JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Evaluating Disruption Recovery across Pareto-Representative Solutions in Multi-Objective University Course Timetabling

Hanifah Salsabila Ryadi *, Ristu Saptono *, and Muhammad Fahmy Nadhif

Faculty of Information Technology and Data Science, Sebelas Maret University, Surakarta 57126, Indonesia;
e-mail : ryadishanifah@student.uns.ac.id; ristu.saptono@staff.uns.ac.id; fahmynadhif@staff.uns.ac.id
* Corresponding Author : Hanifah Salsabila Ryadi ; Ristu Saptono

Abstract: Multi-objective university course timetabling produces a Pareto front of trade-off solutions, but ultimately, one timetable must be selected for deployment. Existing studies commonly base this choice on objective values, robustness criteria, or stakeholder preferences, while the downstream recovery implications of selecting different Pareto representatives remain insufficiently understood. This study proposes a deployment-oriented evaluation framework to examine whether the selected Pareto-representative timetable affects recovery after calendar-based disruptions. The framework keeps the original optimization objectives fixed, selects the best-f_1, best-f_2, and compromise timetables from each run, projects them onto an academic calendar with actual dates and holidays, and evaluates them under five disruption scenarios using greedy direct rescheduling followed by target-meeting fulfillment repair. The framework is evaluated using institutional scheduling and calendar data from the odd and even semesters of the 2025/2026 academic year, which provide a natural contrast in room-slot occupancy and disruption exposure. Weekly timetables are generated using NSGA-II by minimizing a weighted soft-constraint penalty (f_1) and average room-capacity waste (f_2), with the experiments repeated across five independent random seeds. No representative timetable produced a consistent recovery advantage under the tested setting: differences in the direct rescheduling success rate were small relative to seed-to-seed variation, even when the representative timetables differed substantially in their class placements. The larger contrast occurred between semesters. The denser odd semester, with approximately 91% room-slot occupancy, directly recovered about one-third of affected meetings, whereas the less dense even semester, with approximately 52% occupancy, directly recovered nearly all affected meetings. Both semesters achieved 100% target-meeting fulfillment after repair. Overall, the stronger observed contrast occurred between semester settings and was more consistent with differences in structural slack than with the selected representative category.

Keywords: Academic calendar; Disruption recovery; Multi-objective optimization; NSGA-II; Pareto-representative selection; Reactive rescheduling; Structural slack; University course timetabling.

Received: July, 3rd 2026
Revised: July, 17th 2026
Accepted: July, 18th 2026
Published: August, 5th 2026

1. Introduction

University course timetabling assigns course sessions to rooms and time slots while satisfying institutional requirements involving lecturers, student groups, room capacities, and teaching patterns. It is a widely studied problem in educational operations research [1]–[3] and is computationally challenging because the number of possible assignments grows rapidly with the numbers of classes, rooms, and time slots [1], [2]. Indonesian university case studies illustrate both the persistence of manual scheduling problems and the application of algorithmic approaches to reduce scheduling conflicts [4], [5]. Broader reviews nevertheless identify a continuing gap between timetabling research and institutional practice [6].

University timetabling has been addressed using mathematical programming, constraint-based methods, metaheuristics, and hybrid approaches [2], [7]. Genetic algorithms are widely adopted because their population-based search can effectively explore alternative timetable structures and can be combined with guided or local search strategies [8], [9]. When timetable quality involves competing objectives, multi-objective formulations generate a set of non-

dominated solutions rather than a single aggregated solution. Such formulations have been applied to examination timetabling [10]–[12], while NSGA-II provides a well-established mechanism for maintaining convergence and diversity along the resulting Pareto front [13]. Recent studies have further extended university timetabling to multi-level, hybrid-teaching, and heterogeneous real-world institutional settings [14]–[16].

Producing a high-quality initial timetable, however, does not guarantee that it will remain operational throughout the academic term. Holidays, lecturer unavailability, room closures, and unavailable teaching periods may invalidate scheduled meetings after deployment. Existing research addresses this problem mainly through two strategies. Proactive robustness approaches incorporate disruption resistance into the initial optimization model [17]–[20]. In contrast, post-disruption recovery approaches revise an existing baseline timetable using integer or mixed-integer programming, MaxSAT, iterative forward search, and perturbation heuristics [21]–[25], including institutional minimal-perturbation models [26].

These approaches leave an important deployment question insufficiently explored. A multi-objective optimizer produces several Pareto-optimal alternatives, yet an institution must ultimately select a single timetable for operational use. Representative selection is commonly based on objective values, compromise criteria, or stakeholder preferences. However, it remains unclear whether this choice also influences the timetable's recoverability when both the optimization objectives and the recovery mechanism are held fixed. Robustness-oriented studies cannot isolate this effect because they modify the optimization formulation, whereas minimal-perturbation studies generally treat the deployed timetable as a predetermined baseline.

This study therefore addresses the following research question: Does the choice among Pareto-representative weekly timetables affect disruption-recovery outcomes when all representatives are evaluated under the same calendar-based disruptions and recovery procedure? To answer this question, a deployment-oriented evaluation framework comprising two stages is proposed. First, NSGA-II generates weekly timetables by minimizing a weighted soft-constraint penalty (f_1) and average room-capacity waste (f_2). The best-f_1, best-f_2, and compromise timetables are then selected independently from the Pareto front of each run. Second, the selected representatives are projected onto an academic calendar with actual dates and holidays, exposed to five disruption scenarios, and processed using greedy direct rescheduling followed by target-meeting fulfillment repair. The framework is evaluated using institutional scheduling and calendar data from the Faculty of Information Technology and Data Science, Sebelas Maret University, augmented with scenario-based lecturer preference and restriction profiles. The complete evaluation is repeated across five independent random seeds for each semester.

The main contributions of this study are as follows:

- A deployment-oriented evaluation framework that treats Pareto-representative timetable selection as the primary object of evaluation by holding the optimization objectives fixed and assessing recoverability after deployment.
- A matched calendar-based evaluation protocol that projects representative timetables onto actual academic calendars, evaluates them under comparable disruption scenarios, and applies an identical recovery procedure across five independent optimization seeds.
- A comprehensive evaluation methodology that links design-time optimization characteristics with post-deployment recovery performance, enabling the operational consequences of Pareto-representative selection to be examined separately from differences in the initial Pareto fronts and semester settings.

Overall, this study shifts the focus from generating Pareto-optimal timetables to understanding their operational behavior after deployment, providing a practical evaluation perspective that complements conventional optimization-oriented timetabling research. The remainder of this paper is organized as follows. Section 2 reviews related work and positions the deployment-oriented research gap. Section 3 describes the proposed framework and evaluation methodology. Section 4 presents and discusses the experimental results, and Section 5 concludes the paper.

2. Literature Review

2.1 Multi-Objective University Course Timetabling

Multi-objective formulations are relevant because timetable quality usually reflects competing institutional or stakeholder interests. Examination timetabling studies have considered trade-offs involving proximity penalties, timetable length, and room-capacity-related criteria [10]–[12]. NSGA-II supports this setting by producing a set of non-dominated alternatives instead of requiring the objectives to be combined into a single weighted value [13].

Recent work further illustrates the diversity of institutional criteria and planning structures. Dunke and Nickel combine mathematical programming and genetic search across multiple planning levels to balance lecture, tutorial, and individual timetable quality [14]. Davison et al. formulate a multi-objective model that incorporates hybrid teaching and supports strategic university decision making [15]. Müller et al. describe heterogeneous real-world university timetabling instances from ITC-2019 [16].

Room allocation is also an important component of timetable quality. Lemos et al. apply greedy and exact approaches to optimize room occupation using real university data [27], while broader work shows that university teaching space may remain underutilized in practice [28]. These studies provide the optimization basis for the soft-constraint and room-capacity objectives used in the present work. However, their primary focus remains the construction or evaluation of the initial timetable rather than the downstream recovery consequences of selecting different Pareto representatives.

2.2 Robustness and Disruption Recovery in Timetabling

Research on timetable disruptions generally follows proactive robustness or post-disruption recovery strategies. Proactive approaches incorporate robustness into the initial optimization model. Akkan and Gülcü [18] formulate soft-constraint quality and robustness as competing criteria, while Gülcü and Akkan [17] evaluate robust university timetables under single and multiple disruptions. Later work approximates robustness using slack-based surrogate measures [19]. Robustness has also been considered in examination timetabling under uncertainty in student registration and demand [20].

Post-disruption approaches instead revise an existing timetable after infeasibility occurs. Phillips et al. [21] use integer programming to resolve disruptions within a limited neighborhood of the original timetable. Lindahl et al. [22] formulate recovery as a trade-off between timetable quality and the number of changes. Other approaches use MaxSAT [23], iterative forward search [24], minimum-penalty perturbation heuristics [25], and institutional minimal-perturbation models [26].

These studies establish two principal approaches to disruption handling: incorporating robustness into the initial model and recovering a fixed baseline timetable after disruption. The deployment decision that occurs between these stages is examined in the following subsection.

2.3 Deployment-Oriented Positioning of This Study

The dynamic-scheduling literature distinguishes proactive scheduling, predictive-reactive rescheduling, and local reactive repair [29]–[31]. Robust timetabling is closest to proactive scheduling because disruption resistance is introduced before deployment, whereas minimal-perturbation methods are predictive-reactive because they revise a baseline timetable after a disruption becomes known.

The procedure used in the present study is positioned as scenario-based local schedule repair. It begins with a deployed, calendar-projected timetable, identifies meetings affected by a predefined disruption scenario, and relocates them without re-optimizing the complete timetable. The procedure is lighter than global minimal-perturbation optimization, but it is not presented as a continuously operating real-time scheduling system. Its purpose is to provide the same controlled recovery mechanism for every Pareto representative so that the effect of the initial deployment choice can be examined. Table 1 summarizes the differences between representative prior studies and the present work. Existing studies either generate multi-objective timetable alternatives without evaluating their downstream recovery, incorporate robustness into the initial objective formulation, or recover a disrupted timetable that has already been fixed as the baseline. The unresolved decision lies between optimization and

recovery: after a Pareto front has been generated, does the representative selected for deployment carry measurable consequences for later disruption handling?

Table 1. Deployment-oriented comparison of representative timetabling studies

Study	Evaluation focus / reported insight	Disruption-handling strategy	Multi-objective timetable alternatives	Pareto-representative choice evaluated
Cheong et al. [10]	Pareto trade-offs in examination timetabling	No disruption evaluation	✓	✗
Dunke and Nickel [14]	Multi-level and multi-criteria timetable quality	No post-deployment recovery	✓	✗
Gülcü and Akkan [17]	Quality–robustness trade-off	Proactive robustness optimization	✓	✗
Phillips et al. [21]	Limited deviation from a fixed baseline	Integer-programming minimal perturbation	✗	✗
Lindahl et al. [22]	Recovery quality versus number of changes	Multi-objective recovery optimization	✓	✗
Lemos et al. [26]	Institutional timetable disruption recovery	Minimal-perturbation recovery	✗	✗
This study	Downstream consequence of representative selection	Matched greedy repair and fulfillment	✓	✓

The present study addresses this gap by keeping f_1 and f_2 unchanged, selecting best-f_1, best-f_2, and compromise independently from each run, and evaluating them under the same calendar-projection procedure, comparable disruption events, and identical recovery logic. The contribution is therefore not a new robustness objective or a claim of globally optimal recovery. It is a deployment-oriented framework for isolating and evaluating the operational consequences of Pareto-representative selection.

3. Proposed Method

This study proposes a two-stage deployment-oriented framework for evaluating the disruption recovery of Pareto-representative university course timetables. In the first stage, NSGA-II generates multi-objective weekly timetables, from which the best-f_1, best-f_2, and compromise timetables are selected. In the second stage, each representative timetable is projected onto the academic calendar, exposed to matched disruption scenarios, and processed using greedy direct rescheduling followed by target-meeting fulfillment repair. The overall workflow is illustrated in Figure 1.

3.1 Institutional Dataset and Preprocessing

3.1.1 Dataset Characteristics and Suitability

This study uses faculty-level academic records from the Faculty of Information Technology and Data Science, Sebelas Maret University, covering the odd and even semesters of the 2025/2026 academic year. The scheduling unit is an active class instance rather than a course title because the same course may be offered through multiple classes with different student groups, quotas, lecturer assignments, and SKS values. SKS (Satuan Kredit Semester) denotes the Indonesian academic credit unit and determines the required number of weekly session units for each class.

The institutional records comprise class, lecturer–course, lecturer, room, room-travel, time-slot, academic-calendar, and holiday data. They provide class and course attributes, team-teaching assignments, target meeting counts, room capacities and types, buildings, inter-room travel times, regular slot definitions, semester dates, and holidays. Lecturer preferred-time, preferred-building, and restricted-time profiles are included as scenario-based experimental inputs rather than verified declarations of individual lecturer preferences.

Weekly timetable optimization uses 44 regular time slots per room. The recovery stage additionally provides 11 emergency evening slots reserved for target-meeting fulfillment

repair; these slots are not used during weekly optimization or direct rescheduling. Table 2 summarizes the resulting scheduling and calendar characteristics.

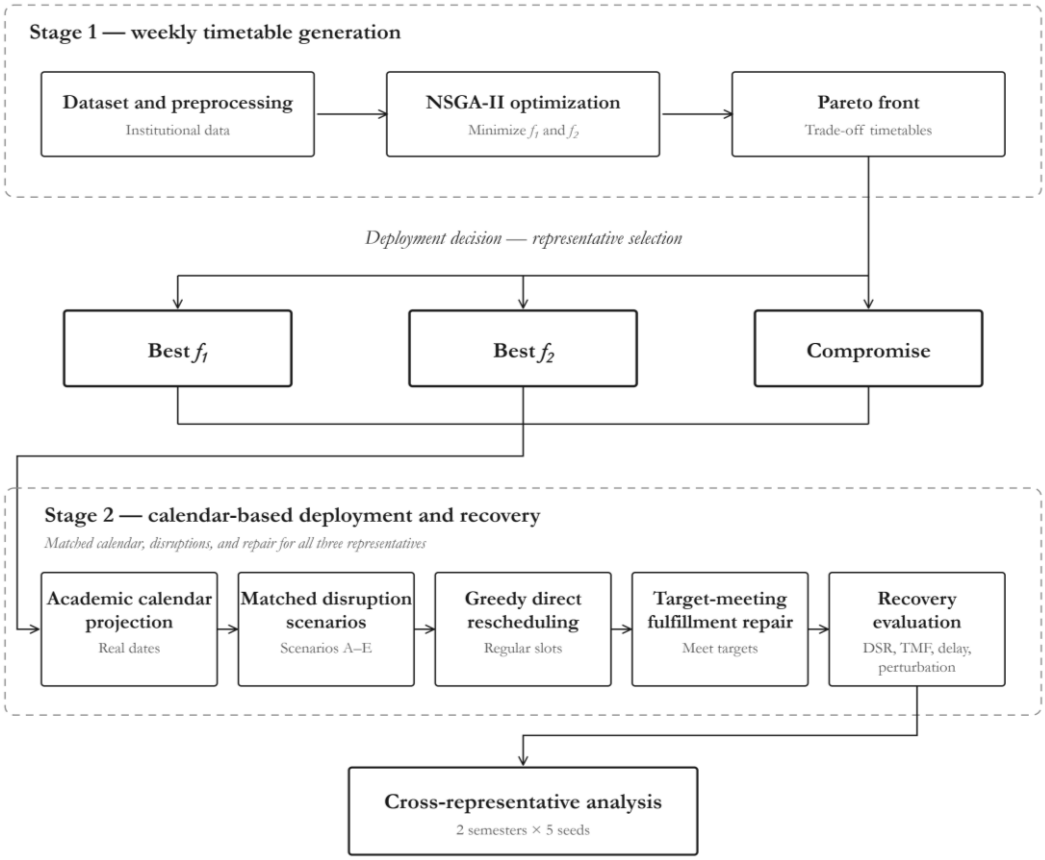

Figure 1. Two-stage deployment-oriented framework for Pareto-representative selection and calendar-based disruption recovery

Table 2. Institutional dataset and calendar characteristics used for framework evaluation

Dataset Characteristics	Odd Semester	Even Semester
Active Classes	188	125
Required Weekly Sessions (ΣSKS)	402	276
Lecturer Assignments	193	133
Active Lecturers	49	31
Rooms	10	12
Regular Weekly Time Slots per Room	44	44
Regular Room-Slot Capacity	440	528
Approximate regular room-slot occupancy *	91.4%	52.3%
Emergency Time Slots for Rescheduling	11	11
Academic-Calendar Period	25 Aug 2025–2 Jan 2026	23 Feb 2026–3 Jul 2026
Academic Weekdays	95	95
Total Holidays in The Calendar Period	3	17
Weekday Holidays Affecting Teaching	3	13
Active Academic Weekdays	92	82

* Approximate regular room-slot occupancy is calculated as the number of required weekly sessions divided by the product of the number of rooms and regular weekly time slots per room.

The two datasets provide a natural contrast in regular room-slot occupancy (91.4% versus 52.3%) and weekday-holiday exposure (3 versus 13), while retaining realistic institutional constraints. This contrast allows the same representative-selection and recovery procedure to

be evaluated under substantially different operational settings. These semester differences are interpreted as a natural institutional contrast rather than a controlled causal experiment. Room-slot occupancy is an aggregate scheduling-density proxy, not a direct measure of local recovery slack, which also depends on room compatibility, lecturer and student-group availability, session-block requirements, and the temporal distribution of vacant slots.

3.1.2 Data Preprocessing and Scheduling-Object Construction

Before optimization, identifiers used for cross-table joins are normalized and the input tables are sorted deterministically. Class records are restricted to active classes represented in the lecturer–course mapping, and all lecturers assigned to each class are retained to preserve team-teaching information. Preferred-time and restricted-time records are converted into lecturer-indexed time-window mappings, while preferred-building records are converted into lecturer-indexed building lists. Class and room types are standardized into theory and laboratory categories, teaching days are mapped to numerical indices, and regular time slots are ordered by day and session.

The processed records are then converted into class, room, and time-slot objects. Class objects store course attributes, lecturer profiles, required weekly session units, and target meetings; room objects store capacity, type, and building; and time-slot objects store day, session number, and start and end times. Room-travel records are transformed into a pairwise travel-time mapping. Academic-calendar and holiday data are retained for the later calendar-projection stage and are not expanded into dated meetings before representative selection.

3.2 Weekly Timetable Formulation and Optimization

The first stage formulates and optimizes complete and feasible weekly timetables using the representation, constraints, and objectives described below. The resulting rank-1 front provides the alternatives used for representative selection in Section 3.3.

3.2.1 Solution Representation and Feasibility Constraints

A weekly timetable is represented as a chromosome composed of class-session assignments. Each gene assigns one weekly session unit of a class to one room and one time slot, as defined in Equation (1).

$$g_i = (c_i, r_i, t_i) \tag{1}$$

where c_i denotes the class assigned by gene i, r_i denotes the assigned room, and t_i denotes the assigned time slot. Let s_c denote the SKS value of class c. A complete chromosome must contain exactly s_c genes for every class. These genes are handled as one class block rather than as independent session assignments. The chromosome representation is designed to satisfy the hard constraints summarized in Table 3.

Table 3. Hard constraints for weekly timetabling

Code	Hard constraint
H1	Every class must receive exactly the number of weekly session units specified by its SKS value.
H2	A room may be assigned to only one class in the same time slot.
H3	A lecturer may not teach more than one class in the same time slot. For a team-taught class, the constraint is checked for every lecturer assigned to the class.
H4	Classes belonging to the same study program, semester, and class group may not be scheduled in the same time slot.
H5	The capacity of the assigned room must be greater than or equal to the class quota.
H6	A laboratory class must be assigned to a laboratory room.
H7	A theory class may use either a theory room or a laboratory room, provided that the room-capacity requirement is satisfied.
H8	A multi-SKS class must be assigned as one block in the same room and on the same teaching day using consecutive sessions. For a two-SKS class, the gap between its sessions may not exceed five minutes.
H9	A class may not be assigned to a time slot included in the restricted-time profile of any lecturer involved in the class.

All session units belonging to the same class are assigned to one room, on one teaching day, and in consecutive sessions. A one-SKS class therefore contains one gene, whereas two- and three-SKS classes contain two and three consecutive genes, respectively. For two-SKS classes, the time gap between the end of the first session and the beginning of the second session must not exceed five minutes, preventing the class block from spanning a major break in the daily schedule. A chromosome is considered complete when every active class appears exactly as many times as required by its SKS value. It is considered feasible when it is complete and satisfies all hard constraints. These constraints collectively define completeness, resource-conflict avoidance, room compatibility, block continuity, and lecturer-restriction requirements. The class-block constraint is preserved during construction, crossover, mutation, and final validation. Hard constraints are not converted into objective penalties; instead, incomplete or infeasible timetables are discarded, and the corresponding resource and restriction checks are reapplied during calendar-based recovery.

3.2.2 Soft Constraints and Objective Functions

Soft constraints are used to distinguish the relative quality of timetables that are already complete and feasible. Their violation does not invalidate a timetable, but contributes to the first objective function. The six soft-constraint components used in this study are summarized in Table 4.

Table 4. Soft-constraint components of f_1

Code	Soft-constraint component	Operational penalty
SC1	Lecturer preferred time	Counts class-session assignments outside all preferred-time windows specified for the lecturer. For team teaching, the preference of each assigned lecturer is evaluated.
SC2	Lecturer preferred building	Counts class-session assignments placed outside the lecturer's preferred building list. For team teaching, each lecturer profile is evaluated.
SC3	Consecutive lecturer classes in distant rooms	Counts consecutive teaching sessions for the same lecturer when the recorded inter-room travel value exceeds the implemented distance threshold.
SC4	Consecutive student-group classes in distant rooms	Counts consecutive sessions for the same study program, semester, and class group when the recorded inter-room travel value exceeds the implemented distance threshold.
SC5	Student-schedule compactness	Sums the number of empty session units occurring between consecutive classes of the same study program, semester, and class group on the same day.
SC6	Lecturer workload balance	Sums the daily teaching-session overload above four session units for each lecturer.

The first objective minimizes the total weighted soft-constraint penalty, as formulated in Eq. (2).

$$f_1(x) = \sum_{j=1}^{m} w_j \, v_j(x) \tag{2}$$

where x is a feasible timetable, m=6 is the number of soft-constraint components, $v_j(x)$ is the penalty produced by component j, and w_j is its corresponding weight. The first four components in Table 4 are expressed as violation counts, while student-schedule compactness and lecturer workload balance measure the accumulated magnitude of schedule gaps and daily overload, respectively.

All components are assigned an equal weight of $w_j = 1$. This setting prevents the experiment from introducing an unsupported institutional priority among the six components. Consequently, best-f_1 should be interpreted as the timetable with the lowest soft-constraint penalty under the equal-weight configuration used in this study, rather than as a timetable optimized using formally elicited stakeholder preferences. The second objective minimizes average room-capacity waste, as defined in Equation (3).

$$f_2(x) = \frac{1}{|C|}\sum_{c \in C} \max\left(0{,}1 - \frac{q_c}{cap_{r(c)}}\right) \quad (3)$$

where C is the set of uniquely scheduled classes, q_c is the quota of class c, $r(c)$ is the room assigned to that class, and $cap_{r(c)}$ is the capacity of the assigned room. Room waste is calculated once for each unique class rather than once for every gene. This prevents a multi-SKS class from contributing the same room-assignment waste repeatedly. A lower f_2 value indicates that the assigned room capacities more closely match the corresponding class quotas. This formulation follows the implemented objective calculation, which evaluates one room assignment per unique class.

The room-capacity waste represented by f_2 is different from the semester-level room-slot occupancy reported in Table 2. The former measures the capacity fit of individual class–room assignments, whereas the latter describes aggregate scheduling density by comparing the number of required weekly session units with the total number of regular room–slot combinations. Both objectives are minimized simultaneously and are not combined into a single scalar fitness value. NSGA-II therefore searches for non-dominated timetables representing different trade-offs between soft-constraint quality and room-capacity efficiency. Recoverability is not introduced as a third objective in this stage. Instead, it is evaluated downstream after representative solutions have been selected and projected onto the academic calendar.

3.2.3 NSGA-II Optimization Procedure

NSGA-II generates complete and feasible weekly timetables representing trade-offs between f_1 and f_2. The initial population is created using randomized greedy class-block placement, with difficult classes prioritized according to compatible-room availability, SKS, group and lecturer load, and quota. Incomplete initial chromosomes are repaired using direct insertion and, when necessary, limited evict-and-reinsert repair. The latter first allows at most two displaced classes and, if required, is repeated with at most three. Only complete and feasible chromosomes enter the population. Each generation evaluates f_1 and f_2, assigns non-domination ranks and crowding distances, and constructs a mating pool through binary tournament selection. Class-based crossover inherits complete class blocks, attempts at most 20 recombinations, and falls back to a parent copy when no feasible child is found. Reassignment and swap mutations are attempted independently, and a modified solution is retained only if it remains complete and feasible; no repair is applied to evolutionary offspring. Algorithm 1 summarizes the implementation sequence.

Algorithm 1. NSGA-II-based Weekly Timetable Optimization

INPUT: Processed scheduling data and NSGA-II parameters.
OUTPUT: Rank-1 Pareto front P of complete and feasible weekly timetables.

1: Construct candidate chromosomes using randomized greedy class-block placement.
2: Apply direct-insertion repair to incomplete initial chromosomes, followed, when necessary, by limited evict-and-reinsert repair.
3: Retain only complete and feasible chromosomes until the initial population reaches the required size.
4: Evaluate the population using f_1 and f_2, then calculate non-domination ranks and crowding distances.
5: Construct one mating pool for the current generation using binary tournament selection.
6: Repeatedly draw two parents at random from the mating pool and generate a child using class-based crossover or parent copying.
7: Independently attempt reassignment and swap mutation, retaining only complete and feasible outcomes.
8: Combine the parent and offspring populations and recalculate their ranks and crowding distances.
9: Form the next generation through elitist survivor selection, prioritizing feasible timetables with unique genotype signatures and backfilling with additional feasible timetables when required.
10: Repeat Steps 4–9 until the generation limit is reached, then return the rank-1 front.

After each generation, parent and offspring populations are combined for elitist survivor selection. Feasible timetables with unique genotype signatures are prioritized to preserve structural variation, while additional feasible timetables are included only when necessary to restore the population size. After the generation limit is reached, the rank-1 Pareto front is returned for representative selection.

3.3 Pareto-Representative Selection

For each semester and independent NSGA-II run, the optimization process produces a rank-1 Pareto front P containing complete and feasible weekly timetables that are non-dominated with respect to f_1 and f_2. Because the Pareto front contains several alternative trade-offs, it does not directly indicate which timetable should be used for the subsequent calendar-based disruption evaluation. Therefore, three representative timetables are selected from each Pareto front: best-f_1, best-f_2, and compromise.

Because structurally different timetables may share the same (f_1, f_2) pair, duplicate objective points are removed before representative selection. Let $P^u \subseteq P$ contain one timetable for each unique objective pair. This filtering removes repeated objective-space points without implying structural identity.

The best-f_1 and best-f_2 timetables are defined by Equation (4). Ties on the primary objective are resolved using the other objective.

$$x_{f_1} = \arg\min_{x \in P^u} f_1(x), \qquad x_{f_2} = \arg\min_{x \in P^u} f_2(x) \tag{4}$$

The third representative timetable is selected using a normalized distance-to-ideal criterion based on the compromise-programming principle in multi-objective decision making [32]. Directly comparing the distance of the objective values would be inappropriate because f_1 and f_2 are measured on different numerical scales. Each objective is therefore normalized to the interval [0,1] over P^u using Equation (5).

$$\bar{f}_k(x) = \frac{f_k(x) - f_k^{\min}}{f_k^{\max} - f_k^{\min}}, \qquad k \in \{1,2\} \tag{5}$$

where f_k^{min} and $f_k^{\max}$ denote the minimum and maximum values of objective k among the solutions in P^u. If $f_k^{\max} = f_k^{\min}$, the denominator is replaced with 1.0 to prevent division by zero. In that condition, the normalized value of the corresponding objective is zero for all solutions because the objective does not distinguish between the alternatives. However, this condition did not occur in any of the experimental runs.

Because both objectives are minimized, the ideal point in the normalized objective space is located at (0,0). The Compromise timetable is selected as the solution with the smallest Euclidean distance from this ideal point, as defined in Equation (6).

$$x_c = \arg\min_{x \in P^u} \sqrt{\bar{f}_1(x)^2 + \bar{f}_2(x)^2} \tag{6}$$

The compromise timetable is the solution closest to the ideal point in normalized objective space; it is not necessarily best on either individual objective or structurally intermediate between the two extremes. Representative selection is performed independently for each semester and seed, so best-f_1, best-f_2, and compromise denote solution categories rather than three fixed timetables. Selection uses only the original f_1 and f_2 values; no disruption or recovery information is introduced before the downstream evaluation.

3.4 Calendar Projection and Disruption Recovery

The calendar-based recovery stage shown in Fig. 1 converts each Pareto-representative weekly timetable into a date-based schedule and evaluates its response to controlled disruption scenarios. The stage consists of academic calendar projection, disruption-event construction, greedy direct rescheduling, and target-meeting fulfillment repair. The same procedure is applied independently to best-f_1, best-f_2, and compromise for every semester and seed.

3.4.1 Academic Calendar Projection

Each representative weekly timetable specifies the weekday, session interval, room, lecturer or teaching team, and class assigned to every weekly meeting block. To obtain a date-based timetable, all weekdays between the official start and end dates of the semester are first generated from the academic calendar, and each weekly meeting block is replicated on every date with the corresponding weekday. Every projected meeting is then assigned an actual date, calendar week, meeting sequence number, class, lecturer, room, and session interval. The room and time assignment remains identical to that of the source weekly timetable, and the projection is performed independently for the three representative timetables produced by each NSGA-II run.

Dates designated as national holidays or other non-instructional days are not removed from the projection. Meetings falling on these dates are retained as expected meetings and marked with a holiday status. Meetings on instructional dates are marked as active expected meetings. Retaining both categories preserves the complete set of meetings implied by the weekly timetable and makes it possible to identify meetings disrupted by the academic calendar.

The number of projected expected meetings is distinct from the minimum target number of meetings assigned to a class. The projection records every occurrence generated by the weekly timetable over the semester, whereas the target meeting value specifies the minimum class-level requirement that must remain satisfied after recovery. This distinction is used in the target-meeting fulfillment repair described in Section 3.4.4.

3.4.2 Calendar-Based Disruption Scenarios and Event Selection

Five cumulative disruption scenarios are defined in Table 5.

Table 5. Calendar-based disruption scenario definitions

Scenario	Disruption Type	Affected Resource	Evaluation Purpose
A	Holiday only	Instructional day	Evaluates recovery of meetings projected onto non-instructional dates.
B	Holiday and lecturer unavailability	Instructional day and selected lecturer	Adds the full-day unavailability of one lecturer.
C	Holiday and room unavailability	Instructional day and selected room	Adds the full-day unavailability of one room.
D	Holiday and session unavailability	Instructional day and selected session time window	Adds the unavailability of two consecutive regular sessions.
E	Combined disruption	Instructional day, lecturer, room, and session time window	Evaluates recovery when all tested disruption types are applied together.

Holiday disruptions are derived directly from the academic calendar. Additional lecturer, room, and session-window events are selected separately for each semester–seed combination using an aggregate-balanced rule over the three representatives. Eligible dates are restricted to active dates in the middle half of the semester, excluding holidays and boundary periods; lecturer and room events cover a full day, whereas session events cover two consecutive regular sessions.

For each candidate event, the number of unique meetings affected in best-f_1, best-f_2, and compromise is calculated. Candidates affecting all three representatives are prioritized and ranked by total affected meetings, the smallest affected count among representatives, and event date. If none affects all three, the candidate with the largest positive aggregate impact is used. The selected event is then applied identically to all three representatives, although affected counts may differ because their assignments differ. The selected events and their per-representative impacts are reported in Table 11.

A meeting is marked as affected when its date and assigned resource match the event; session events affect any meeting overlapping the unavailable window. Overlapping conditions in Scenario E are merged so that each meeting is processed once. Every scenario starts

from the original projected timetable, and recovered schedules are not carried across scenarios.

3.4.3 Greedy Direct Rescheduling

After the disruption events have been applied, all affected meetings are collected as recovery requests. The active baseline is formed from the non-holiday projected meetings that remain unaffected. Meetings falling on holidays and meetings removed by resource-unavailability events are recorded as cancelled meetings and are processed by the direct rescheduling procedure. The affected meetings are processed in a deterministic order, beginning with the earliest disrupted date. Meetings sharing the same date are subsequently ordered by semester, study program, class group, and original start session. Because the occupied schedule is updated immediately after every successful placement, this processing order may influence the candidates available to later meetings.

For each affected meeting, candidate dates are inspected chronologically and must occur strictly after the original disruption date. Only active, non-holiday academic dates are considered. On each candidate date, regular start sessions are ordered according to their absolute distance from the original start session. Candidate rooms are ordered by giving priority to the original room, followed by compatible rooms with the smallest sufficient capacity. A candidate is accepted only if it preserves the meeting duration and consecutive-session structure, occurs on an active non-holiday date, and satisfies the applicable room, lecturer, student-group, room-capacity, room-type, and restricted-time constraints in Table 3.

The first candidate satisfying this ordered search is selected. Its room, date, and session assignment are added to the active schedule, and the occupancy records are updated before the next affected meeting is processed. Emergency evening slots are excluded from direct rescheduling and are reserved for the repair stage. If no feasible regular candidate can be found on any future active date, the meeting is recorded as failed direct. The procedure does not re-optimize f_1 or f_2 and does not reconstruct the full timetable. It performs a local, event-driven repair of the affected meetings while leaving unaffected meetings unchanged.

3.4.4 Target-Meeting Fulfillment Repair and Validation

A failed direct placement is recorded at the expected-meeting level and does not necessarily imply that the class falls below its minimum meeting target. After direct rescheduling, valid scheduled and directly rescheduled meetings are counted for each class, and only classes below their target enter fulfillment repair. Consequently, failed direct meetings and makeup meetings do not have a one-to-one relationship.

For each deficient class, the repair procedure searches after its earliest disruption date, first in regular slots and then, if necessary, in the extended pool containing emergency evening slots. All room, lecturer, student-group, capacity, room-type, session-block, active-day, and restricted-time requirements remain enforced. Within the earliest date containing feasible alternatives, candidate selection prioritizes the original room, a smaller session shift, and lower room-capacity waste. Inserted makeup meetings immediately update occupancy and the class meeting count; the search continues until the target is met or no feasible candidate remains. A class that cannot reach its target is recorded as failed makeup. The final timetable is validated using scheduled, directly rescheduled, and makeup meetings. Validation checks target fulfillment, hard-constraint feasibility, and restricted-time compliance. Algorithm 2 summarizes direct rescheduling, fulfillment repair, and final validation.

Algorithm 2. Calendar-Based Disruption Recovery

INPUT: Projected calendar timetable T, disruption events E, academic calendar C, regular slots R, emergency slots M, scheduling-resource and constraint data, and class meeting targets H.

OUTPUT: Recovered calendar timetable T' and recovery-status records.

Identify the meetings affected by E and remove duplicated affected records.
Build the active baseline T' from unaffected meetings on instructional dates.
Sort the affected meetings in the predefined deterministic order.
FOR each affected meeting m:
    Search future active dates using regular slots and the date–session–room priority.
    IF a feasible candidate is found:
        Assign m to the candidate and update schedule occupancy.

Algorithm 2. cont

8: ELSE:
9: Record m as failed direct.
10: END IF
11: END FOR
12: Count the valid meetings of every class.
13: FOR each class c whose valid meeting count is below $H[c]$:
14: WHILE the valid meeting count of c is below $H[c]$:
15: Search for a feasible makeup meeting in regular slots.
16: IF no regular candidate exists, search the extended pool containing emergency slots.
17: IF a feasible candidate is found:
18: Insert the makeup meeting and update occupancy and the class meeting count.
19: ELSE:
20: Record c as failed makeup and BREAK.
21: END IF
22: END WHILE
23: END FOR
24: Validate target fulfillment, hard-constraint feasibility, and restricted-time compliance.
25: Return T' and the recovery-status records.

3.5 Evaluation Measures and Result Aggregation

The evaluation distinguishes four primary recovery metrics—the Direct Rescheduling Success Rate (DSR), Target Meeting Fulfillment (TMF), average recovery delay, and schedule perturbations—from a structural diagnostic and supporting descriptive measures. The primary metrics capture immediate recovery success, preservation of class meeting targets, temporal displacement, and the extent of schedule modification. Genotype-space distance characterizes structural differences among representatives, while the remaining measures describe optimization validity, diversity, disruption exposure, and additional operational effects.

3.5.1 Primary Recovery Metrics

DSR measures the proportion of affected meetings that are successfully transferred directly to feasible regular slots by the greedy rescheduling procedure, as defined in Equation (7).

$$\mathrm{DSR} = \frac{N^{\mathrm{direct}}}{N^{\mathrm{affected}}} \times 100\% \tag{7}$$

where N^{direct} is the number of affected meetings successfully rescheduled during the direct stage and N^{affected} is the total number of affected meetings. Makeup meetings are not included in N^{direct}because they are inserted later to resolve class-level meeting shortages rather than to replace individual affected meetings directly. A higher DSR indicates that a larger proportion of the disruption can be handled without invoking the target-fulfillment repair stage.

TMF evaluates whether the recovered timetable preserves the minimum meeting requirement of each class, as defined in Equation (8).

$$\mathrm{TMF} = \frac{N^{\mathrm{fulfilled}}}{N^{\mathrm{classes}}} \times 100\% \tag{8}$$

where $N^{\mathrm{fulfilled}}$ is the number of classes whose final valid meeting count is at least equal to their target and N^{classes} is the total number of evaluated classes. Meetings with scheduled, directly rescheduled, or makeup status are counted as valid meetings. TMF therefore evaluates the final class-level outcome after both direct rescheduling and target-meeting fulfillment repair.

DSR and TMF represent different levels of recovery. DSR is calculated at the affected-meeting level, whereas TMF is calculated at the class level. Consequently, a failed direct meeting does not necessarily reduce TMF or produce a makeup meeting. A makeup meeting is required only when the remaining valid meetings of its class fall below the minimum target.

Average recovery delay measures the temporal displacement introduced by successful direct rescheduling, as defined in Equation (9).

$$\text{AvgDelay} = \frac{1}{N^{\text{direct}}} \sum_{i=1}^{N^{\text{direct}}} \left(\text{date}_i^{\text{new}} - \text{date}_i^{\text{original}}\right) \tag{9}$$

where $\text{date}_i^{\text{new}}$ and $\text{date}_i^{\text{original}}$ denote the replacement and original dates of directly rescheduled meeting i, respectively. Delay is measured in calendar days. Makeup meetings are excluded because a makeup insertion addresses a class-level shortage and may not correspond uniquely to one original cancelled meeting.

Schedule perturbations count meeting-level changes, as defined in Equation (10).

$$\text{Perturbation} = N^{\text{direct}} + N^{\text{makeup}} \tag{10}$$

where N^{makeup} is the number of meetings inserted during target-meeting fulfillment repair. Each successful direct relocation and each makeup insertion count as one perturbation; failed direct meetings do not because they produce no new placement. A lower perturbation count is preferable only when disruption exposure and recovery success are comparable, so this metric is interpreted jointly with affected counts, DSR, and TMF. A recovered timetable is considered valid when every class reaches its target, no failed makeup remains, and no hard-constraint or lecturer restricted-time violation is found.

3.5.2 Structural and Supporting Metrics

Genotype-space distance is used to measure the structural difference between two Pareto-representative weekly timetables. For timetables A and B, the pairwise assignment distance is defined as in Equation (11).

$$\text{Distance}(A, B) = \frac{1}{N} \sum_{i=1}^{N} I\left[p_i(A) \neq p_i(B)\right] \times 100\% \tag{11}$$

where N is the number of corresponding timetable placements, $p_i(A)$ and $p_i(B)$ denote the room–weekday–session assignment of placement i in timetables A and B, and $I[\cdot]$ is an indicator function that returns 1 when the two assignments differ and 0 otherwise. A distance of 0% indicates identical placements, whereas a distance of 100% indicates that all compared placements differ. Genotype-space distance is a structural diagnostic rather than a performance objective; it is used to determine whether similar recovery outcomes arise from structurally similar timetables or from substantially different class placements.

Supporting optimization measures include completeness, feasibility, Pareto-front size, objective-space diversity, genotype diversity, and runtime. Objective-space diversity is described by the number and proportion of unique (f_1, f_2) pairs and the relative range of each objective, while genotype diversity reports distinct timetable structures.

Supporting recovery measures include affected rate, affected session rate, failed direct meetings, makeup and failed-makeup counts, direct room changes, and runtime. Affected session rate expresses disruption exposure in session-units to account for meeting duration. These measures characterize validity, exposure, and operational effects and are not treated as independent evidence of representative superiority.

3.5.3 Scenario Pooling and Multi-Seed Aggregation

The basic evaluation unit comprises one semester, one NSGA-II seed, one representative timetable, and one disruption scenario. The complete design therefore contains $2 \times 5 \times 3 \times 5 = 150$ recovery cases. Each scenario is evaluated independently from the original projected timetable. For every semester–seed–representative combination, Scenarios A–E are pooled by summing their underlying counts. Ratio metrics are then recomputed from the pooled numerators and denominators, average delay is calculated from total delay divided by direct successes, and count and runtime measures are summed. TMF is calculated per scenario, with the minimum value retained as the seed-level result. Finally, the five seed-level values for each semester and representative category are summarized using the arithmetic mean and sample standard deviation. Weighted affected rate and weighted DSR are calculated separately by pooling their counts across all seeds.

4. Results and Discussion

This section presents the experimental results and interprets them in relation to the deployment-oriented research question. The analysis first verifies that the weekly optimization stage consistently produces valid Pareto alternatives, then examines the selected representatives in objective and genotype spaces, and finally evaluates whether the choice among these representatives affects calendar-based disruption recovery. Sections 4.2–4.6 provide a brief interpretation of each result, while Section 4.7 synthesizes their broader scientific and practical implications.

4.1 Experimental Setup

The experiments followed the two-stage evaluation framework shown in Figure 1. The institutional datasets, semester characteristics, and preprocessing procedure are described in Section 3.1 and summarized in Table 2; they are therefore not repeated in this section.

Weekly timetable optimization was conducted separately for the odd and even semesters using the fixed NSGA-II configuration shown in Table 6. The parameter values were held constant across all runs because the purpose of this study was not to tune or compare NSGA-II configurations. Instead, the optimization stage was used to generate Pareto alternatives under a consistent setting, allowing the subsequent analysis to isolate the operational implications of selecting different Pareto representatives.

Table 6. Experimental configuration and computational environment

Parameter	Value
Platform	Kaggle Notebook
Computing environment	CPU without GPU or TPU acceleration
Programming language	Python
Main libraries	pandas, numpy, matplotlib
Optimization algorithm	NSGA-II
Population size	100
Number of generations	200
Crossover probability	0.9
Mutation probability per operator	0.3
Independent random seeds	11, 22, 33, 44, 55
Representative solutions	Best-f_1, Best-f_2, and Compromise
Recovery procedure	Greedy direct rescheduling followed by target-meeting fulfillment repair
Disruption scenarios	A, B, C, D, E

Five independent runs were performed for each semester to account for the stochastic behavior of NSGA-II. The random seeds affected the search process but did not change the input data, objective functions, constraints, or algorithm parameters. From the rank-1 front of each run, the best-f_1, best-f_2, and compromise timetables were selected independently using the procedure in Section 3.3.

Each selected timetable was subsequently projected onto the corresponding academic calendar and evaluated under the five disruption scenarios in Table 5. Within a given semester and seed, the three representatives were subjected to the same selected lecturer-, room-, and session-unavailability events and processed using the same recovery procedure. This matched design was used to reduce differences in external disruption exposure, so that the downstream comparison focuses on whether the initial deployment choice among Pareto representatives produces different recovery outcomes.

4.2 Weekly Timetable Optimization Results

Table 7 summarizes the weekly timetabling results across the five independent seeds. All ten runs produced complete and feasible weekly timetables. Every required session-unit was assigned, and no hard-constraint violation remained in the final schedules. Within the tested configuration, the construction, initialization repair, crossover, mutation, and survivor-

selection procedures therefore consistently provided valid weekly timetables for the subsequent representative-selection stage.

Table 7. Weekly timetabling optimization results across five independent seeds *

Metric	Odd Semester	Even Semester
Complete Rate (%) ↑	100.00	100.00
Feasible Rate (%) ↑	100.00	100.00
Pareto Front Size	24.20 ± 8.98	41.60 ± 33.27
Pareto Front Unique Objective Pairs	14.60 ± 5.64	18.40 ± 2.79
Final Population Unique Schedules	100.00 ± 0.00	100.00 ± 0.00
Runtime (min) ↓	190.64 ± 40.35	49.71 ± 9.86

* Values with ± are reported as mean ± sample standard deviation across five seeds. ↑/↓ indicate that higher/lower values are preferred; front-size and diversity measures are descriptive.

The even semester produced a larger but more variable rank-1 front than the odd semester. However, some non-dominated timetables shared the same (f_1, f_2) pair, so front size should not be interpreted as the number of distinct objective trade-offs. Section 4.4 examines this distinction at the seed level.

The final populations nevertheless retained full genotype diversity: all 100 schedules were structurally distinct in every run. Thus, repeated objective pairs did not result from exact timetable copies; different room–weekday–session arrangements could map to the same soft-constraint penalty and room-capacity waste. This supports the duplicate-objective filtering used before representative selection and motivates the genotype-space analysis in Section 4.5.

The odd-semester optimization required 190.64 ± 40.35 minutes on average, nearly four times the 49.71 ± 9.86 minutes required for the even semester. This difference is consistent with the institutional characteristics reported in Table 2. The odd semester contains more active classes and required session-units, fewer rooms, and substantially higher nominal room-slot occupancy. These conditions provide fewer readily feasible placement alternatives and increase the effort involved in constructing and evaluating complete schedules. However, because the two semester instances also differ in class composition, lecturer assignments, and room inventories, the runtime difference cannot be attributed to room-slot occupancy alone.

Thus, all runs supplied valid and structurally diverse Pareto alternatives for representative selection, while weekly optimization alone did not indicate which representative category would perform better after deployment.

4.3 Pareto Representative Solutions

Table 8 summarizes the objective values of the three Pareto-representative timetables selected from each of the five independent runs. The best-f_1 timetable represents the lowest weighted soft-constraint penalty available on the filtered Pareto front, whereas the best-f_2 timetable represents the lowest average room-capacity waste. The compromise timetable is selected using the normalized distance-to-ideal criterion defined in Eq. (6).

Table 8. Objective characteristics of Pareto-representative timetables across seeds

Semester	Representative Timetable	f_1 ↓ mean ± SD	f_1 min–max	f_2 ↓ mean ± SD	f_2 min–max
Odd	Best-f_1	268.20 ± 8.11	258-279	0.233203 ± 0.003381	0.229699-0.238095
	Best-f_2	299.40 ± 20.82	281-328	0.229116 ± 0.002050	0.226942-0.232469
	Compromise	282.40 ± 14.71	266-305	0.230785 ± 0.003332	0.227771-0.236340
Even	Best-f_1	127.40 ± 5.22	119-132	0.161039 ± 0.004005	0.157518-0.167736
	Best-f_2	159.60 ± 10.04	149-171	0.150335 ± 0.002249	0.148686-0.154295
	Compromise	139.80 ± 5.89	134-147	0.154224 ± 0.002824	0.151069-0.158832

The objective values show the expected trade-off between the two deployment priorities. In the odd semester, moving from best-f_1 to best-f_2 reduces the mean f_2 value from

0.233203 to 0.229116, a decrease of approximately 1.8%, but increases the mean f_1 penalty from 268.20 to 299.40, an increase of approximately 11.6%. The even semester shows the same pattern more strongly: f_2 decreases by approximately 6.7%, from 0.161039 to 0.150335, while f_1 increases by approximately 25.3%, from 127.40 to 159.60.

These percentages describe the observed change between the two objective extremes, but they should not be interpreted as a direct exchange rate between f_1 and f_2 because the objectives measure different quantities. Their main purpose is to show that improving room-capacity efficiency is accompanied by a loss in soft-constraint quality, and vice versa.

The mean objective values of the compromise timetable lie between those of the two objective-specific representatives in both semesters. This is consistent with its normalized distance-to-ideal definition: the compromise represents an intermediate objective-space trade-off rather than a solution that dominates either extreme. Figure 2 illustrates the representative-selection outcome for one rank-1 Pareto front from each semester. The best-f_1 and best-f_2 points mark the two objective-specific extremes, while the compromise point is selected using Eq. (6).

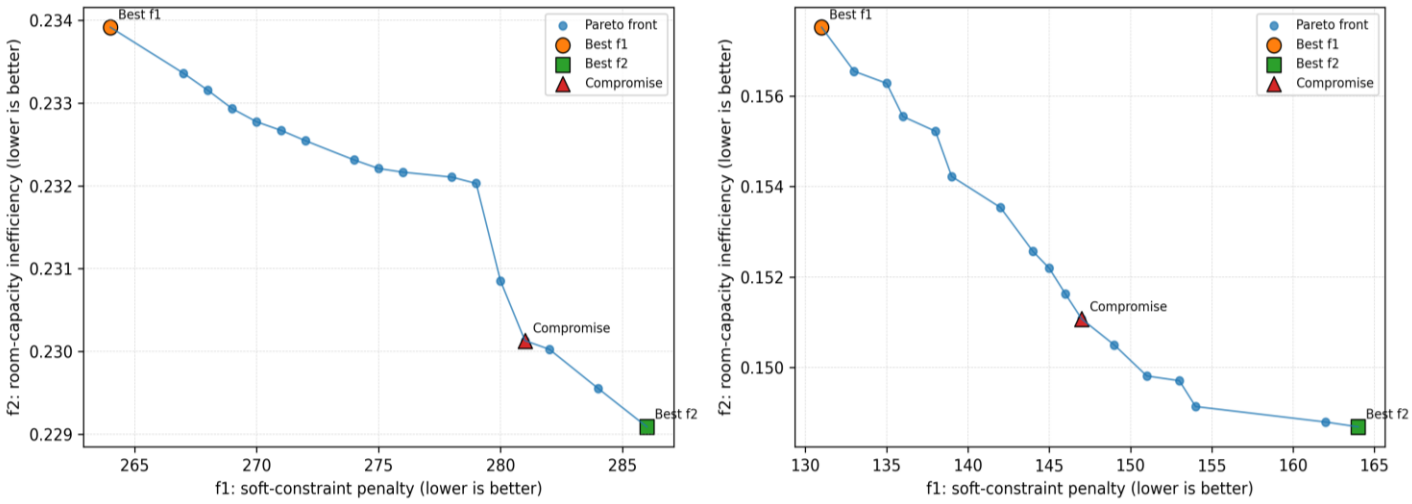

Figure 2. Illustrative rank-1 Pareto fronts for (**a**) the odd semester, seed 44, and (**b**) the even semester, seed 11. Highlighted points indicate best-f_1, best-f_2, and compromise; quantitative comparisons use all five seeds reported in Table 8.

The three representatives therefore encode distinct deployment priorities, but their weekly objective values alone do not determine their downstream recovery performance.

4.4 Pareto-Front Diversity across Seeds

The Pareto fronts differ not only in size but also in the number of distinct objective trade-offs they contain. Table 9 reports the front size, the proportion of unique objective pairs, and the relative range of each objective for every semester and seed. Across all seeds, the Pareto fronts span a wider relative range in f_1 than in f_2. In the odd semester, the f_1 range varies from 7.5% to 17.6%, whereas the f_2 range remains between 0.64% and 2.42%. The even semester exhibits wider fronts on both objectives, with f_1 ranges between 15.5% and 43.7% and f_2 ranges between 5.87% and 8.71%. Thus, the available non-dominated alternatives vary more substantially in soft-constraint penalty than in room-capacity waste, particularly in the odd semester.

This pattern helps explain why the reduction in f_2 between best-f_1 and best-f_2 in Table 8 is numerically modest compared with the accompanying change in f_1. The Pareto front offers a broader range of soft-constraint outcomes, while the room-capacity-waste values are more tightly clustered. Nevertheless, a narrow range does not imply that f_2 is unimportant; it indicates only that the feasible non-dominated schedules generated within a run tend to occupy a relatively limited interval on that objective.

Front size alone provides an incomplete description of the available trade-offs. The proportion of unique objective pairs ranges from 16.0% to 85.0% across the ten runs. The clearest case is even-semester seed 33, in which all 100 members of the final population are non-

dominated, but only 16.0% of the front corresponds to unique (f_1, f_2) pairs. Therefore, interpreting the front as containing 100 distinct objective trade-offs would substantially overstate its diversity. Because Pareto-front size, location, and spread vary across stochastic runs, representative categories must be evaluated across multiple seeds rather than inferred from one front. Figure 3 illustrates the variation in the location and shape of the rank-1 Pareto fronts across the five independent seeds.

Table 9. Objective-space diversity and width of the rank-1 Pareto fronts across seeds

Semester	Seed	Front Size	Unique Objective Pairs (%)	f_1 Range (%)	f_2 Range (%)
Odd	11	37	59.5	17.6	2.25
	22	14	71.4	8.9	1.48
	33	19	42.1	7.5	0.64
	44	29	58.6	8.3	2.11
	55	22	72.7	15.4	2.42
Even	11	20	85.0	25.2	5.94
	22	22	77.3	18.3	7.77
	33	100	16.0	15.5	7.25
	44	30	63.3	25.0	5.87
	55	36	63.9	43.7	8.71

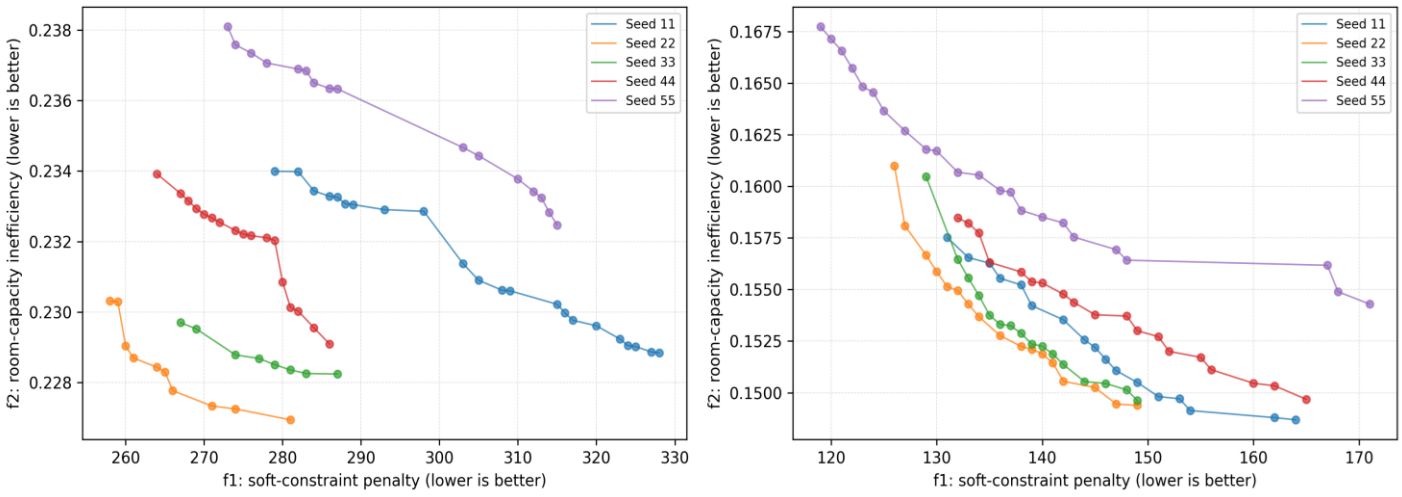

Figure 3. Pareto front overlays across five independent seeds: (a) odd semester; (b) even semester. Each line represents the rank-1 Pareto front from one seed using the original objective values.

4.5 Genotype-Space Differences

Objective-space proximity does not necessarily imply similarity in the underlying weekly timetable. Table 10 reports the pairwise genotype-space distance among best-f_1, best-f_2, and compromise for every seed. The distance is calculated using Eq. (11) as the proportion of corresponding class placements assigned to different room–weekday–session combinations.

The distance between best-f_1 and best-f_2 varies considerably across seeds. In the odd semester, the two extreme representatives differ in only 14.9% of class placements for seed 33 but in 98.4% and 98.9% for seeds 11 and 55, respectively. The even semester shows a similarly broad pattern, ranging from 8.0% in seed 33 to 100% in seed 55. Thus, selecting opposite ends of the Pareto front can produce either relatively similar or almost completely different weekly timetables, depending on the run.

The compromise timetable is also not consistently located between best-f_1 and best-f_2 in genotype space. In odd-semester seed 11, compromise differs from best-f_1 by only 14.4% of placements but differs from best-f_2 by 98.4%. In odd-semester seed 55, the opposite pattern appears: compromise is only 5.3% away from best-f_2 but 98.9% away from best-f_1.

Table 10. Pairwise timetable-assignment distance among Pareto-representative solutions across seeds*

Semester	Seed	Best-f_1 vs Best-f_2 (%)	Best-f_1 vs Compromise (%)	Best-f_2 vs Compromise (%)
Odd	11	98.4	98.4	14.4
	22	18.1	7.4	12.8
	33	14.9	14.9	4.3
	44	42.6	38.8	6.4
	55	98.9	5.3	98.9
Even	11	20	14.4	12
	22	10.4	8.8	6.4
	33	8	8	8
	44	16.8	8.8	12.8
	55	100	10.4	100

*A distance of 0% indicates identical placements, whereas 100% indicates that all compared placements differ. Greater distance denotes structural difference, not higher timetable quality.

These results do not conflict with the distance-to-ideal selection in Eq. (6). The compromise solution is intermediate in normalized objective space, but it is not required to be structurally intermediate between the two extreme timetables. A small change in objective values may be produced by many coordinated changes in class placement, while a timetable that is structurally close to one extreme may still occupy an intermediate position in objective space. The representatives therefore correspond to genuinely different timetable structures in several runs. Similar recovery outcomes cannot simply be attributed to identical deployments; they must be interpreted against both objective-space and genotype-space differences.

4.6 Calendar Projection and Disruption-Recovery Results

Each Pareto-representative weekly timetable was projected onto the corresponding academic-calendar period reported in Table 2. Meetings falling on non-instructional dates were retained as expected meetings and treated as holiday disruptions rather than removed from the projected timetable. The odd semester retained 92 active academic weekdays after holiday exclusion, whereas the even semester retained 82. The latter therefore had greater holiday exposure, but also substantially lower nominal room-slot occupancy. Tables 12 and 13 report the disruption-recovery results after Scenarios A–E were pooled for each seed and subsequently summarized across the five seeds as described in Section 3.5.3. Table 12 presents the four primary metrics that directly address the deployment-oriented research question, while Table 13 reports supporting measures of disruption exposure and operational behavior.

The DSR results provide no consistent recovery advantage for any representative category. In the odd semester, mean DSR ranges from 33.23% for best-f_1 to 36.19% for best-f_2, a maximum difference of 2.96 percentage points. In the even semester, the range is narrower, from 95.40% to 96.07%. In the even semester, the range is narrower, from 95.40% to 96.07%. In both cases, the differences among representative means are smaller than the observed seed-to-seed standard deviations. The weighted DSR values lead to the same ordering and nearly identical percentages, indicating that the observed pattern is not an artifact of small differences in affected-meeting counts. As illustrated in Fig. 4, the separation between semesters is substantially larger than the separation among best-f_1, best-f_2, and compromise within either semester.

The seed-level results in Fig. 5 provide a more direct view of the deployment comparison. In the odd semester, the lines cross repeatedly, indicating that the representative with the highest DSR changes across seeds. In the even semester, the three representatives produce very similar DSR values within each seed. These observations do not support a general rule that best-f_1, best-f_2, or compromise should always be selected to improve direct recoverability. To ensure that these comparisons are not influenced by substantially different disruption exposure among representative timetables, the aggregate-balanced procedure generated 30 additional disruption events. Of these, 18 affected all three representatives equally, while the largest observed difference among representatives was only three meetings. Table 11 summarizes the selected events and their per-representative impact-balance diagnostics. Overall,

disruption exposure remained closely comparable, although not perfectly identical, within each semester–seed combination.

Table 11. Selected additional disruption events and per-representative impact-balance diagnostics across seeds

Semester	Seed	Event	Date	Target Resource	Time Scope	Best-f_1	Best-f_2	Compromise	Imbalance Ratio *
Odd	11	Lecturer	26/09/2025	L1	Full day	2	2	2	1.00
		Room	02/10/2025	R1	Full day	4	7	7	1.75
		Session	26/09/2025	All rooms	Sessions 4–5	15	18	18	1.20
	22	Lecturer	29/09/2025	L2	Full day	3	3	3	1.00
		Room	02/10/2025	R1	Full day	8	8	8	1.00
		Session	02/10/2025	All rooms	Sessions 6–7	16	16	16	1.00
	33	Lecturer	02/10/2025	L1	Full day	3	3	2	1.50
		Room	01/10/2025	R1	Full day	8	8	8	1.00
		Session	02/10/2025	All rooms	Sessions 5–6	17	17	17	1.00
	44	Lecturer	29/09/2025	L3	Full day	2	3	3	1.50
		Room	30/09/2025	R1	Full day	8	8	8	1.00
		Session	29/09/2025	All rooms	Sessions 3–4	16	17	16	1.06
	55	Lecturer	29/09/2025	L4	Full day	3	2	3	1.50
		Room	29/09/2025	R1	Full day	7	5	7	1.40
		Session	26/09/2025	All rooms	Sessions 6–7	17	17	17	1.00
Even	11	Lecturer	01/04/2026	L3	Full day	2	3	3	1.50
		Room	31/03/2026	R2	Full day	3	3	3	1.00
		Session	31/03/2026	All rooms	Sessions 3–4	13	14	14	1.08
	22	Lecturer	31/03/2026	L3	Full day	3	3	3	1.00
		Room	02/04/2026	R3	Full day	3	3	3	1.00
		Session	02/04/2026	All rooms	Sessions 6–7	13	13	13	1.00
	33	Lecturer	30/03/2026	L5	Full day	2	2	2	1.00
		Room	31/03/2026	R4	Full day	5	5	5	1.00
		Session	10/04/2026	All rooms	Sessions 5–6	13	13	12	1.08
	44	Lecturer	01/04/2026	L6	Full day	2	2	2	1.00
		Room	02/04/2026	R1	Full day	6	6	6	1.00
		Session	10/04/2026	All rooms	Sessions 4–5	12	12	12	1.00
	55	Lecturer	02/04/2026	L7	Full day	2	3	2	1.50
		Room	01/04/2026	R5	Full day	3	3	3	1.00
		Session	01/04/2026	All rooms	Sessions 3–4	16	14	14	1.14

* The imbalance ratio is calculated as the largest affected-meeting count divided by the smallest positive affected-meeting count across the three representatives; a value of 1.00 indicates equal impact.

Table 12. Primary disruption-recovery results across five independent seeds *

Semester	Representative Solution	DSR (%) ↑	Weighted DSR (%) ↑	TMF (%) ↑	Average delay (days) ↓	Schedule perturbations
Odd	Best-f_1	33.23 ± 4.97	33.28	100.00 ± 0.00	11.42 ± 1.46	196.20 ± 31.67
	Best-f_2	**36.19** ± 3.72	**36.14**	100.00 ± 0.00	**11.41** ± 1.22	214.60 ± 19.78
	Compromise	35.51 ± 4.31	35.47	100.00 ± 0.00	11.94 ± 1.22	211.80 ± 24.13
Even	Best-f_1	95.40 ± 2.99	95.40	100.00 ± 0.00	**6.05** ± 0.57	1565.40 ± 45.13
	Best-f_2	95.73 ± 2.51	95.73	100.00 ± 0.00	6.28 ± 0.64	1566.80 ± 43.37
	Compromise	**96.07** ± 2.49	**96.06**	100.00 ± 0.00	6.24 ± 0.57	1572.60 ± 31.62

* DSR = direct rescheduling success rate; TMF = target-meeting fulfillment. Values with ± are mean ± sample standard deviation across five seeds; weighted DSR pools the underlying counts across all seeds. ↑/↓ indicate that higher/lower values are preferred. Schedule perturbations are descriptive extent-of-change counts with no unconditional preferred direction. Bold indicates the best mean within each semester for directional metrics and does not imply statistical significance.

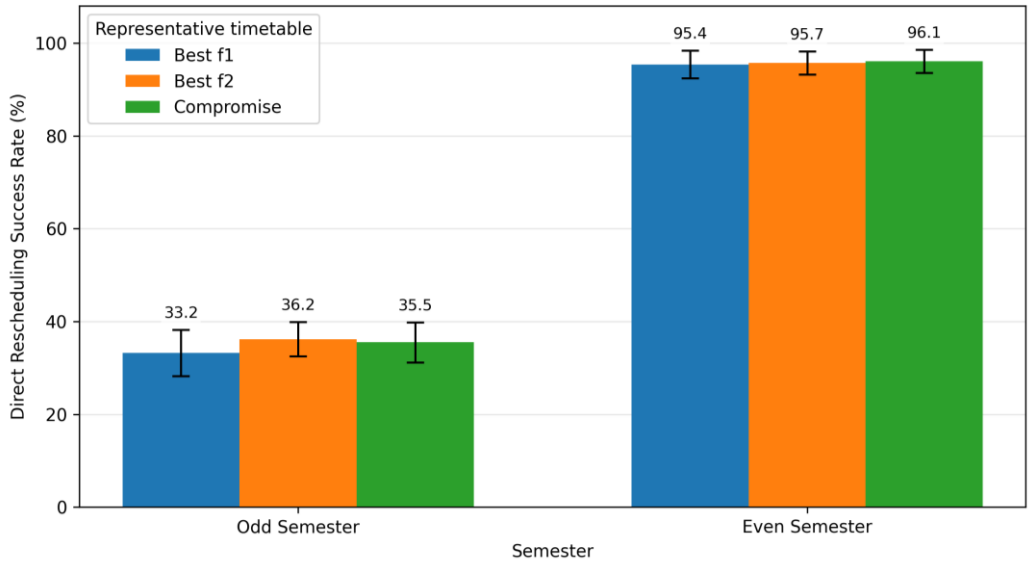

Figure 4. Direct Rescheduling Success Rate (DSR) across five seeds for the three Pareto-representative timetables. Bars show the mean DSR, and error bars indicate one sample standard deviation across seeds.

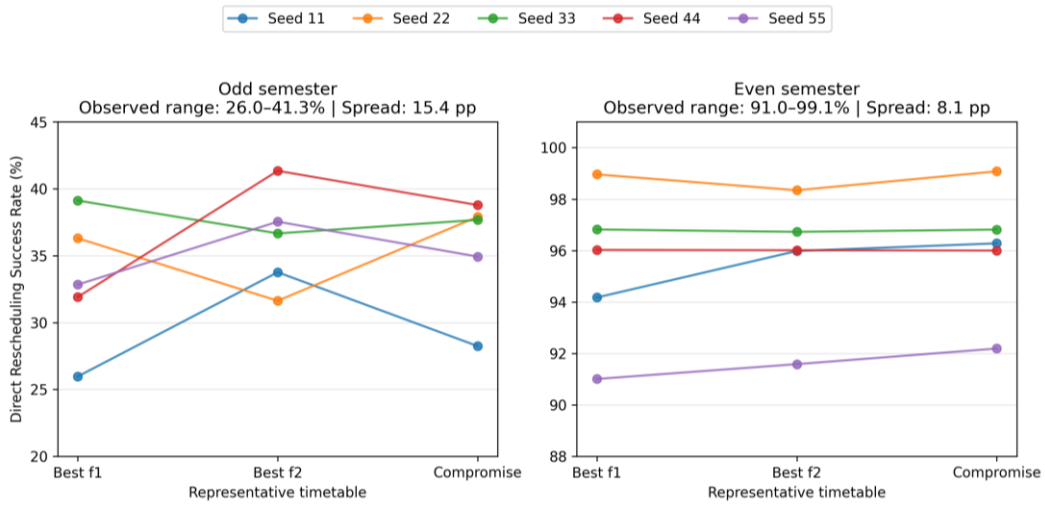

Figure 5. Seed-level DSR for the three Pareto-representative timetables. Each line connects results from the same NSGA-II seed. The panels use different y-axis scales and should be interpreted through within-seed patterns rather than absolute vertical distances.

All representative timetables reached 100% TMF in both semesters. Thus, meeting-level direct failures did not prevent any class from satisfying its minimum target after fulfillment repair. Average delay shows little separation among representatives but a clear contrast between semesters. Directly recovered meetings in the odd semester were delayed by approximately 11.4–11.9 calendar days, compared with approximately 6.1–6.3 days in the even semester. The denser odd-semester timetable therefore not only recovered fewer meetings directly, but the meetings that were successfully recovered generally required a later available date.

The perturbation counts require a different interpretation. The even semester records approximately 1,565–1,573 perturbations per representative, whereas the odd semester records approximately 196–215. This does not mean that the even semester was less recoverable. A perturbation is recorded when a meeting is successfully moved or a makeup meeting is inserted. The even semester exposed more meetings to holiday disruption and successfully relocated almost all of them, producing a large number of actual schedule changes. By contrast, many affected meetings in the odd semester could not be moved directly and therefore

did not become perturbations. Since this metric is a raw count, comparisons between semesters must be interpreted together with disruption exposure and DSR.

Table 13. Supporting disruption-exposure and operational measures across five independent seeds *

Metric	Odd Semester			Even Semester		
	Best-f_1	Best-f_2	Compromise	Best-f_1	Best-f_2	Compromise
Aff. Rate (%)	3.30 ± 0.13	3.32 ± 0.11	3.34 ± 0.10	13.75 ± 0.10	13.74 ± 0.10	13.73 ± 0.12
Aff. Rate Weighted (%)	3.30	3.32	3.34	13.75	13.74	13.73
Aff. Session Rate (%)	3.29 ± 0.08	3.30 ± 0.05	3.31 ± 0.03	13.90 ± 0.11	13.82 ± 0.18	13.86 ± 0.11
Failed Direct ↓	393.40 ± 27.76	379.20 ± 30.84	385.40 ± 32.55	75.20 ± 49.20	69.60 ± 40.75	64.20 ± 41.19
Makeup Meetings	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	7.40 ± 5.59	5.00 ± 5.29	6.00 ± 4.53
Failed Makeup Total ↓	0	0	0	0	0	0
Direct Room Changes	152.20 ± 22.88	172.60 ± 19.36	159.80 ± 31.36	1213.20 ± 53.77	1253.40 ± 77.82	1252.20 ± 57.48
Runtime (s) ↓	1538.05 ± 239.47	1334.23 ± 238.83	1425.47 ± 115.50	1934.04 ± 780.14	1939.17 ± 817.18	1771.78 ± 614.38

* Aff. = affected. Values with ± are mean ± sample standard deviation across five seeds. ↑/↓ indicate that higher/lower values are preferred where shown; affected-rate, makeup-meeting, and direct-room-change measures are descriptive.

The affected rates are nearly identical among the three representatives within each semester. The largest within-semester difference is only 0.04 percentage points in the odd semester and 0.02 percentage points in the even semester. The affected session rates show the same pattern. These results support the intended matched comparison: the representatives were exposed to similar disruption magnitudes, so the observed DSR differences are unlikely to be explained simply by one category receiving substantially fewer affected meetings.

The odd semester records approximately 379–393 failed direct meetings per representative but no makeup meetings. This does not indicate that the repair stage failed to operate. Rather, the remaining unaffected and directly rescheduled meetings were already sufficient for every class to satisfy its minimum target. In the even semester, a small number of class-level shortages remained after direct rescheduling, requiring an average of 5.0–7.4 makeup meetings. All of these shortages were successfully repaired, as indicated by zero failed makeups and 100% TMF.

Direct room changes follow the same exposure-sensitive pattern as perturbations. The even semester has many more room changes because it contains more affected meetings and successfully relocates a much larger proportion of them. Within each semester, however, neither room changes nor runtime consistently favor the same representative. These supporting measures therefore provide no additional basis for selecting one Pareto-representative category as operationally superior.

4.7 Synthesis of Findings and Practical Implications

The results support three main findings concerning the deployment of Pareto-representative timetables. The first finding is that representative selection does not produce a stable recovery ranking under the tested greedy procedure. Despite clear objective-space and genotype-space differences, the best-performing category changes across runs. This does not establish mathematical equivalence; it only shows that the experiment provides no basis for a universal recovery-oriented deployment rule.

This result extends conventional multi-objective timetabling studies, which primarily evaluate the quality and spread of Pareto alternatives at the optimization stage [10]–[12]. The present analysis shows that a clear design-time trade-off between soft-constraint penalty and room-capacity waste does not necessarily become a stable operational trade-off after deployment. In particular, minimizing room-capacity waste does not directly maximize recovery slack. The f_2 objective evaluates how closely room capacity matches class quota, but it does not measure the number or distribution of feasible future room–time alternatives. Similarly, f_1 does not explicitly reward recoverability. Neither objective therefore gives one representative a systematic advantage under the local recovery search.

The second finding is that the dominant recovery contrast occurs between semester settings and is consistent with differences in structural slack. The denser odd-semester setting is

consistent with fewer feasible alternatives for displaced meetings than the less dense even-semester setting, despite the latter having greater holiday exposure.

This interpretation is conceptually related to slack-based robustness research [19], but the roles of slack differ. Prior work incorporates or approximates robustness during optimization, whereas this study uses semester-level occupancy only as explanatory context after the original objectives have been held fixed. Moreover, the two semesters differ in class composition, lecturer assignments, room inventories, and holiday distributions as well as occupancy. The result should therefore be described as evidence consistent with a structural-slack explanation, not as a causal estimate of the isolated effect of room-slot occupancy.

The recovery strategy also differs from minimal-perturbation approaches that recompute a disrupted timetable using mathematical optimization or satisfiability methods [21], [22], [26]. The present study does not seek a globally optimal recovered timetable. The same chronological greedy search and target-fulfillment procedure are intentionally applied to all representatives. This common recovery mechanism makes it possible to examine whether the deployed baseline timetable itself creates a stable advantage, but it also limits the conclusion to the tested local-repair strategy. A different recovery algorithm could interact with the representatives differently.

The third finding is that multi-seed evaluation is necessary for deployment-oriented conclusions. Pareto-front size, objective-space width, unique objective pairs, genotype distance, and the within-seed ranking of representative DSR all vary across independent runs. A conclusion based on one seed could therefore identify a representative as superior even though that ordering reverses in another run. Repeating the complete representative-selection and recovery evaluation across five seeds reduces the risk of treating one stochastic outcome as a general deployment rule.

For timetable administrators, the results imply that the selection among best-f_1, best-f_2, and compromise may be guided primarily by the institutional priorities represented by f_1 and f_2 when recovery is handled using a procedure similar to the one tested here. Choosing a timetable with lower soft-constraint penalties, lower room-capacity waste, or a normalized compromise did not produce a consistent recovery disadvantage. This should not be interpreted as permission to ignore recoverability. The stronger operational risk appears in dense scheduling environments where displaced meetings have few feasible alternatives. Institutions should therefore evaluate structural bottlenecks and preserve room–time flexibility, particularly in semesters or timetable regions operating near capacity.

These findings remain bounded by the experimental design: one faculty, one academic year, two semester instances, equal soft-constraint weights, selected disruption scenarios, and a single greedy recovery procedure. Because no formal equivalence test was performed, the results indicate the absence of a consistent observed advantage rather than proof that representative selection never matters.

5. Conclusion

Under the tested calendar-based greedy recovery procedure, selecting the best-f_1, best-f_2, or compromise timetable did not produce a consistent operational advantage. The representative that performed best changed across seeds, and differences within a semester were small compared with the much larger contrast between the two semester settings. The central answer to the research question is therefore not that all Pareto representatives are equivalent, but that no representative category emerged as a reliable predictor of better disruption recovery in the evaluated setting.

The main contribution of this study is a deployment-oriented evaluation framework that separates representative selection from robustness optimization. The original objectives are held fixed, several timetables are selected from each Pareto front, and their downstream behavior is evaluated under matched calendar-based disruptions and an identical recovery procedure. This design makes it possible to examine the operational consequences of the deployment decision without adding recoverability as another optimization objective.

The results show that design-time differences in soft-constraint penalty and room-capacity waste can correspond to substantial structural differences without producing a stable recovery ranking. In contrast, the between-semester recovery pattern is more consistent with the availability of structural slack, while target-meeting fulfillment was preserved for all representative categories after repair.

Within the evaluated setting, the results provide no recovery-based reason to consistently prefer one representative category. Institutions using a comparable local greedy-repair procedure may therefore select among the evaluated representatives according to their preferred f_1–f_2 trade-off, while still evaluating recoverability explicitly and preserving feasible room–time alternatives in dense timetable regions.

These conclusions are limited to the evaluated faculty, academic year, disruption design, and greedy repair mechanism. Future studies should test the framework on additional institutional settings, use more direct local-slack indicators, increase the number of independent runs, and examine whether objective-aware rescheduling, minimal-perturbation optimization, or metaheuristic recovery reveals stronger differences among Pareto-representative timetables.

Author Contributions: Conceptualization: H.S.R. and R.S.; Methodology: H.S.R., R.S., and M.F.N.; Software: H.S.R.; Validation: H.S.R. and R.S.; Formal analysis: H.S.R.; Investigation: H.S.R. and R.S.; Resources: H.S.R. and R.S.; Data curation: H.S.R.; Writing—original draft preparation: H.S.R.; Writing—review and editing: R.S. and M.F.N.; Visualization: H.S.R.; Supervision: R.S.; Project administration: H.S.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets analyzed in this study contain institutional scheduling data and lecturer information from the Faculty of Information Technology and Data Science, Sebelas Maret University, and are not publicly available due to privacy and institutional restrictions. The data may be made available from the corresponding author upon reasonable request and with permission from the faculty.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] S. Abdipoor, R. Yaakob, S. L. Goh, and S. Abdullah, "Meta-heuristic approaches for the University Course Timetabling Problem," *Intell. Syst. with Appl.*, vol. 19, p. 200253, Sep. 2023, doi: 10.1016/j.iswa.2023.200253.

[2] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Comput. Ind. Eng.*, vol. 86, pp. 43–59, Aug. 2015, doi: 10.1016/j.cie.2014.11.010.

[3] M. C. Chen, S. N. Sze, S. L. Goh, N. R. Sabar, and G. Kendall, "A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities," *IEEE Access*, vol. 9, pp. 106515–106529, 2021, doi: 10.1109/ACCESS.2021.3100613.

[4] T. Amariesta, A. K. Wardani, R. N. Adila, and S. R. Nurshiami, "Implementation of Back Tracking Algorithm in The Scheduling of Mathematics Study Program Faculty of MIPA Unsoed," *Oper. Res. Int. Conf. Ser.*, vol. 3, no. 3, pp. 87–100, Sep. 2022, doi: 10.47194/orics.v3i3.129.

[5] G. S. Budhi, K. Gunadi, and D. A. Wibowo, "Genetic Algorithm for Scheduling Courses," in *Communications in Computer and Information Science*, 2015, pp. 51–63. doi: 10.1007/978-3-662-46742-8_5.

[6] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: a systematic review," *Ann. Oper. Res.*, vol. 275, no. 1, pp. 145–160, Apr. 2019, doi: 10.1007/s10479-017-2688-8.

[7] R. Lewis, "A survey of metaheuristic-based techniques for University Timetabling problems," *OR Spectr.*, vol. 30, no. 1, pp. 167–190, Nov. 2007, doi: 10.1007/s00291-007-0097-0.

[8] S. Yang and S. N. Jat, "Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling," *IEEE Trans. Syst. Man, Cybern., Part C Applications Rev.*, vol. 41, no. 1, pp. 93–106, Jan. 2011, doi: 10.1109/TSMCC.2010.2049200.

[9] A. Rezaeipanah, S. S. Matoori, and G. Ahmadi, "A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search," *Appl. Intell.*, vol. 51, no. 1, pp. 467–492, Jan. 2021, doi: 10.1007/s10489-020-01833-x.

[10] C. Y. Cheong, K. C. Tan, and B. Veeravalli, "A multi-objective evolutionary algorithm for examination timetabling," *J. Sched.*, vol. 12, no. 2, pp. 121–146, Apr. 2009, doi: 10.1007/s10951-008-0085-5.

[11] P. Côté, T. Wong, and R. Sabourin, "A Hybrid Multi-objective Evolutionary Algorithm for the Uncapacitated Exam Proximity Problem," in *Lecture Notes in Computer Science*, 2005, pp. 294–312. doi: 10.1007/11593577_17.

[12] N. Leite, R. Neves, N. Horta, F. Melício, and A. C. Rosa, "Solving a Capacitated Exam Timetabling Problem Instance Using a Bi-objective NSGA-II," in *Studies in Computational Intelligence*, 2015, pp. 115–129. doi: 10.1007/978-3-319-11271-8_8.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[14] F. Dunke and S. Nickel, "A matheuristic for customized multi-level multi-criteria university timetabling," *Ann. Oper. Res.*, vol. 328, no. 2, pp. 1313–1348, Sep. 2023, doi: 10.1007/s10479-023-05325-2.

[15] M. Davison, A. Kheiri, and K. G. Zografos, "Modelling and solving the university course timetabling problem with hybrid teaching considerations," *J. Sched.*, vol. 28, no. 2, pp. 195–215, Apr. 2025, doi: 10.1007/s10951-024-00817-w.
[16] T. Müller, H. Rudová, and Z. Müllerová, "Real-world university course timetabling at the International Timetabling Competition 2019," *J. Sched.*, vol. 28, no. 2, pp. 247–267, Apr. 2025, doi: 10.1007/s10951-023-00801-w.
[17] A. Gülcü and C. Akkan, "Robust university course timetabling problem subject to single and multiple disruptions," *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 630–646, Jun. 2020, doi: 10.1016/j.ejor.2019.11.024.
[18] C. Akkan and A. Gülcü, "A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem," *Comput. Oper. Res.*, vol. 90, pp. 22–32, Feb. 2018, doi: 10.1016/j.cor.2017.09.007.
[19] C. Akkan, A. Gülcü, and Z. Kuş, "Bi-criteria simulated annealing for the curriculum-based course timetabling problem with robustness approximation," *J. Sched.*, vol. 25, no. 4, pp. 477–501, Aug. 2022, doi: 10.1007/s10951-022-00722-0.
[20] B. Bassimir and R. Wanka, "On the computation of robust examination timetables: methods and experimental results," *J. Sched.*, vol. 28, no. 2, pp. 159–181, Apr. 2025, doi: 10.1007/s10951-024-00815-y.
[21] A. E. Phillips, C. G. Walker, M. Ehrgott, and D. M. Ryan, "Integer programming for minimal perturbation problems in university course timetabling," *Ann. Oper. Res.*, vol. 252, no. 2, pp. 283–304, May 2017, doi: 10.1007/s10479-015-2094-z.
[22] M. Lindahl, T. Stidsen, and M. Sørensen, "Quality recovering of university timetables," *Eur. J. Oper. Res.*, vol. 276, no. 2, pp. 422–435, Jul. 2019, doi: 10.1016/j.ejor.2019.01.026.
[23] A. Lemos, P. T. Monteiro, and I. Lynce, "Minimal Perturbation in University Timetabling with Maximum Satisfiability," in *Lecture Notes in Computer Science*, 2020, pp. 317–333. doi: 10.1007/978-3-030-58942-4_21.
[24] T. Müller, H. Rudová, and R. Barták, "Minimal Perturbation Problem in Course Timetabling," in *Lecture Notes in Computer Science*, 2005, pp. 126–146. doi: 10.1007/11593577_8.
[25] C. Akkan, A. Gülcü, and Z. Kuş, "Minimum penalty perturbation heuristics for curriculum-based timetables subject to multiple disruptions," *Comput. Oper. Res.*, vol. 132, p. 105306, Aug. 2021, doi: 10.1016/j.cor.2021.105306.
[26] A. Lemos, P. T. Monteiro, and I. Lynce, "Disruptions in timetables: a case study at Universidade de Lisboa," *J. Sched.*, vol. 24, no. 1, pp. 35–48, Feb. 2021, doi: 10.1007/s10951-020-00666-3.
[27] A. Lemos, F. S. Melo, P. T. Monteiro, and I. Lynce, "Room usage optimization in timetabling: A case study at Universidade de Lisboa," *Oper. Res. Perspect.*, vol. 6, p. 100092, 2019, doi: 10.1016/j.orp.2018.100092.
[28] C. Beyrouthy, E. K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes, "Towards improving the utilization of university teaching space," *J. Oper. Res. Soc.*, vol. 60, no. 1, pp. 130–143, Jan. 2009, doi: 10.1057/palgrave.jors.2602523.
[29] W. Herroelen and R. Leus, "Robust and reactive project scheduling: a review and classification of procedures," *Int. J. Prod. Res.*, vol. 42, no. 8, pp. 1599–1620, Apr. 2004, doi: 10.1080/00207540310001638055.
[30] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *J. Sched.*, vol. 12, no. 4, pp. 417–431, Aug. 2009, doi: 10.1007/s10951-008-0090-8.
[31] H. Aytug, M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy, "Executing production schedules in the face of uncertainties: A review and some future directions," *Eur. J. Oper. Res.*, vol. 161, no. 1, pp. 86–110, Feb. 2005, doi: 10.1016/j.ejor.2003.08.027.
[32] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidiscip. Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004, doi: 10.1007/s00158-003-0368-6.