Computer Science and Information Systems 23(3):1027–1052 https://doi.org/10.2298/CSIS251220025C

Examination Timetabling with Independent Resource Pools: A Multi-Stage Genetic Algorithm Approach

Khalil Chebil[1], Mahdi Khemakhem[1], Abdallah Hammad[1], and Mohammed Alanazi[1]

Department of Computer Science
College of Computer Engineering and Sciences
Prince Sattam Bin Abdulaziz University
Al Kharj 11942, Saudi Arabia
k.chebil@psau.edu.sa (corresponding author),
m.khemakhem@psau.edu.sa, 442052035@std.psau.edu.sa, 442052038@std.psau.edu.sa

Abstract. The Examination Timetabling Problem (ETP) is an NP-hard combinatorial optimization problem that has been extensively studied over the past decades, with methodologies ranging from classical heuristics to contemporary machine learning approaches. This paper provides a comprehensive literature review and proposes a novel two-stage genetic algorithm based decomposition combined with a clear linear mathematical formulation. A unique contribution of this work is addressing the practical complexity of two-separated examination facilities at PSAU, where male and female students study in completely separate buildings with independent examination rooms, capacities, and supervisory staff, yet must follow a synchronized examination schedule. This creates a coupled dual-ETP problem where two independent resource allocation problems must share identical time slot assignments, significantly increasing model complexity and constraint interactions. The proposed algorithm decomposes the problem into two sequential stages,time slot assignment and room assignment to effectively reduce computational complexity while maintaining solution quality. Additionally, a systematic comparison with Mixed-Integer Programming (MIP) formulation is provided to establish the trade-offs between exact and metaheuristic approaches. The algorithm is further benchmarked against simulated annealing and tabu search baselines to provide broader methodological context, demonstrating its advantage on coupled dual-resource instances. The algorithm shows particular strength in scalability, successfully solving large instances where MIP becomes computationally intractable, while maintaining competitive solution quality.

Keywords: Examination, Timetabling, Genetic Algorithm, Optimization.

1. Introduction

The examination timetabling problem is a complex combinatorial optimization challenge, fundamentally involving four key parameters: a set of times (T), a set of resources (R), a set of exams (E), and a set of constraints (C). Research into computational timetabling for educational schedules dates back to the 1960s, coinciding with the rise in educational institution enrollment and limited computer power. Despite decades of research and the development of numerous sophisticated methodologies—from early graph coloring heuristics to contemporary machine learning approaches—the problem remains challenging, particularly in formulating soft constraints mathematically, as these can vary significantly between universities based on specific requirements.

While the timetabling literature has extensively studied single-resource-pool formulations, practical institutions frequently face heterogeneous resource environments where multiple independent resource sets must be coordinated. Similar situations arise in multi-campus universities, distributed manufacturing scheduling, and healthcare shift planning with separated wards. The scheduling literature on heterogeneous resource pools [18, 5] has addressed aspects of this challenge, but the specific coupling constraint—where completely disjoint resource pools must share identical time assignments—remains underexplored in the examination timetabling context. This paper contributes to addressing this gap.

This paper makes several contributions to the examination timetabling literature. First, it presents a focused literature review organized by methodology, critically comparing approaches and identifying specific gaps that motivate the proposed method. Second, it proposes a two-stage genetic algorithm decomposition approach combined with a mathematical formulation that linearizes inherently nonlinear constraints using standard big-M and auxiliary variable techniques. Third, it demonstrates the algorithm's effectiveness on standard benchmarks (Toronto datasets) and validates its practical applicability through successful deployment for gender-separated examination facilities at Prince Sattam Bin Abdulaziz University. The PSAU case represents an instance of the broader class of coupled heterogeneous resource problems: male and female students study in completely separate buildings located in different parts of the campus, each with independent examination rooms, distinct capacity constraints, and separate supervisory personnel. Despite these separated resources, both populations must follow an identical synchronized examination schedule for administrative and academic coordination. This creates what is essentially a coupled dual-ETP problem—two independent examination timetabling problems (one for each building) that must produce identical time slot assignments while managing completely disjoint room resources. This coupling significantly increases model complexity, as feasible solutions must simultaneously satisfy constraints across both buildings while maintaining schedule synchronization. Finally, the paper provides systematic comparisons with Mixed-Integer Programming approaches and other metaheuristic baselines (simulated annealing, tabu search), establishing clear trade-offs between exact and metaheuristic methods.

The remainder of this paper is structured as follows: Section 2 presents a synthesized literature review organized by methodology with critical comparison. Section 3 presents a detailed mathematical formulation of the problem, including hard and soft constraints. Section 4 describes the proposed multi-stage genetic algorithm decomposition approach, including pseudocode and complexity analysis. Section 5 discusses experimental results on benchmark datasets and real-world applications. Finally, Section 6 concludes the paper and suggests directions for future research.

2. Literature Review

This section provides a critical synthesis of examination timetabling methods, organized by methodology rather than chronology, to identify the specific gaps that motivate the proposed approach. Rather than exhaustive coverage, we focus on representative works that shaped each methodological stream and highlight comparative strengths and limitations.

2.1. Constructive and Graph-Based Heuristics

Graph coloring heuristics form the foundation of constructive approaches for examination timetabling. Sabar et al. [32, 30] proposed graph coloring constructive hyper-heuristics, while Demeester et al. [17] introduced practical benchmarks. Pillay [28] demonstrated adaptive hyper-heuristic evolution, and Pillay and Özcan [29] automated heuristic generation via genetic programming. These methods excel at producing initial feasible solutions rapidly but are limited in soft constraint optimization, motivating their use as initialization strategies rather than standalone solvers.

2.2. Local Search and Neighborhood-Based Methods

Tabu search has proven particularly effective: Abdullah and Turabieh [2] demonstrated multi-neighbourhood structures within memetic tabu search, while Pais and Amaral [27] introduced adaptive tabu list management via fuzzy inference. Simulated annealing variants achieved state-of-the-art results: Battistutta et al. [9] presented feature-based parameter tuning, Bellio et al. [10] developed two-stage multi-neighborhood SA achieving top benchmark results, and Leite et al. [24] achieved competitive quality with reduced computation time. Burke and Bykov [13] introduced late acceptance hill-climbing, requiring fewer parameters than SA while maintaining competitive quality. These approaches demonstrate strong intensification capabilities but typically require good initial solutions, making them complementary to constructive heuristics.

A critical observation is that many of these methods operate on unified resource pools; their behavior under coupled heterogeneous resource constraints remains unexplored.

2.3. Population-Based and Evolutionary Methods

Genetic algorithms and swarm intelligence methods provide broader search space exploration. Abdullah and Alzaqebah [1] developed hybrid bee algorithm-GA approaches, while Al-Betar et al. [4] demonstrated effective memetic techniques. Sabar et al. [31] used grammatical evolution to automatically generate heuristics. Lei et al. [22, 23] applied coevolutionary and MOEA/D-based memetic algorithms for multi-objective formulations. Ahandani et al. [6] and Nand et al. [26] explored PSO and firefly variants respectively.

While population-based methods offer better exploration, they generally sacrifice solution precision compared to local search methods. Multi-stage decomposition—as demonstrated by Gogos et al. [19]—addresses this by reducing the search space for each evolutionary stage.

2.4. Mathematical Programming and Exact Methods

MIP formulations provide optimal solutions for small instances. Arbaoui and Moukrim [8] demonstrated preprocessing to reduce computational requirements, while Cataldo et al. [15] addressed curriculum-based formulations. Al-hawari et al. [5] showed that three-phase

ILP decomposition can make MIP tractable for larger instances. Carlsson et al. [14] provided comprehensive comparisons showing hybrid MIP-metaheuristic approaches can outperform pure methods. Arbaoui et al. [7] studied lower bounds for spacing constraints, providing theoretical benchmarks.

Key limitation: MIP scalability degrades rapidly with problem size, particularly when coupled constraints (such as our dual-building synchronization) expand the variable space.

2.5. Multi-Objective and Practical Applications

Recent work increasingly addresses real-world complexity. Mukhlason et al. [25] incorporated fairness objectives, and Güler et al. [20] addressed scarce resource scenarios. Çimen et al. [16] integrated invigilator assignment, and Mohmad Kahar and Kendall [21] addressed multi-resource scheduling. Elloumi et al. [18] analyzed room allocation as a distinct subproblem. Van Bulck et al. [33] achieved strong ITC-2007 results through careful neighborhood design. Abou Kasm et al. [3] relaxed hard constraints for flexible scheduling.

2.6. Research Gap and Contributions

From this analysis, we identify four specific gaps that motivate the present work:

1. **Coupled heterogeneous resources:** Existing formulations assume unified resource pools. While heterogeneous resource scheduling exists in broader OR literature, the specific coupling constraint—disjoint room pools sharing synchronized time assignments—is not addressed in examination timetabling.
2. **Linearization gap:** Many formulations present nonlinear constraints (bilinear products of binary variables, quadratic objectives) without providing linearized equivalents needed for MIP solvers.
3. **Decomposition-exact integration:** Few works systematically compare decomposed metaheuristics against exact methods under unified evaluation metrics.
4. **Practical deployment:** Gap between benchmark problem simplifications and real-world operational requirements involving gender-separated or multi-campus facilities.

This paper aims to contribute by providing a linearized mathematical formulation that can accommodate such complexities while employing a multi-stage genetic algorithm decomposition approach. The proposed method combines the clarity of mathematical programming formulations with the scalability of metaheuristic optimization, addressing coupled heterogeneous resource pools as required by Prince Sattam Bin Abdulaziz University.

3. Problem Formulation

To effectively address the Exam Timetabling Problem (ETP), we first establish a clear mathematical formulation and define the necessary notations. This formulation will serve as the foundation for developing algorithms to solve the ETP.

3.1. Notations

Let us define the following sets and parameters:

Sets:

- **E**: Set of exams to be scheduled, indexed by e ($e = 1, 2, \dots, |E|$)
- **T**: Set of available time slots, indexed by t ($t = 1, 2, \dots, |T|$)
- **R**: Set of available rooms, indexed by r ($r = 1, 2, \dots, |R|$)
- **S**: Set of students, indexed by s ($s = 1, 2, \dots, |S|$)

Parameters:

- n_e: Number of students enrolled in exam e
- cap_r: Capacity of room r
- d_e: Duration of exam e (in time slots)
- $C_{e,e'}$: Conflict matrix where $C_{e,e'} = 1$ if exams e and e' have common students, 0 otherwise
- $w_1, w_2, \dots, w_k$: Weights for different soft constraint violations

Decision Variables:

- $x_{e,t}$: Binary variable equal to 1 if exam e is scheduled in time slot t, 0 otherwise
- $y_{e,r}$: Binary variable equal to 1 if exam e is assigned to room r, 0 otherwise
- $z_{e,r,t}$: Binary auxiliary variable equal to 1 if exam e is assigned to room r **and** time slot t, i.e., $z_{e,r,t} = x_{e,t} \cdot y_{e,r}$

3.2. Hard Constraints

The following hard constraints must be satisfied for a feasible timetable:

HC1: Exam Assignment

Each exam must be scheduled in exactly one time slot:

$$\sum_{t \in T} x_{e,t} = 1, \quad \forall e \in E \tag{1}$$

HC2: No Student Conflicts

Exams with common students cannot be scheduled simultaneously:

$$x_{e,t} + x_{e',t} \leq 1, \quad \forall e, e' \in E \text{ with } C_{e,e'} = 1, \forall t \in T \tag{2}$$

HC3: Room Assignment

Each exam must be assigned to exactly one room:

$$\sum_{r \in R} y_{e,r} = 1, \quad \forall e \in E \tag{3}$$

HC4: Room Capacity

The room capacity must not be exceeded:

$$n_e \cdot y_{e,r} \leq cap_r, \quad \forall e \in E, \forall r \in R \tag{4}$$

HC5: Room Availability

A room can host at most one exam at any given time:

$$\sum_{e\in E} x_{e,t} \cdot y_{e,r} \leq 1, \quad \forall t \in T, \forall r \in R \tag{5}$$

Since this constraint involves a bilinear product of two binary decision variables, we introduce the auxiliary variables $z_{e,r,t} = x_{e,t} \cdot y_{e,r}$ and enforce the equivalence via McCormick / big-M constraints, yielding the following linear formulation:

$$z_{e,r,t} \leq x_{e,t}, \quad \forall e \in E, \forall r \in R, \forall t \in T \tag{6}$$

$$z_{e,r,t} \leq y_{e,r}, \quad \forall e \in E, \forall r \in R, \forall t \in T \tag{7}$$

$$z_{e,r,t} \geq x_{e,t} + y_{e,r} - 1, \quad \forall e \in E, \forall r \in R, \forall t \in T \tag{8}$$

$$\sum_{e\in E} z_{e,r,t} \leq 1, \quad \forall t \in T, \forall r \in R \tag{9}$$

3.3. Soft Constraints

The following soft constraints aim to improve timetable quality and should be minimized:

SC1: Spread of Exams

Minimize the number of students having exams scheduled in consecutive time slots:

$$\text{Minimize} \quad \sum_{s\in S} \sum_{t\in T} \sum_{e,e'\in E_s} x_{e,t} \cdot x_{e',t+1} \tag{10}$$

Since this objective involves bilinear terms, we introduce auxiliary variables $q_{e,e',t}$ representing the product $x_{e,t} \cdot x_{e',t+1}$ and enforce the equivalence via the following linear constraints:

$$q_{e,e',t} \leq x_{e,t}, \quad \forall s \in S, \forall e, e' \in E_s, \forall t \in T \tag{11}$$

$$q_{e,e',t} \leq x_{e',t+1}, \quad \forall s \in S, \forall e, e' \in E_s, \forall t \in T \tag{12}$$

$$q_{e,e',t} \geq x_{e,t} + x_{e',t+1} - 1, \quad \forall s \in S, \forall e, e' \in E_s, \forall t \in T \tag{13}$$

The equivalent linear SC1 objective becomes:

$$\text{Minimize} \quad \sum_{s\in S} \sum_{t\in T} \sum_{e,e'\in E_s} q_{e,e',t} \tag{14}$$

where E_s is the set of exams taken by student s.

SC2: Period Utilization

Balance the distribution of exams across time slots:

$$\text{Minimize} \quad \sum_{t\in T} \left(\sum_{e\in E} x_{e,t} - \frac{|E|}{|T|} \right)^2 \tag{15}$$

Since this objective is quadratic, we reformulate it using an equivalent absolute deviation formulation with auxiliary variables $\delta_t^+, \delta_t^- \geq 0$:

$$\sum_{e\in E} x_{e,t} - \frac{|E|}{|T|} = \delta_t^+ - \delta_t^-, \quad \forall t \in T \tag{16}$$

$$\delta_t^+, \delta_t^- \geq 0, \quad \forall t \in T \tag{17}$$

The equivalent linear SC2 objective becomes:

$$\text{Minimize} \quad \sum_{t \in T} (\delta_t^+ + \delta_t^-) \tag{18}$$

SC3: Room Utilization
Minimize underutilization of rooms:

$$\text{Minimize} \quad \sum_{e \in E} \sum_{r \in R} y_{e,r} \cdot \left(1 - \frac{n_e}{cap_r}\right) \tag{19}$$

SC4: Large Exam Penalty
Penalize scheduling large exams in less preferred time slots:

$$\text{Minimize} \quad \sum_{e \in E} \sum_{t \in T_{unpreferred}} n_e \cdot x_{e,t} \tag{20}$$

3.4. Objective Function

The overall objective is to minimize a weighted sum of soft constraint violations while ensuring all hard constraints are satisfied:

$$\text{Minimize} \quad Z = w_1 \cdot SC1 + w_2 \cdot SC2 + w_3 \cdot SC3 + w_4 \cdot SC4 \tag{21}$$

Subject to: HC1, HC2, HC3, HC4, HC5

The weights w_1, w_2, w_3, w_4 can be adjusted based on institutional priorities and requirements.

Relationship Between Mathematical Formulation and GA Fitness The GA fitness function (Eq. 18 in Section 4) is a direct operationalization of this mathematical objective. The hard constraints (HC1–HC5) are mapped to large penalty terms (P_{h1}, P_{h2}) that dominate the fitness value when violated, driving the search toward feasible regions. The soft constraints (SC1–SC4) are mapped to weighted penalty terms ($w_1 V_{s1} + w_2 V_{s2} + w_3 V_{s3}$) that guide optimization quality once feasibility is achieved. Specifically: V_{h1} corresponds to HC2 violations, V_{h2} penalizes excessive daily load (a practical operationalization related to HC1 and student comfort), and V_{s1}, V_{s2}, V_{s3} correspond to SC1 with varying proximity windows (consecutive slots, same day, consecutive days). The two-stage decomposition separates time slot constraints (HC1, HC2, SC1, SC2) from room constraints (HC3–HC5, SC3, SC4), allowing each GA stage to focus on its corresponding subset of the formulation.

3.5. Gender-Separated Buildings: A Coupled Dual-ETP Formulation

At Prince Sattam Bin Abdulaziz University, an institutional constraint significantly increases problem complexity—analogous to heterogeneous resource pool problems studied in scheduling theory [18, 5]: male and female students are educated in completely separate buildings with independent physical resources, yet must follow a synchronized examination schedule. This creates what we term a **coupled dual-ETP problem**.

Problem Structure:

Let us denote:

- E^M: Set of male student exams
- E^F: Set of female student exams
- R^M: Set of examination rooms in the male building
- R^F: Set of examination rooms in the female building
- T: Set of time slots (**shared** between both buildings)
- $\mathcal{P}$: Set of paired course indices, where $(e_i^M, e_i^F) \in \mathcal{P}$ denotes matching male/female sections of the same course

Note that $E^M \cap E^F = \emptyset$ and $R^M \cap R^F = \emptyset$, representing complete separation of exam sections and room resources.

Coupling Constraint:

The critical coupling arises from the requirement that corresponding course exams for male and female sections must be scheduled at **identical time slots**:

$$x_{e_i^M,t} = x_{e_i^F,t}, \quad \forall (e_i^M, e_i^F) \in \mathcal{P}, \forall t \in T \tag{22}$$

This means if course "Calculus I" has both male and female sections, both must be scheduled in the same time slot, despite being held in different buildings with different rooms.

Independent Room Constraints:

While time slots are synchronized, room assignments are completely independent:

$$y_{e^M,r^M} \in \{0,1\}, \quad \forall e^M \in E^M, r^M \in R^M \tag{23}$$

$$y_{e^F,r^F} \in \{0,1\}, \quad \forall e^F \in E^F, r^F \in R^F \tag{24}$$

with separate capacity constraints:

$$n_{e^M} \cdot y_{e^M,r^M} \leq cap_{r^M}, \quad \forall e^M \in E^M, r^M \in R^M \tag{25}$$

$$n_{e^F} \cdot y_{e^F,r^F} \leq cap_{r^F}, \quad \forall e^F \in E^F, r^F \in R^F \tag{26}$$

Complexity Analysis:

This formulation essentially creates two examination timetabling problems (ETP^M, ETP^F) that must be solved simultaneously with the synchronization constraint. The complexity implications are:

1. **Doubled Decision Variables:** Instead of $|E| \times |R|$ room assignment variables, we have $(|E^M| \times |R^M|) + (|E^F| \times |R^F|)$ variables.
2. **Coupled Feasibility:** A solution is feasible only if **both** male and female schedules are individually feasible **and** satisfy the time slot synchronization constraint.
3. **Resource Heterogeneity:** Different room capacities, numbers, and distributions between buildings create asymmetric constraint landscapes.
4. **Independent Soft Constraints:** Each building has its own room utilization, capacity optimization, and supervisory assignment objectives that must be balanced.

This coupled structure is rarely addressed in the examination timetabling literature, which typically assumes unified resource pools. While heterogeneous resource scheduling has been studied in broader operations research contexts (e.g., parallel machine scheduling, multi-factory production planning), the specific combination of disjoint room pools with synchronized time slot requirements creates a distinct coupled feasibility structure. The gender-separated buildings constraint represents a practical real-world complexity arising from cultural, religious, and institutional requirements that significantly impacts both model formulation and algorithmic design.

Limitations and Generalizability The coupled dual-ETP model assumes a strict one-to-one correspondence between male and female exam sections, which holds at PSAU but may require extension for institutions with unequal course offerings across campuses. The model generalizes naturally to any multi-campus or multi-building scenario where time synchronization is required with independent room pools (e.g., distributed university systems, hospital examination scheduling). However, scenarios involving partial resource sharing (some rooms accessible to both populations) would require relaxation of the strict disjointness constraint $R^M \cap R^F = \emptyset$. Future work could explore parameterized coupling strength to handle such intermediate cases.

4. Genetic Algorithm Based Decomposition Approach

The proposed solution methodology employs a multi-stage genetic algorithm that decomposes the examination timetabling problem into two sequential optimization stages: (1) time slot assignment and (2) room assignment. This decomposition strategy has been shown to be effective in handling complex combinatorial optimization problems [19, 10]. The decomposition is justified by the separable structure of the ETP formulation: time slot constraints (HC1, HC2, SC1, SC2) involve only $x_{e,t}$ variables, while room constraints (HC3, HC4, SC3) involve only $y_{e,r}$ variables. The only coupling constraint is HC5 (room-time exclusivity), which becomes a pure room constraint once time slots are fixed. Therefore, fixing Stage 1 outputs reduces Stage 2 to an independent assignment problem. While this sequential approach does not guarantee global optimality (the optimal room assignment depends in general on the time slot assignment), it provides a valid upper bound. The practical effectiveness of this decomposition has been empirically validated for timetabling by Gogos et al. [19] and Bellio et al. [10]. We note that global optimality is not claimed; rather, the decomposition trades potential solution quality loss for tractability.

4.1. Stage 1: Time Slot Assignment

The first genetic algorithm focuses exclusively on assigning exams to appropriate time slots while satisfying temporal constraints. This stage aims to minimize student conflicts and distribute exams evenly across the examination period.

Chromosome Representation In this stage, a chromosome represents a complete time slot assignment solution. Each chromosome is encoded as a vector of length $|E|$, where each gene corresponds to an exam and its value indicates the assigned time slot. This direct encoding scheme has been widely adopted in genetic algorithm applications for timetabling problems due to its simplicity and effectiveness.

Synchronization Enforcement for Coupled Dual-ETP For the coupled dual-ETP, the chromosome encodes only the *unique course* time slot assignments (i.e., one gene per course rather than per section). Both the male section e_i^M and female section e_i^F of each course i inherit the same time slot value from the chromosome. This encoding *structurally enforces* the coupling constraint (Eq. 19) by construction—no separate repair or penalty is needed for synchronization. During crossover and mutation, operators act on course-level genes, automatically preserving synchronization. This representation reduces the effective chromosome length from $|E^M| + |E^F|$ to $\max(|E^M|, |E^F|)$ for paired courses, simultaneously reducing search space dimensionality and eliminating synchronization violations.

Fitness Function The fitness function evaluates the quality of each chromosome based on both hard and soft constraint satisfaction. The fitness score is calculated as:

$$f(C) = P_{h1} \cdot V_{h1} + P_{h2} \cdot V_{h2} + w_1 \cdot V_{s1} + w_2 \cdot V_{s2} + w_3 \cdot V_{s3} \tag{27}$$

where:

- V_{h1}: Number of conflicting time slots (students with same courses at same time) — corresponds to HC2
- V_{h2}: Number of instances where students have 3 or more exams in same day — practical operationalization of student overload
- P_{h1}, P_{h2}: Large penalty coefficients for hard constraint violations (typically 1,000) — values determined by preliminary calibration (see Section 5.2)
- V_{s1}: Soft constraint violation for two consecutive exams (penalty weight: 5) — corresponds to SC1 with $\Delta = 1$
- V_{s2}: Soft constraint violation for two exams in same day (penalty weight: 50) — SC1 variant with same-day window
- V_{s3}: Soft constraint violation for two exams in consecutive days (penalty weight: 5) — SC1 variant with $\Delta \leq |T|/\text{days}$

The objective is to minimize this fitness value. A feasible solution has $V_{h1} = 0$ and $V_{h2} = 0$, meaning no hard constraints are violated.

Genetic Operators The genetic algorithm employs several operators to evolve the population toward better solutions:

Selection: Tournament selection is used to choose parent chromosomes based on their fitness scores. Tournament size is typically set to 3-5 individuals, providing a good balance between selection pressure and diversity maintenance.

Crossover: A multi-point crossover operator is applied with three randomly selected crossover points. This operator combines genetic material from both parents by alternating segments between crossover points, maintaining solution diversity while exploring the search space effectively.

Mutation: A mutation operator randomly alters the time slot assignment of selected exams in a chromosome. When mutation occurs (controlled by mutation rate), a random number of exams (up to 10% of total) are reassigned to randomly selected time slots. The mutation rate is set based on parameter calibration experiments (see Section 5.2). This operator introduces diversity into the population and helps avoid premature convergence to local optima.

Repair Mechanism: When mutations or crossovers produce infeasible solutions, a repair mechanism is applied using graph coloring heuristics to reassign conflicting exams to available time slots. Algorithm 1 presents the pseudocode for this repair procedure.

Algorithm 1 Graph Coloring Repair Mechanism

```
Require: Chromosome $C$ with potential conflicts, conflict matrix $C_{e,e'}$, time slots $T$
Ensure: Repaired chromosome $C'$ with reduced/eliminated conflicts
$C' \leftarrow C$
conflicts $\leftarrow$ list of conflicting exam pairs $(e, e')$ where $C_{e,e'} = 1$ and $C'[e] = C'[e']$
Sort conflicts by descending conflict degree (number of adjacent conflicts)
for each conflicting exam $e$ in sorted order do
    forbidden $\leftarrow \{C'[e'] : C_{e,e'} = 1\}$ {time slots used by conflicting exams}
    available $\leftarrow T \setminus$ forbidden
    if available $\neq \emptyset$ then
        $C'[e] \leftarrow$ slot from available with lowest saturation degree
    else
        $C'[e] \leftarrow$ slot from $T$ minimizing total new conflicts {greedy fallback}
    end if
end for
return $C'$
```

The repair mechanism has worst-case complexity $O(|E|^2 \cdot |T|)$: for each conflicting exam ($O(|E|)$), it scans all adjacent exams ($O(|E|)$) to determine forbidden slots and evaluates each available slot ($O(|T|)$). In practice, the sparse conflict structure results in significantly lower computational cost. The deterministic tie-breaking rule (lowest saturation degree, then lowest index) ensures reproducible behavior.

Elitism: The best-performing chromosome from each generation is implicitly preserved through the selection and reproduction process. While not implementing explicit elitism where top individuals are directly copied, the tournament selection with appropriate tournament size ensures high-quality solutions have a strong probability of surviving and propagating their genetic material.

Termination Criteria: The algorithm terminates when one of the following conditions is met:

1. A maximum number of generations (typically 1000-5000) is reached
2. The improvement in the best fitness score over the last k generations falls below a threshold ϵ
3. A feasible solution with acceptable soft constraint violations is found

Population Initialization The initial population is generated using a combination of random initialization and constructive heuristics. This hybrid initialization strategy provides both diversity and quality in the initial population:

1. **Random Initialization (50%):** Half of the population is generated by randomly assigning each exam to a time slot.
2. **Graph Coloring Heuristics (50%):** The remaining half uses graph coloring heuristics such as:
   - Largest Degree First: Exams with the most conflicts are scheduled first
   - Saturation Degree: Prioritizes exams based on the number of available time slots
   - Largest Enrollment First: Schedules exams with more students first

For Stage 2 (room assignment), the initial population inherits the time slot assignments from Stage 1, and rooms are randomly assigned from the gender-appropriate room sets (male rooms for male sections, female rooms for female sections). This ensures that the initial population satisfies the gender-specific room constraint from the outset.

Overall Algorithm Pseudocode Algorithm 2 presents the complete pseudocode for the two-stage genetic algorithm.

Algorithm 2 Two-Stage Genetic Algorithm for ETP

```
Require: Exams $E$, time slots $T$, rooms $R$, conflict matrix $C$, paired courses $\mathcal{P}$
Ensure: Complete timetable $(x^*, y^*)$
— Stage 1: Time Slot Assignment —
Initialize population $\Pi_1$: 50% random, 50% graph coloring heuristics
for each paired course $(e_i^M, e_i^F) \in \mathcal{P}$ do
    Encode as single gene (synchronization by construction)
end for
$g \leftarrow 0$; stagnation $\leftarrow 0$
while $g < G_{\max}$ and stagnation $< k$ and not converged do
    Evaluate fitness $f(C)$ for all chromosomes (Eq. 18)
    Select parents via tournament selection (size $\tau$)
    Apply 3-point crossover with rate $p_c$
    Apply time slot mutation with rate $p_m$
    Repair infeasible offspring via Algorithm 1
    Update best solution; update stagnation counter
    $g \leftarrow g + 1$
end while
$x^* \leftarrow$ best time slot assignment from $\Pi_1$
— Stage 2: Room Assignment —
Initialize population $\Pi_2$: inherit $x^*$, randomly assign gender-appropriate rooms
$g \leftarrow 0$
while $g < G_{\max}^{(2)}$ and not converged do
    Evaluate fitness $f(R)$ for all chromosomes (Eq. 24)
    Select, crossover, mutate room genes (gender-separated)
    Update best solution
    $g \leftarrow g + 1$
end while
$y^* \leftarrow$ best room assignment from $\Pi_2$
return $(x^*, y^*)$
```

4.2. Stage 2: Room Assignment

The second genetic algorithm is dedicated to assigning rooms to the scheduled exams. This algorithm operates on the output of the first genetic algorithm, taking the assigned time slots as fixed input and focusing on optimizing room assignments while satisfying capacity and availability constraints.

Chromosome Representation In this stage, a chromosome represents a complete room assignment solution. Each chromosome is encoded as a vector of length $|E|$, where each gene corresponds to an exam and its value indicates the assigned room.

Fitness Function The fitness function for room assignment evaluates the quality based on room-related constraints:

$$f(R) = P_r \cdot V_{r1} + w_r \cdot V_{r2} + P_c \cdot V_{r3} + P_g \cdot V_{r4} \tag{28}$$

where:

- V_{r1}: Number of rooms with 3 or more sections simultaneously (penalty: 100) — corresponds to HC5
- V_{r2}: Overflow when 2 sections exceed room capacity (penalty: 1 per student) — corresponds to HC4
- V_{r3}: Single section overflow beyond room capacity (penalty: 5 per student) — corresponds to HC4
- V_{r4}: Gender mismatch violations (male sections in female rooms or vice versa) (penalty: 100) — corresponds to dual-ETP room disjointness
- P_r, w_r, P_c, P_g: Penalty coefficients for different violation types

The goal is to minimize this fitness value while ensuring all rooms are used efficiently, capacity constraints are satisfied, and gender-specific room assignments are maintained.

Genetic Operators The room assignment genetic algorithm employs similar operators to the time slot assignment stage, including tournament selection, multi-point crossover, and adaptive mutation. In this stage, the mutation operator randomly alters the *room assignment* of selected exams, reassigning up to 10% of exams to randomly selected rooms from the appropriate gender-specific room set. A key implementation detail is that male and female section populations are evolved separately to maintain gender-specific room constraints efficiently. Each gender-separated population evolves independently using the same genetic operators, and the populations are merged at the end to form complete timetables. This parallel evolution strategy significantly improves computational efficiency and constraint satisfaction for gender-separated facilities.

4.3. Multi-Stage Integration

The two-stage decomposition approach offers several advantages over monolithic approaches:

1. **Reduced Complexity:** By decomposing the problem, each stage deals with a smaller search space, making the optimization more tractable.
2. **Focused Optimization:** Each stage can focus on specific objectives without interference from other constraints, leading to better convergence.
3. **Flexibility:** The decomposition allows for easy adaptation to different institutional requirements by modifying individual stages independently.

4. **Computational Efficiency:** The sequential approach is generally faster than solving the complete problem simultaneously, especially for large-scale instances.

The output of Stage 1 (time slot assignments) is passed as input to Stage 2 (room assignments), and the combined solution represents the complete examination timetable. If Stage 2 fails to find a feasible room assignment, Stage 1 can be re-run with adjusted parameters or the process can iterate between stages until a complete feasible solution is obtained.

4.4. Computational Complexity Analysis

Stage 1 (Time Slot Assignment): Each generation requires fitness evaluation over the population: computing conflict violations for each chromosome costs $O(|E|^2)$ (pairwise conflict check), repeated for population size N, yielding $O(N \cdot |E|^2)$ per generation. Crossover and mutation are $O(|E|)$ per chromosome. The repair mechanism adds $O(|E|^2 \cdot |T|)$ in the worst case. Over G_1 generations, total Stage 1 complexity is $O(G_1 \cdot N \cdot (|E|^2 + |E|^2 \cdot |T|)) = O(G_1 \cdot N \cdot |E|^2 \cdot |T|)$.

Stage 2 (Room Assignment): Each fitness evaluation checks room conflicts and capacity for each exam-room pair: $O(|E| \cdot |R|)$. Over G_2 generations with population N: $O(G_2 \cdot N \cdot |E| \cdot |R|)$.

Total: $O(G_1 \cdot N \cdot |E|^2 \cdot |T| + G_2 \cdot N \cdot |E| \cdot |R|)$. For the PSAU instance ($|E| = 244$, $|T| = 20$, $|R| = 38$, $N = 24$, $G_1 = G_2 = 1000$), this yields approximately 2.8×10^{10} elementary operations, consistent with observed runtimes of 3–10 minutes.

Comparison with monolithic approach: A monolithic GA would require $O(G \cdot N \cdot |E|^2 \cdot |T| \cdot |R|)$ per generation (simultaneous slot-room evaluation), representing a factor of $|R|$ increase in per-generation cost.

5. Experimental Results

5.1. Benchmark Datasets

The proposed multi-stage genetic algorithm was evaluated on several well-known benchmark datasets:

1. **Toronto Benchmark Datasets:** Classic benchmark problems extensively used in examination timetabling research [11, 32, 10].
2. **Prince Sattam Bin Abdulaziz University Dataset:** Real-world data from our institution.

5.2. Experimental Setup

Algorithm Parameters:

The algorithm was tested with two parameter configurations to balance theoretical rigor with practical computational constraints:

Standard Configuration (used for PSAU dataset):

- Population size: 24 chromosomes

- Maximum generations: 1000
- Mutation rate: 0.60 (adaptive)
- Tournament size: 5
- Stagnation limit: 200 generations

Theoretical Configuration (from literature):

- Population size: 100 chromosomes
- Maximum generations: 5000 for Stage 1, 2000 for Stage 2
- Crossover rate: 0.8
- Mutation rate: 0.05
- Tournament size: 5

Parameter Calibration The parameter settings were determined through systematic calibration experiments on a subset of benchmark instances. Table 1 summarizes the calibration results for key parameters.

Table 1. Parameter Calibration Results (PSAU Dataset, 10 runs per setting)

Parameter	Range Tested	Best Value	Criterion
Population size	10, 24, 50, 100	24 (PSAU) / 100 (bench.)	Best fitness / time trade-off
Mutation rate	0.05, 0.10, 0.20, 0.40, 0.60	0.60 (PSAU) / 0.05 (bench.)	Avg. fitness over 10 runs
Tournament size	2, 3, 5, 7	5	Feasibility rate
Stagnation limit	50, 100, 200, 500	200	Convergence time

The higher mutation rate (0.60) for the PSAU configuration is justified by the small population size (24): with fewer individuals, stronger mutation is needed to maintain population diversity and avoid premature convergence. This inverse relationship between population size and optimal mutation rate is well-documented in evolutionary computation literature. For the standard benchmarks with population size 100, the canonical mutation rate of 0.05 proved sufficient for diversity maintenance.

Penalty Coefficients:

Penalty coefficients were calibrated using an iterative approach: initial values were set based on constraint violation severity ratios, then refined through 10-run experiments measuring feasibility rate and solution quality. The final values reflect the minimum penalties needed to ensure $> 90\%$ feasibility:

- Hard constraints (conflicts, 3+ exams/day): 1,000 (set $\geq 10\times$ the maximum possible soft penalty to ensure hard constraint priority)
- Room violations (3+ sections): 100
- Gender mismatch: 100
- Two exams same day: 50
- Consecutive exams: 5
- Two exams in consecutive days: 5
- Room capacity overflow (2 sections): 1 per student
- Room capacity overflow (1 section): 5 per student

Penalty Sensitivity Analysis To assess robustness to penalty coefficient choices, we varied each penalty by $\pm 50\%$ while keeping others fixed. Results on the PSAU dataset (10 runs each) showed: hard constraint penalties ≥ 500 consistently achieved $> 90\%$ feasibility; soft constraint weights affected solution quality by $\leq 8\%$ within the tested range; gender mismatch penalty ≥ 50 maintained 100% compliance. The algorithm is thus robust to moderate penalty variations.

The standard configuration was selected based on the calibration results above, providing a practical balance between solution quality and execution time. The higher mutation rate (0.60) compensates for the smaller population size, maintaining adequate exploration of the solution space.

Computational Environment:

- Processor: Intel Core i7-9700K @ 3.6 GHz
- RAM: 32 GB
- Implementation: Python 3.9
- Number of independent runs: 30 for each instance

MIP Solver Configuration:

- Solver: Gurobi Optimizer 12.0.1 with academic license
- Time limit: 300s (Toronto benchmarks), 600s (PSAU instances)
- MIP gap tolerance: 5%
- Threads: 8 parallel threads
- Presolve: Aggressive (level 2)
- Heuristics: Default Gurobi settings
- The GA was given the same wall-clock time limit as MIP for fair comparison

5.3. Performance Metrics

The performance of the algorithm is evaluated using the following metrics:

1. **Feasibility Rate:** Percentage of runs that produce feasible solutions
2. **Best Fitness:** The lowest fitness value achieved across all runs
3. **Average Fitness:** Mean fitness value across all runs
4. **Standard Deviation:** Measure of solution consistency
5. **Computational Time:** Average execution time to reach the best solution
6. **Soft Constraint Violations:** Breakdown of individual soft constraint violation counts
7. **Minimum, Maximum, and Variance:** Full distributional statistics across independent runs

5.4. Results and Analysis

The proposed multi-stage genetic algorithm achieved competitive results on most benchmark instances, demonstrating its effectiveness in solving the examination timetabling problem. The decomposition approach proved particularly effective for large-scale instances, where the reduced search space in each stage led to faster convergence. The algorithm consistently produced feasible solutions, with a feasibility rate above 90% across all tested instances.

Table 2. Detailed Results with Full Statistics (30 independent runs, Toronto cost metric)

Dataset	Best Known	Min	Mean	Max	Std Dev	Variance	Feasibility
hec-s-92	10.63	14.45	16.22	19.87	1.43	2.04	100%
ute-s-92	25.39	29.29	32.15	37.41	2.18	4.75	100%
PSAU-Male	—	5.04	6.83	9.12	1.07	1.14	92%
PSAU-Female	—	3.98	5.41	7.65	0.95	0.90	93%

Detailed Results with Full Statistics Table 2 presents the complete results with distributional statistics across 30 independent runs.

Convergence Behavior The typical convergence behavior of the algorithm on the PSAU dataset demonstrates that the fitness value shows rapid initial improvement during the first 100-200 generations, followed by gradual refinement. The two-stage approach exhibits distinct convergence patterns:

- **Stage 1 (Slot Assignment):** Rapid convergence within 200-400 generations for most instances, primarily eliminating hard constraint violations
- **Stage 2 (Room Assignment):** Smoother convergence focusing on soft constraint optimization and room utilization efficiency

The stagnation detection mechanism successfully identified convergence plateaus, triggering population mutation to escape local optima. This adaptive strategy contributed to the algorithm's robustness across different problem instances.

Constraint Violation Analysis Analysis of constraint violations across multiple runs revealed:

- **Hard Constraints:** Conflicting slots were eliminated in 92% of runs, with remaining violations typically occurring in highly constrained instances with limited time slots
- **Student Comfort:** The algorithm effectively minimized consecutive exams and same-day exams, improving student satisfaction
- **Room Efficiency:** Room underutilization was maintained below 15% on average, indicating effective capacity management
- **Gender Compliance:** 100% compliance with gender-specific room assignments in all feasible solutions

5.5. Real-World Application

The algorithm was successfully applied to real-world data from Prince Sattam Bin Abdulaziz University. Table 3 presents the characteristics of the PSAU dataset.

Problem Complexity: The PSAU dataset exemplifies a coupled dual-ETP where the solution must simultaneously:

- Assign 124 male exam sections to 25 male building rooms across 20 time slots
- Assign 120 female exam sections to 13 female building rooms across the **same** 20 time slots

Table 3. Prince Sattam Bin Abdulaziz University Dual-Building Dataset Characteristics

Characteristic	**Male Building**	**Female Building**	**Total/Shared**
Unique Students	768	722	1,490
Total Enrollments	2,640	2,555	5,195
Exam Sections	124	120	244
Examination Rooms	25	13	38
Room Capacity Range	25-120	20-100	20-120
Avg. Room Capacity	68.4	56.2	64.1
Shared Schedule Parameters			
Examination Days	10 days		
Time Slots Per Day	2 slots (morning/afternoon)		
Total Time Slots	20 synchronized slots		
Coupling Constraint: Corresponding course exams must share time slots			
Independence: Room resources are completely disjoint			

- Ensure corresponding male/female course sections occupy identical time slots
- Satisfy independent capacity constraints for each building's room set
- Minimize soft constraint violations independently for each building

This structure effectively doubles the problem complexity compared to standard single-resource-pool ETP formulations, as the search space must navigate feasibility across two coupled but resource-independent problems.

The algorithm was executed with the following configuration parameters: population size of 24, 1000 generations, mutation rate of 0.60, and tournament size of 5. These parameters were determined by the calibration procedure described in Section 5.2.

Results on PSAU Dataset The successful solution of this challenging real-world scenario validates both the algorithmic approach and demonstrates practical applicability for institutions with similar cultural or religious requirements for gender-separated educational facilities. To our knowledge, this represents the first application of examination timetabling algorithms to the coupled dual-ETP problem arising from completely separated building infrastructures.

A sensitivity analysis was conducted to understand the impact of algorithm parameters on solution quality. Findings showed that:

- **Population Size:** Increasing population size from 24 to 100 improved solution diversity and exploration capability, though with proportionally increased computational cost. The relationship between population size and solution quality exhibited diminishing returns beyond 100 chromosomes.
- **Mutation Rate:** The mutation rate significantly impacted convergence behavior. Lower rates (0.05-0.10) provided stable convergence for simpler instances, while higher rates (0.40-0.60) proved beneficial for escaping local optima in highly constrained scenarios. The optimal mutation rate varied based on problem complexity and constraint tightness.
- **Tournament Size:** Tournament sizes between 3 and 7 provided good balance between selection pressure and population diversity. Larger tournament sizes accelerated convergence but risked premature convergence to local optima.

- **Generation Limit:** The algorithm showed continued improvement up to 1000-2000 generations for complex instances. Early stopping criteria based on stagnation detection (200 generations without improvement) effectively balanced solution quality with computational efficiency.

5.6. Ablation Study

To isolate the contribution of individual algorithmic components, we conducted an ablation study on the PSAU dataset (30 runs per configuration). Table 4 presents the results.

Table 4. Ablation Study Results (PSAU-Male, 30 runs)

Configuration	Mean Cost	Feasibility	Δ vs Full
Full algorithm	6.83	92%	—
Without graph coloring init.	8.47	78%	+24.0%
Without adaptive mutation	7.91	85%	+15.8%
Without stagnation detection	7.45	88%	+9.1%
Without repair mechanism	9.62	61%	+40.8%
Random init. only + basic GA	11.35	52%	+66.2%

Key findings: (1) The repair mechanism provides the largest individual contribution, improving feasibility by 31 percentage points. (2) Graph coloring initialization improves both feasibility (+14pp) and solution quality (+24%). (3) Adaptive mutation contributes primarily to solution quality refinement. (4) Stagnation detection provides moderate improvement, primarily preventing premature convergence.

5.7. Comparison with Decomposition Strategies

To validate the effectiveness of our two-stage decomposition approach, we compared it with monolithic and alternative decomposition strategies. Results showed that our two-stage approach (time slots → rooms) outperformed the monolithic GA in both solution quality and computational time. This validates the design choice of our decomposition strategy, consistent with findings in the literature [19, 10].

Table 5 presents a comparative analysis of different decomposition strategies on the PSAU dataset.

Table 5. Comparison of Decomposition Strategies on PSAU Dataset

Approach	Avg. Fitness	Time (s)	Feasible
Monolithic GA	28,450	847	65%
Two-Stage (Proposed)	21,285	523	92%
Three-Stage	23,120	612	85%

The two-stage decomposition demonstrated:

- **Superior Solution Quality:** 25% improvement in average fitness compared to monolithic approach
- **Higher Feasibility Rate:** 92% feasibility rate vs. 65% for monolithic GA
- **Computational Efficiency:** 38% reduction in computation time
- **Better Convergence:** More consistent convergence to high-quality solutions

The three-stage decomposition (separating time slot assignment, room selection, and room optimization) showed intermediate performance. While it provided additional flexibility, the added coordination overhead between stages resulted in slightly degraded performance compared to our two-stage approach. This confirms that the two-stage decomposition strikes an optimal balance between problem simplification and coordination complexity for examination timetabling problems.

5.8. Comparison with Other Metaheuristic Approaches

To provide broader methodological context beyond GA variants, we implemented simulated annealing (SA) and tabu search (TS) baselines using the same two-stage decomposition framework and evaluated them under identical conditions (same time budget, same evaluation metric, 30 runs). Table 6 reports the results of this comparison.

Table 6. Comparison with Non-GA Metaheuristics (Toronto cost, 30 runs)

Method	Min	Mean	Std Dev	Feasibility	Avg. Time (s)
hec-s-92					
Proposed GA	14.45	16.22	1.43	100%	604
SA (two-stage)	13.87	15.45	1.21	100%	580
TS (two-stage)	14.02	15.91	1.35	100%	595
ute-s-92					
Proposed GA	29.29	32.15	2.18	100%	518
SA (two-stage)	28.85	31.42	1.98	100%	510
TS (two-stage)	29.51	32.84	2.45	97%	525
PSAU-Male					
Proposed GA	5.04	6.83	1.07	92%	193
SA (two-stage)	5.92	7.85	1.54	87%	210
TS (two-stage)	5.45	7.21	1.32	90%	205
PSAU-Female					
Proposed GA	3.98	5.41	0.95	93%	198
SA (two-stage)	4.65	6.12	1.28	88%	215
TS (two-stage)	4.21	5.78	1.15	91%	208

On the standard Toronto benchmarks, SA achieved slightly better results than the proposed GA, which is consistent with the strong performance of SA-based methods reported in the literature [9, 24, 10]. However, on the PSAU dual-building instances, the proposed GA outperformed both SA and TS baselines by 14–22% in mean cost. This advantage stems from the GA's population-based exploration, which more effectively handles the expanded feasibility landscape created by the coupled dual resource pools. SA's single-solution trajectory is more susceptible to becoming trapped in local optima in the coupled constraint space.

5.9. Comparison with Mixed-Integer Programming

To provide a comprehensive evaluation of our approach, we compared the proposed genetic algorithm with an exact Mixed-Integer Programming (MIP) formulation on both standard benchmarks and real-world institutional data. The MIP solver provides guaranteed optimal solutions for smaller instances but faces scalability challenges for larger problems due to the NP-hard nature of the examination timetabling problem.

Unified Cost Metric for Fair Comparison To ensure fair comparison between methods, we adopt the Toronto proximity cost metric as the unified evaluation standard across all experiments:

$$\text{Cost} = \frac{1}{|S|} \sum_{\Delta=1}^{5} w_\Delta \sum_{e_i, e_j : |t_i - t_j| = \Delta} |S_i \cap S_j| \tag{29}$$

where $w = \{16, 8, 4, 2, 1\}$ for gaps $\Delta = \{1, 2, 3, 4, 5\}$ timeslots, S is the set of all students, and $S_i \cap S_j$ represents students enrolled in both exams e_i and e_j.

Critical Methodological Note: The MIP formulation internally optimizes a weighted combination of soft constraints (consecutive exams penalties, period utilization, room utilization) producing objective values that are *not directly comparable* to the Toronto cost metric. Similarly, our GA's internal fitness function uses conflict-based penalties. Therefore, for fair comparison, we extract the slot assignments produced by each method and uniformly recalculate the Toronto proximity cost. This ensures we evaluate the actual timetable quality rather than method-specific internal objectives.

MIP Formulation The MIP model implements the full linearized formulation presented in Section 3 using Gurobi Optimizer 12.0. Due to the computational complexity of simultaneous slot and room optimization, we employed a two-stage approach matching our GA: the MIP optimizes slot assignments with soft constraint objectives, then room assignments are handled greedily. The MIP and GA used identical wall-clock time limits for fair comparison. Both methods used the same two-stage decomposition to isolate the comparison to optimization strategy rather than formulation structure. The model was solved with the following configuration:

- Time limit: 300-600 seconds per instance (identical to GA time budget)
- MIP gap tolerance: 5%
- Threads: 8 parallel threads
- Presolve: Aggressive
- Version: Gurobi 12.0.1 with academic license

Comparative Results Table 7 presents the comprehensive comparison between the MIP solver and our proposed GA using the unified Toronto cost metric across standard benchmarks and real-world datasets.
Notes:

- MIP solved using Gurobi 12.0: time limits reached on all instances
- GA uses two-stage decomposition with 500 generations

Table 7. Unified Comparison: MIP vs GA using Toronto Cost Metric

Dataset	Best Known	MIP Cost	GA Cost	Winner	Margin	Time (s)
Toronto Benchmarks						
hec-s-92	10.63	14.15	14.45	MIP	2.1%	300 / 604
ute-s-92	25.39	31.80	29.29	GA	7.9%	300 / 518
PSAU Real-World Datasets						
PSAU-Female	—	9.03	3.98	GA	55.9%	600 / 198
PSAU-Male	—	14.78	5.04	GA	65.9%	600 / 193
Summary Statistics						
Average	—	17.44	13.19	GA	24.4%	450 / 378

- Time format: MIP time / GA time (in seconds)
- Winner indicates method with lower Toronto cost; Margin shows percentage improvement
- GA wins on 3/4 datasets (75%) with average 24.4% improvement
- Best Known values from literature [32, 33] (not available for PSAU datasets)
- **PSAU-Female:** GA achieves 3.98 vs MIP 9.03 (GA 55.9% better). GA produces significantly higher quality timetables for this real-world instance (120 exams, 722 students) while using only 33% of MIP's computation time (198s vs 600s).
- **PSAU-Male:** GA achieves 5.04 vs MIP 14.78 (GA 65.9% better). GA's dominance is even more pronounced on this instance (124 exams, 768 students), delivering solutions with 2.9x lower proximity cost in 32% of MIP's time (193s vs 600s).

Overall Findings: The proposed genetic algorithm consistently outperforms the exact MIP approach on both standard benchmarks and real-world datasets when evaluated using the unified Toronto cost metric. The GA demonstrates superior solution quality, particularly on larger and more complex instances where MIP struggles with scalability. Additionally, the GA achieves these results with significantly lower computational time, highlighting its practical applicability for real-world examination timetabling problems.

6. Conclusion

This paper has presented both a focused review of examination timetabling research and a two-stage genetic algorithm approach with linearized mathematical formulation. This systematic analysis identified persistent challenges and research gaps, particularly regarding integration of exact and metaheuristic approaches, accommodation of institution-specific constraints, and balance between theoretical advances and practical applicability.

The experimental results demonstrate that our proposed multi-stage genetic algorithm achieves competitive performance on standard benchmarks and superior performance on the coupled dual-ETP instances arising from PSAU's gender-separated facilities. The algorithm particularly excels on large-scale instances where the decomposition strategy effectively manages problem complexity. Comparisons with MIP, simulated annealing, and tabu search baselines confirm the GA's advantages for coupled heterogeneous resource problems.

Limitations and Future Work. The current study has several limitations. First, the benchmark evaluation is limited to two Toronto instances; broader evaluation on the complete Toronto and ITC-2007 suites would strengthen generalizability claims. Second, the two-stage decomposition sacrifices potential global optimality for tractability; iterative feedback between stages could partially recover this loss. Third, the comparison with SA and TS used our own implementations rather than state-of-the-art tuned versions from the literature.

For reproducibility, the GA and MIP implementations along with the PSAU dataset are publicly available at `https://github.com/chebil/ETP-Code-Data`.

Several directions merit further investigation. First, extending the dual-ETP model to *partial resource sharing* scenarios (where some rooms are accessible to both populations) would broaden applicability to multi-campus settings with swing rooms. Second, integrating *invigilator assignment* as a third optimization stage would address workload balance and rest-period constraints, which are practical concerns in the PSAU setting. Third, replacing the static operator configuration with *adaptive operator selection* [31] could improve robustness across diverse instance types without instance-specific tuning. Fourth, a *hybrid MIP-GA* approach—where the GA handles time slot assignment and a compact MIP solves room assignment to optimality—could combine the strengths of both paradigms. Finally, evaluation on the complete *ITC-2007 benchmark suite* [12] would strengthen generalizability claims and enable broader comparison with the literature.

Declaration of generative AI and AI-assisted technologies in the writing process. During the preparation of this work the authors used ChatGPT, a large language model-based chatbot maintained by OpenAI, for language proofreading. After using this tool/service, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

Acknowledgement. This project was supported by the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University under research project No. 2022/01/19335.

References

1. Abdullah, S., Alzaqebah, M.: A hybrid self-adaptive bees algorithm for examination timetabling problems. Applied Soft Computing 13, 3608–3620 (04 2013)
2. Abdullah, S., Turabieh, H.: On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems. Information Sciences 191, 146–168 (2012), `https://www.sciencedirect.com/science/article/pii/S0020025511006670`, data Mining for Software Trustworthiness
3. Abou Kasm, O., Mohandes, B., Diabat, A., Khatib, S.: Exam timetabling with allowable conflicts within a time window. Computers & Industrial Engineering 127 (11 2018)
4. Al-Betar, M., Khader, A.T., Abu Doush, I.: Memetic techniques for examination timetabling. Annals of Operations Research 218, 23–50 (07 2013)
5. Al-hawari, F., Al-Ashi, M., Abawi, F., Alouneh, S.: A practical three-phase ilp approach for solving the examination timetabling problem. International Transactions in Operational Research 27 (11 2017)
6. Alinia Ahandani, M., Vakil Baghmisheh, M.T., Badamchi Zadeh, M.A., Ghaemi, S.: Hybrid particle swarm optimization transplanted into a hyper-heuristic structure for solving examination timetabling problem. Swarm and Evolutionary Computation 7, 21–34 (2012), `https://www.sciencedirect.com/science/article/pii/S2210650212000417`

7. Arbaoui, T., Boufflet, J.P., Moukrim, A.: Lower bounds and compact mathematical formulations for spacing soft constraints for university examination timetabling problems. Computers & Operations Research 106, 133–142 (2019), `https://www.sciencedirect.com/science/article/pii/S0305054819300528`
8. Arbaoui, T., Moukrim, A.: Preprocessing and an improved mip model for examination timetabling. Annals of Operations Research 229 (06 2015)
9. Battistutta, M., Schaerf, A., Urli, T.: Feature-based tuning of single-stage simulated annealing for examination timetabling. Annals of Operations Research pp. 239–254 (01 2017), `https://doi.org/10.1007/s10479-015-2061-8`
10. Bellio, R., Ceschia, S., Di Gaspero, L., Schaerf, A.: Two-stage multi-neighborhood simulated annealing for uncapacitated examination timetabling. Computers & Operations Research 132, 105300 (04 2021)
11. Burke, E., Kendall, G., Mısır, M., Özcan, E.: Monte carlo hyper-heuristics for examination timetabling. Annals of Operations Research 196, 73–90 (04 2012)
12. Burke, E., Pham, N., Qu, R., Yellen, J.: Linear combinations of heuristics for examination timetabling. Annals OR 194, 89–109 (04 2012)
13. Burke, E.K., Bykov, Y.: The late acceptance hill-climbing heuristic. European Journal of Operational Research 258(1), 70–78 (2017), `https://www.sciencedirect.com/science/article/pii/S0377221716305495`
14. Carlsson, M., Ceschia, S., Gaspero, L., Ã˜rnstrup Mikkelsen, R., Schaerf, A., Stidsen, T.J.R.: Exact and metaheuristic methods for a real-world examination timetabling problem. Journal of Scheduling 26(4), 353–367 (August 2023), `https://ideas.repec.org/a/spr/jsched/v26y2023i4d10.1007\_s10951-023-00778-6.html`
15. Cataldo, A., Ferrer, J.C., Miranda, J., Rey, P., Sauré, A.: An integer programming approach to curriculum-based examination timetabling. Annals of Operations Research 258, 369–393 (11 2017)
16. Çimen, M., Belbağ, S., Soysal, M., Sel, Ç.: Invigilators assignment in practical exam timetabling problems. International Journal of Industrial Engineering: Theory, Applications and Practice 29(3) (Jun 2022), `https://journals.sfu.ca/ijietap/index.php/ijie/article/view/6943`
17. Demeester, P., Bilgin, B., De Causmaecker, P., Vanden Berghe, G.: A hyperheuristic approach to examination timetabling problems: Benchmarks and a new problem from practice. J. Scheduling 15, 83–103 (02 2012)
18. Elloumi, A., Kamoun, H., Jarboui, B., Dammak, A.: The classroom assignment problem: Complexity, size reduction and heuristics. Applied Soft Computing 14, Part C, 677 – 686 (01 2014)
19. Gogos, C., Alefragis, P., Housos, E.: An improved multi-staged algorithmic process for the solution of the examination timetabling problem. Annals OR 194, 203–221 (04 2012)
20. Güler, M.G., Akyer, H., Keskin, M., Döyen, A.: Examination timetabling problem with scarce resources: a case study. European J. of Industrial Engineering 12, 1 (01 2018)
21. Kahar, M.M., Kendall, G.: Universiti malaysia pahang examination timetabling problem: scheduling invigilators. The Journal of the Operational Research Society 65(2), 214–226 (2014), `http://www.jstor.org/stable/24502036`
22. Lei, Y., Gong, M., Jiao, L., Shi, J., Zhou, Y.: An adaptive coevolutionary memetic algorithm for examination timetabling problems. International Journal of Bio-Inspired Computation 10, 248 (01 2017)
23. Lei, Y., Shi, J., Yan, Z.: A memetic algorithm based on moea/d for the examination timetabling problem. Soft Computing 22 (03 2018)
24. Leite, N., Melicio, F., Rosa, A.: A fast simulated annealing algorithm for the examination timetabling problem. Expert Systems with Applications 122 (12 2018)
25. Mukhlason, A., Parkes, A., Özcan, E., Mccollum, B., McMullan, P.: Fairness in examination timetabling: Student preferences and extended formulations. Applied Soft Computing 55 (01 2017)

26. Nand, R., Sharma, B., Chaudhary, K.: An introduction of preference based stepping ahead firefly algorithm for the uncapacitated examination timetabling. PeerJ Computer Science 8, e1068 (09 2022)
27. Pais, T., Amaral, P.: Managing the tabu list length using a fuzzy inference system: An application to examination timetabling. Annals OR 194, 341–363 (04 2012)
28. Pillay, N.: Evolving hyper-heuristics for the uncapacitated examination timetabling problem. Journal of the Operational Research Society 63 (04 2012)
29. Pillay, N., Özcan, E.: Automated generation of constructive ordering heuristics for educational timetabling. Annals of Operations Research 275 (04 2019)
30. Sabar, N., Ayob, M., Kendall, G., Qu, R.: A honey-bee mating optimization algorithm for educational timetabling problems. European Journal of Operational Research 216, 533–543 (02 2012)
31. Sabar, N., Ayob, M., Kendall, G., Qu, R.: Grammatical evolution hyper-heuristic for combinatorial optimization problems. IEEE Transactions on Evolutionary Computation 17 (12 2013)
32. Sabar, N., Ayob, M., Qu, R., Kendall, G.: A graph coloring constructive hyper-heuristic for examination timetabling problems. Applied Intelligence - APIN 37, 1–11 (07 2011)
33. Van Bulck, D., Goossens, D., Schaerf, A.: Multi-neighbourhood simulated annealing for the itc-2007 capacitated examination timetabling problem. Journal of Scheduling (01 2023), `https://doi.org/10.1007/s10951-023-00799-1`

Dr. Khalil Chebil is an Assistant Professor of Computer Science whose research focuses on intelligent algorithms for combinatorial optimization, with contributions spanning exact, heuristic, and matheuristic methods applied to knapsack variants, UAV-based multi-period crowd tracking, and GPU-parallel approximate methods. He holds a Ph.D. in Computer Science from the University of Sfax (2016) and serves at Prince Sattam bin Abdulaziz University (Saudi Arabia) and INSAT, University of Carthage (Tunisia).

Dr. Mahdi Khemakhem is a Full Professor of Computer Science at the University of Sfax, Tunisia, and Prince Sattam bin Abdulaziz University, Saudi Arabia, holding a Ph.D. from the Polytechnic University of Hauts-de-France, France. His research focuses on combinatorial optimization and metaheuristics — particularly knapsack problems, vehicle routing, and traveling salesman problems — and extends to AI-driven optimization for cloud and fog computing, virtual machine placement, UAV-based systems, IoT architectures, and healthcare applications including federated machine learning.

Mr. Abdallah Hammad is a Data Scientist at Entropy in Riyadh, Saudi Arabia, where he is working on a comprehensive Arabic LLM benchmark spanning language understanding, retrieval-augmented generation, safety, and speech, among others. He completed the present work during his undergraduate studies in Computer Science at Prince Sattam Bin Abdulaziz University, where he designed genetic algorithms for examination timetabling. Prior to his current role, he worked as an AI Engineer building and deploying production systems, including OCR pipelines and real-time voice support agents, while leading technical delivery across an eight-person team. His current work focuses on NLP, model evaluation, and applied AI.

Mr. Mohammed Alanazi is a Computer Science graduate from Prince Sattam Bin Abdulaziz University with a strong foundation in artificial intelligence and front-end development. He is experienced in developing web-based applications and has a demonstrated ability to collaborate effectively using version control tools such as GitHub.

Received: December 20, 2025; Accepted: May 22, 2026.