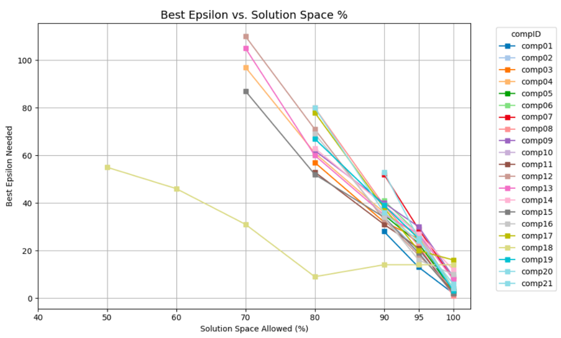
Solution Space s (%)	Q_s (%)	Recovered Instances
50	0.00	0
60	242.30	2
70	531.30	6
80	1251.99	14
90	388.96	19
95	126.53	21
100	0.00	21

Table 5.19: Average quality decrease and recovery counts for Removed Period.

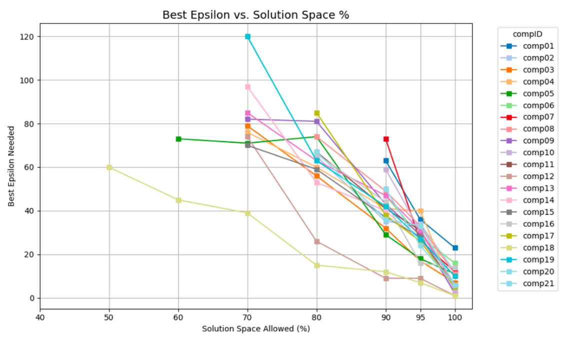
Solution Space s (%)	Q_s (%)	Recovered Instances
40	0.00	0
50	366.70	1
60	178.80	1
70	42.40	1
80	257.15	3
90	295.72	11
95	140.36	17
100	0.00	21

Perturbation effort for recovery

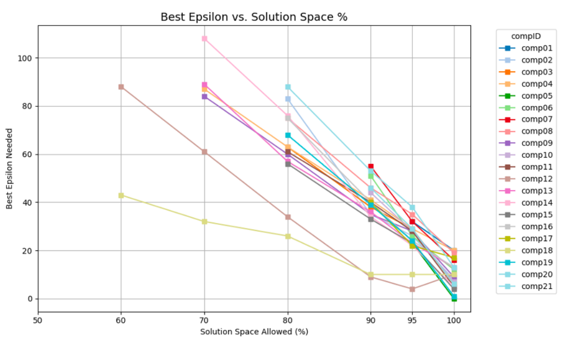
Figure 5.3 shows the perturbation effort, measured as the epsilon corresponding to the best-quality solution, as a function of solution space. Each plot displays the required perturbation effort for each instance across varying levels of solution space availability, separately for each disruption type. The x-axis represents the percentage of available solution space, while the y-axis shows the number of changes (epsilon) needed to achieve the best-quality solution found.



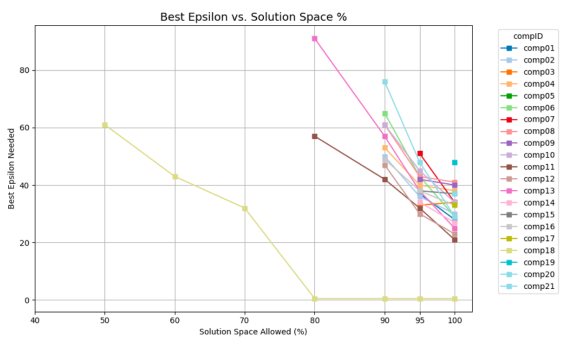
(a) Invalid Assignment



(b) Insert Curriculum



(c) Removed Room



(d) Removed Period

Figure 5.3: Best epsilon needed to recover the best quality solution as a function of solution space for each disruption type.

The perturbation effort increases substantially as scheduling flexibility decreases, with notable differences within disruption types. While some instances maintain low epsilon values even under restricted conditions, others require significantly higher perturbation budgets to achieve feasibility. To ensure consistency with the preceding analysis of quality degradation, we again focus on the 95% solution space level as the most balanced comparison point. As shown in Table 5.23, *Removed Period* requires the highest average perturbation effort E_s at this level (36.97), with four instances failing to recover as previously noted. Despite their absence, *Removed Period* still has the highest average, further reinforcing its status as the most disruptive scenario. For the other disruption types, the required effort remains within the same order of magnitude, *Invalid Assignment* (22.76) (Table 5.20), *Insert Curriculum* (26.19) (Table 5.21) and *Removed Room* (25.48) (Table 5.22), suggesting that while recovery becomes costlier as flexibility decreases, it remains manageable across these scenarios.

Table 5.20: Average perturbation effort and recovery counts for Invalid Assignment.

Solution Space s (%)	E_s	Recovered Instances
40	0.00	0
50	55.00	1
60	46.00	1
70	86.00	5
80	61.57	14
90	36.29	21
95	22.76	21
100	5.71	21

Table 5.21: Average perturbation effort and recovery counts for Insert Curriculum.

Solution Space s (%)	E_s	Recovered Instances
40	0.00	0
50	60.00	1
60	59.00	2
70	79.30	10
80	60.88	16
90	41.29	21
95	26.19	21
100	8.05	21

Table 5.22: Average perturbation effort and recovery counts for Removed Room.

Solution Space s (%)	E_s	Recovered Instances
50	0.00	0
60	65.50	2
70	76.83	6
80	63.21	14
90	38.16	19
95	25.48	21
100	10.19	21

Table 5.23: Average perturbation effort and recovery counts for Removed Period.

Solution Space s (%)	E_s	Recovered Instances
40	0.00	0
50	61.00	1
60	43.00	1
70	32.00	1
80	49.50	3
90	51.05	11
95	36.97	17
100	31.08	21

Influence of load factor and solution space on recovery

Finally, The interaction between instance-specific load characteristics and solution space flexibility is analysed through logistic regression. The model estimates the probability of successfully recovering a feasible solution based after disruption on (i) solution space availability s and (ii) instance-specific load factor λ_i. The model is defined as:

$$\log\left(\frac{\Pr(\text{Feasible}_{i,s})}{1-\Pr(\text{Feasible}_{i,s})}\right) = \beta_0 + \beta_1 \cdot s + \beta_2 \cdot \lambda_i$$

where $\beta_0, \beta_1, \beta_2$ are the estimated coefficients. The outcome variable predicts whether a feasible recovery will be found at solution space level s for instance i and the model quantifies how feasibility is influenced by flexibility and timetable density. The logistic regression model takes the form:

$$\Pr(\text{Feasible}_{i,s}) = \frac{1}{1+\exp(-(\beta_0 + \beta_1 \cdot s + \beta_2 \cdot \lambda_i))}$$

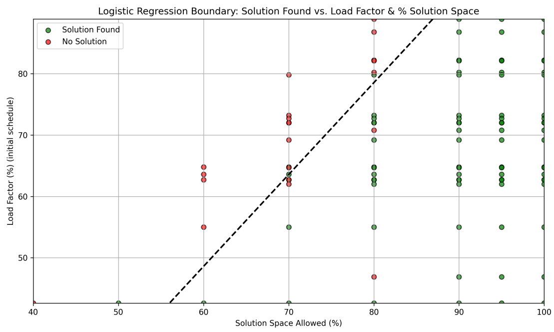
where $\beta_0, \beta_1, \beta_2$ are the estimated model coefficients, s denotes the percentage of allowed solution space and λ_i is the load factor. Each subplot in Figure 5.4 visualises individual recovery attempts as scatter points, where each dot corresponds to a recovery run for an instance at a specific combination of solution space availability and instance-specific load factor. The dotted line in each plot reflects the estimated decision boundary, defined as the point where the predicted probability of recovery reaches 50%. Points to the right of this line represent scenarios with a predicted success probability greater than 50%, while those to the left fall below this threshold. This boundary is defined by the equation:

$$\beta_0 + \beta_1 \cdot s + \beta_2 \cdot \lambda_i = 0$$

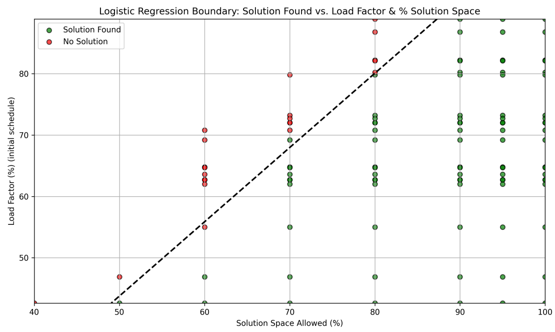
which can be rearranged as:

$$\lambda_i = -\frac{\beta_1}{\beta_2} \cdot s - \frac{\beta_0}{\beta_2}.$$

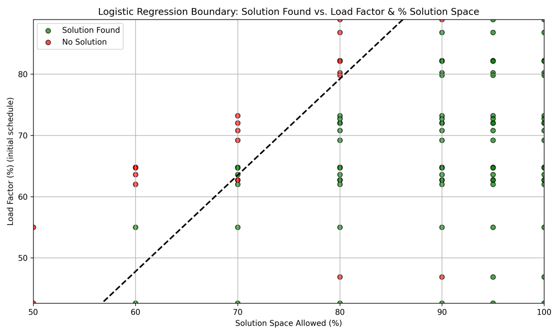
The slope of the decision boundary is therefore determined by the ratio $-\frac{\beta_1}{\beta_2}$. A steep or nearly vertical boundary indicates that solution space availability is the dominant predictor of recovery feasibility, whereas a flatter or nearly horizontal boundary suggests a stronger role for load factor. A diagonal boundary reflects that both factors contribute meaningfully to the prediction. While the regression model operates in terms of log-odds, its true strength lies in its ability to provide clear, interpretable predictions. By inputting only two values, the instance's load factor and the percentage of available solution space, the model can estimate the likelihood of successful recovery for a given disruption scenario. This makes it possible to anticipate recovery feasibility in advance without solving the full optimisation problem. As such, the logistic model serves not only as an analytical tool but also as a practical decision-support mechanism for timetable planners.



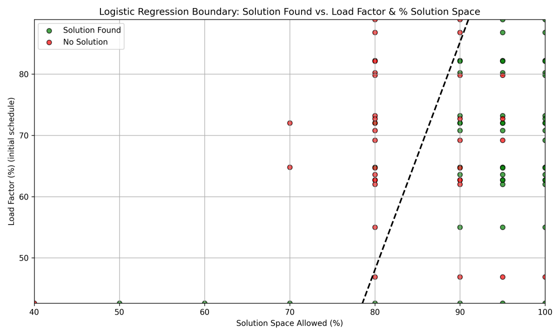
(a) Invalid Assignment



(b) Insert Curriculum



(c) Removed Room



(d) Removed Period

Figure 5.4: Logistic regression boundary between solution found and no solution found, as a function of solution space and load factor for each disruption type.

The numerical results of the logistic regression models are summarised in Figure 5.5. These figures report the estimated coefficients for solution space availability (β_1) and load factor (β_2), which quantify the relationship between each input variable and the model's predicted probability of recovery. To evaluate predictive performance, two complementary metrics are reported: accuracy and ROC AUC.

Accuracy measures the proportion of correctly classified instances and provides a straightforward indication of overall model performance:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},$$

where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, respectively. In this study, a ”positive” instance refers to a recovery attempt that successfully restored feasibility, while a ”negative” instance corresponds to a failed recovery. Accuracy thus reflects how often the model correctly predicts whether recovery will succeed or fail. While intuitive, it can be sensitive to class imbalance and does not consider prediction confidence.

To address this, the ROC AUC (Receiver Operating Characteristic Area Under Curve) provides a threshold-independent measure of the model's ability to distinguish between successful and unsuccessful recoveries. It is defined as the probability that a randomly selected successful recovery attempt (positive instance) receives a higher predicted score than a failed one (negative instance):

$$\text{ROC AUC} = \Pr\left(\hat{y}_1 > \hat{y}_0\right),$$

where $\hat{y}_1$ and $\hat{y}_0$ denote predicted probabilities for randomly selected positive and negative instances, respectively. In practice, the ROC AUC is estimated by comparing all such positive-negative pairs: a score of 1 is counted if the positive instance receives a higher score, 0.5 if scores are equal, and 0 otherwise. Values close to 1 indicate strong discriminatory power, while a ROC AUC of 0.5 corresponds to random guessing. Having introduced the model structure and evaluation metrics, we now turn to the interpretation of the estimated coefficients and predictive performance across disruption types, as stated in Figure 5.5.

```
🔍 Logistic Regression Evaluation
========================================
✅ Accuracy: 0.92

🧮 Confusion Matrix:
          Predicted 0  Predicted 1
Actual 0           17            4
Actual 1            4           80

📊 Classification Report:
              precision    recall  f1-score   support

           0      0.810     0.810     0.810        21
           1      0.952     0.952     0.952        84

    accuracy                          0.924       105
   macro avg      0.881     0.881     0.881       105
weighted avg      0.924     0.924     0.924       105

📈 ROC AUC: 0.95

📉 Model Coefficients:
  Solution Space (%): 0.2883
  Load Factor (%): -0.1926
  Intercept: -7.9390
```

(a) Invalid Assignment

```
🔍 Logistic Regression Evaluation
========================================
✅ Accuracy: 0.98

🧮 Confusion Matrix:
          Predicted 0  Predicted 1
Actual 0           21            1
Actual 1            1           91

📊 Classification Report:
              precision    recall  f1-score   support

           0      0.955     0.955     0.955        22
           1      0.989     0.989     0.989        92

    accuracy                          0.982       114
   macro avg      0.972     0.972     0.972       114
weighted avg      0.982     0.982     0.982       114

📈 ROC AUC: 1.00

📉 Model Coefficients:
  Solution Space (%): 0.9756
  Load Factor (%): -0.8062
  Intercept: -13.4984
```

(b) Insert Curriculum

```
🔍 Logistic Regression Evaluation
========================================
✅ Accuracy: 0.91

🧮 Confusion Matrix:
          Predicted 0  Predicted 1
Actual 0           16            5
Actual 1            4           79

📊 Classification Report:
              precision    recall  f1-score   support

           0      0.800     0.762     0.780        21
           1      0.940     0.952     0.946        83

    accuracy                          0.913       104
   macro avg      0.870     0.857     0.863       104
weighted avg      0.912     0.913     0.913       104

📈 ROC AUC: 0.91

📉 Model Coefficients:
  Solution Space (%): 0.2090
  Load Factor (%): -0.1330
  Intercept: -6.1887
```

(c) Removed Room

```
🔍 Logistic Regression Evaluation
========================================
✅ Accuracy: 0.79

🧮 Confusion Matrix:
          Predicted 0  Predicted 1
Actual 0           22           14
Actual 1            5           49

📊 Classification Report:
              precision    recall  f1-score   support

           0      0.815     0.611     0.698        36
           1      0.778     0.907     0.838        54

    accuracy                          0.789        90
   macro avg      0.796     0.759     0.768        90
weighted avg      0.793     0.789     0.782        90

📈 ROC AUC: 0.81

📉 Model Coefficients:
  Solution Space (%): 0.1167
  Load Factor (%): -0.0314
  Intercept: -7.8258
```

(d) Removed Period

Figure 5.5: Logistic regression evaluation results for recovery prediction under different disruption types.

The effect of load factor, captured by coefficient β_2, is negative across all disruption types, indicating that higher timetable density reduces the predicted likelihood of successful recovery. The strongest impact is observed for Insert Curriculum ($\beta_2 = -0.8062$), followed by Invalid Assignment ($\beta_2 = -0.1926$) and Removed Room ($\beta_2 = -0.1330$), while the effect is minimal for Removed Period ($\beta_2 = -0.0314$).

Solution space availability, captured by coefficient β_1, emerges as the dominant predictor overall. All disruptions show a positive value, confirming that greater flexibility improves recovery feasibility. The effect is strongest for Insert Curriculum ($\beta_1 = 0.9756$), followed by Invalid Assignment ($\beta_1 = 0.2883$) and Removed Room ($\beta_1 = 0.2090$), while the impact is smaller but still positive for Removed Period ($\beta_1 = 0.1167$).

This interpretation is visually supported by the regression boundaries shown in Figure 5.4. For Insert Curriculum, Invalid Assignment and Removed Room, the decision boundaries show a clear diagonal orientation, reflecting the joint influence of both solution space and load factor on recovery feasibility. In contrast, the boundary for Removed Period is nearly vertical, indicating that the outcome in this scenario depends almost entirely on solution space availability, with load factor contributing minimally. This is consistent with the small magnitude of β_2 in this setting (Figure 5.5d).

The models achieve near-perfect predictive performance for Insert Curriculum, Invalid Assignment and Removed Room, with ROC AUC values of 1.00, 0.95, 0.91 and similarly high accuracy scores. In contrast, Removed Period shows weaker but still acceptable performance, with an accuracy of 0.79 and ROC AUC of 0.81. These results suggest that, although some disruptions are harder to predict due to overlapping recovery outcomes, the logistic regression model remains robust and interpretable across all cases.

Beyond statistical evaluation, the logistic regression models also offers practical value by allowing direct prediction of recovery outcomes for new scenarios. Given only a load factor and a solution space percentage, the model estimates the probability of recovering a feasible solution for a given disruption. This enables planners to assess whether additional flexibility is needed without rerunning complex optimisation models.

For example, in the case of *Insert Curriculum*, the estimated model coefficients are $\beta_0 = -13.4984$, $\beta_1 = 0.9756$ and $\beta_2 = -0.8062$ (Figure 5.5b). A scenario with 80% solution space and a load factor of 80% (entered as 80 and not 0.8) results in the following logit:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \cdot 80 + \beta_2 \cdot 80 = -13.4984 + 78.048 - 64.496 = 0.0536$$

corresponding to a predicted recovery probability of:

$$p = \frac{1}{1+e^{-0.0536}} \approx 51.3\%.$$

This places the point almost exactly on the decision boundary shown in Figure 5.4b, confirming the interpretation of the dashed line as the 50% threshold where feasibility becomes uncertain.

Two contrasting scenarios highlight how sensitive the model is to changes in load factor. If the load factor is reduced to 77%, with solution space still at 80%, the logit becomes:

$$\log\left(\frac{p}{1-p}\right) = -13.4984 + 78.048 - 62.8774 = 1.6722$$

yielding a much higher predicted probability of finding a feasible solution:

$$p = \frac{1}{1+e^{-1.6722}} \approx 84.2\%.$$

Conversely, increasing the load factor to 83% shifts the logit to:

$$\log\left(\frac{p}{1-p}\right) = -13.4984 + 78.048 - 66.9146 = -2.3650$$

resulting in a significantly lower recovery probability of finding a feasible solution:

$$p = \frac{1}{1 + e^{2.3650}} \approx 8.6\%.$$

Together, these examples demonstrate how the model offers actionable insight and can be used to anticipate whether a particular combination of load and flexibility is likely to allow recovery under a given disruption type.

5.5.3 Summary and key observations

Table 5.24 summarises the recovery performance across the different disruption types. To enable consistent comparison, we evaluate the performance indicators introduced in Section 5.5.1 using fixed benchmark values. Specifically, R_{90} denotes the percentage of instances that recover a feasible solution at 90% solution space. E_{95} captures the average perturbation effort at 95% flexibility and Q_{95} captures the quality degradation observed at 95% flexibility compared to the full solution space baseline.

This distinction in benchmarks is motivated by differences in recovery feasibility and solution richness. At 90% solution space, *Invalid Assignment*, *Insert Curriculum* recover feasibility across all 21 instances, *Removed room* in 19 out of 21 instances, whereas *Removed Period* drops to just 11 feasible instances (see Tables 5.16–5.19). At 90%, the impact of reduced flexibility on feasibility becomes evident, making it a meaningful benchmark for comparing disruption types through R_{90}.

By contrast, average perturbation effort and quality degradation (E_{95}, Q_{95}) are best assessed at 95% flexibility, where a sufficient number of instances still recover across all disruption types (Tables 5.16–5.19). Using this level ensures that comparisons are based on representative samples, while still reflecting non-trivial recovery effort. Anchoring these metrics at 95% thus balances completeness and interpretability.

Table 5.24: Comparison of recovery performance for solution space reductions across different disruption types.

Disruption	R_{90}	E_{95}	$\%Q_{95}$
Invalid Assignment	100.0%	22.76	127.03%
Insert Curriculum	100.0%	26.19	224.01%
Removed Room	90.5%	25.48	126.53%
Removed Period	52.4%	36.97	140.36%

Invalid Assignment and Insert Curriculum disruptions demonstrate high robustness, with 100% of instances recovering under a 95% solution space, whereas recovery rates drop slightly for Removed Room disruptions (90.5%) and are the lowest for Removed Period disruptions (52.4%). Similarly, the average perturbation effort under 95% solution space is considerably lower for Invalid Assignment (22.76) compared to Insert Curriculum (26.19), Removed Room (25.48) and Removed Period (36.97). Both *Insert Curriculum* and *Removed Period* exhibit the most severe quality degradation at 95% flexibility, highlighting their disruptive nature. In the case of *Removed Period*, the reported average of 140.36% likely underestimates the true cost as four challenging instances were not included in the calculation.

Beyond these comparative metrics, additional insight is gained from the logistic regression model, which quantifies how solution space and instance-specific load factor jointly influence recovery feasibility. Across all disruption types, solution space emerged as the dominant predictor. Load factor also played a significant role, particularly for *Insert Curriculum* and *Invalid Assignment*, where higher timetable density reduced the likelihood of feasible recovery. In contrast, the effect of load factor was minimal for *Removed Period*. Importantly, the model allows planners to estimate recovery probabilities in advance, based solely on an instance's load factor and available flexibility without the need to solve the full optimisation model. This predictive capacity makes it a valuable decision-support tool for robust timetable design.

In sum, the analysis confirms that both solution space flexibility and timetable load factor meaningfully influence recovery outcomes. While solution space remains the primary driver, instance-specific load characteristics further shape feasibility under constrained conditions, highlighting the multifaceted nature of timetable robustness.

Chapter 6

Conclusion

The goal of this thesis was to assess how well university timetables can be recovered after disruptions, particularly when recovery options are limited. To address this objective, a bi-objective recovery framework was developed based on the curriculum-based course timetabling model by Di Gaspero et al. (2007), which defines a mixed-integer formulation for 21 benchmark instances from the ITC-2007 dataset. This model was extended by Lindahl et al. (2019) into a bi-objective formulation that jointly minimises solution quality loss and perturbation effort using the epsilon-constraint method. Building further on both, this thesis makes a contribution by introducing additional recovery-specific constraints to simulate more realistic and restricted recovery conditions. In doing so, it moves beyond the original full-flexibility setting and systematically explores how recovery performance changes under three distinct and practically motivated flexibility limitations: limited room reassignments, restricted period adjustments and solution space reductions. These extensions provide new insight into the robustness of timetables under disruption and contribute a more grounded perspective at the interface of tactical planning and operational execution to the recovery literature.

The results show that recovery performance depends strongly on both the available flexibility and the nature of the disruption. Under full perturbation flexibility, feasibility and quality were consistently restored across disruption types. However, the type of disruption plays a critical role in shaping recovery outcomes. Disruptions such as *Invalid Assignment* and *Insert Curriculum* were generally easier to recover from, while *Removed Room* and especially *Removed Period* posed greater challenges. When recovery was limited to either room or period adjustments, recovery effectiveness depended on the match between disruption type and recovery mechanism: room-only recovery was more successful for spatial disruptions such as *Invalid Assignment*, whereas period-only recovery was better suited to temporal disruptions like *Removed Period* and *Insert Curriculum*. In the restricted solution space setting, both feasibility and quality degraded rapidly as flexibility decreased, with *Removed Period* emerging as the most sensitive to flexibility loss.

Beyond the optimisation-based analysis, this thesis also introduces a logistic regression model that predicts recovery feasibility based on instance load and available solution space. This model enhances interpretability and provides planners with a forward-looking decision-support tool, allowing them to anticipate recovery success without rerunning complex optimisation routines.

Taken together, these findings emphasise the importance of embedding robustness into timetable design and support the use of flexible, disruption-aware recovery mechanisms in practice. By systematically evaluating constrained recovery strategies and providing predictive insight through logistic modelling, this thesis offers both theoretical and practical contributions toward more resilient academic scheduling.

6.1 Future work

Future research could build on this work in several directions. First, future work could build on the recovery patterns identified in this study to design a *hyperheuristic rescheduling mechanism* that dynamically selects the most appropriate recovery strategy based on the nature of the disruption. For example, period-only recovery may be preferred for temporal disruptions such as *Removed Period*, while room-only recovery could be more suitable for spatial disruptions like *Invalid Assignment*. Such a rule-based mechanism would enable more targeted and efficient responses in operational settings.

Second, the framework could be enhanced with *learning-based methods* that detect patterns in instance features and predict the most effective recovery strategy accordingly in case of a disruption. This data-driven approach would allow the system to generalise across unseen instances and fine-tune strategy selection over time, improving adaptability and robustness in complex timetabling environments.

Third, future research could explore additional disruption types that go beyond the current scope, such as multi-day unavailability or constraints involving instructors or curricula. Expanding the disruption taxonomy would further broaden the applicability of the recovery framework and enhance its relevance to real-world academic scheduling.

Bibliography

Badoni, R. P. and Gupta, D. (2015). A hybrid algorithm for university course timetabling problem. *Innovative Systems Design and Engineering*, 6(2):60–66.

Di Gaspero, L., McCollum, B., and Schaerf, A. (2007). Curriculum-based course timetabling: Specification and benchmark. *Proceedings of the International Timetabling Competition*.

El Sakkout, H., Richards, T., and Wallace, M. (1998). Minimal perturbation in dynamic scheduling. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI)*, pages 504–508.

Geiger, M. J. (2010). An application of threshold accepting to curriculum based course timetabling. In *Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling (PATAT)*, pages 227–240.

Haimes, Y. Y., Lasdon, L. S., and Wismer, D. A. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimisation. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297.

Hao, J.-K. and Benlic, U. (2011). Lower bounds for the itc-2007 curriculum-based course timetabling problem. *European Journal of Operational Research*, 212(3):464–472.

Lach, G. and Lübbecke, M. E. (2012). Curriculum-based course timetabling: new solutions to udine benchmark instances. *Annals of Operations Research*, 194(1):255–272.

Lindahl, M., Mason, A., Stidsen, T. R., and Sørensen, M. (2018a). A strategic view of university timetabling. *European Journal of Operational Research*, 266(1):35–45.

Lindahl, M., Stidsen, T. R., and Sørensen, M. (2018b). A fix-and-optimise matheuristic for university timetabling. *Journal of Heuristics*, 24(5):645–665.

Lindahl, M., Stidsen, T. R., and Sørensen, M. (2019). Quality recovering of university timetables. *European Journal of Operational Research*, 276(2):422–435.

Maenhout, B. and Vanhoucke, M. (2011). An evolutionary approach for the integrated nurse staffing and scheduling problem. *OR Spectrum*, 33(2):441–466.

Müller, T., Rudová, H., and Murray, K. (2005). Timetabling with flexibility. In *Practice and Theory of Automated Timetabling V*, pages 131–146. Springer.

Phillips, M., Burke, E. K., and McCollum, B. (2017). An integer programming approach to the minimal perturbation problem in course timetabling. *Annals of Operations Research*, 252(2):283–304.

Zivan, R., Meisels, A., and Razgon, I. (2011). A hybrid framework for dynamic constraint satisfaction problems. *Journal of Artificial Intelligence Research*, 40:1–45.