# A comparative analysis of classical and cooperative coevolutionary genetic algorithms for solving nurse scheduling problems

Maizatul Farhana Mohamad Nazri [a,1,*], Zeratul Izzah Mohd Yusoh [a,2], Halizah Basiron [a,3], Azlina Daud [b,4]

[a] Universiti Teknikal Malaysia Melaka (UTeM), Malaysia
[b] International Islamic University Malaysia (IIUM), Malaysia
[1] mfarhanamnazri@gmail.com; [2] zeratul@utem.edu.my; [3] halizah@utem.edu.my, [4] damia@iium.edu.my
* corresponding author

ARTICLE INFO

**Article history**
Received February 29, 2026
Revised April 10, 2026
Accepted April 10, 2026
Available online August 31, 2026

**Keywords**
Nurse Scheduling Problem
Cooperative Co-evolutionary
Genetic Algorithm
Evolutionary Optimization
Healthcare Scheduling

ABSTRACT

The Nurse Scheduling Problem (NSP) is a complex workforce planning task that involves assigning nurses to shifts while satisfying operational feasibility, legal regulations, and preference-based quality requirements. Classical Genetic Algorithms (GA) are widely applied to NSP but rely on monolithic optimization structures that evaluate all constraints within a single population, which can lead to constraint interference and reduced stability as problem realism increases. This study investigates a Cooperative Co-Evolutionary approach for NSP (Coop-NSP), which decomposes optimization into two interacting subpopulations corresponding to hard and soft constraints. Both subpopulations employ an identical nurse-by-day chromosome representation and evolve independently under specialized objectives, with cooperation introduced through contextual fitness evaluation. Experiments were conducted on a weekly NSP with a seven-day planning horizon and multiple nurse roles, evaluated for 15,000 generations across 30 independent runs under identical parameter settings. The Classical GA achieved a mean best penalty of 651.30 ± 37.90, with a minimum best penalty of 581.7, while Coop-NSP obtained a higher mean best penalty of 768.72 but achieved a lower minimum best penalty of 507.64. The Classical GA exhibited rapid convergence, whereas Coop-NSP demonstrated stepwise convergence with sustained population diversity. Although Coop-NSP incurred higher computational cost, its structured cooperative optimization framework provides a stable and extensible foundation for future integrated healthcare scheduling research.

© 2026 The Author(s).
This is an open access article under the CC–BY-SA license.

## 1. Introduction

Healthcare systems operate under increasing pressure due to rising service demand, workforce shortages, and the need to maintain high standards of safety and care quality [1], [2]. Effective workforce planning is therefore critical, as staffing decisions directly affect service continuity, patient safety, and staff well-being [1], [3]. Nurse scheduling represents a core operational task, requiring hospitals to assign nurses to shifts while complying with legal regulations, coverage requirements, and workforce policies [1], [2]. The Nurse Scheduling Problem (NSP) involves the simultaneous satisfaction of mandatory feasibility constraints and preference-driven quality considerations. Hard constraints such as staffing coverage, minimum working hours, rest requirements, and role-dependent rules must be strictly enforced to ensure operational viability, while soft constraints related to workload balance, fairness, and

individual preferences are increasingly emphasized to support staff satisfaction and long-term sustainability [1], [2], [4].

The interaction between these constraint types results in a highly coupled optimization problem, where improvements in one aspect of the schedule may negatively affect another [5], [6]. NSP is widely recognized as an NP-hard problem that cannot be solved optimally at realistic scales using exact methods alone [5]. Early studies employed mathematical programming and exact optimization techniques to model staffing coverage and regulatory constraints [7], [8]. While effective for small instances, these approaches scale poorly as problem size and constraint coupling increase. Consequently, heuristic and metaheuristic approaches have become dominant in NSP research, offering scalable alternatives capable of handling large solution spaces and complex constraint structures [5], [9].

Genetic Algorithms (GA) and their variants are among the most widely adopted metaheuristics for NSP. GA-based approaches typically encode a complete roster as a chromosome and evaluate solutions using penalty-based fitness functions that aggregate hard and soft constraint violations [10], [11], [12]. Numerous studies report that GA can generate feasible and high-quality schedules, particularly when enhanced with problem-specific operators, repair mechanisms, or hybrid local search strategies [13], [14], [15], [16], [17]. However, most GA-based NSP models rely on a single monolithic optimization structure, where feasibility and schedule quality are optimized simultaneously within one population [5], [6]. As constraint realism increases, this design can lead to constraint interference, sensitivity to penalty calibration, and reduced population diversity after feasibility is achieved [9], [18], [19].

This limitation arises because feasibility enforcement and schedule quality improvement are simultaneously driven within a single population, causing competing optimization pressures to interfere with each other. As a result, improvements in soft constraint satisfaction may lead to violations of hard constraints, while strict feasibility enforcement may restrict further refinement of schedule quality. This coupling reduces the stability of the evolutionary process and limits the algorithm's ability to maintain diverse yet feasible solutions, particularly under complex constraint structures. Beyond nurse rostering, healthcare scheduling research increasingly highlights interdependencies with other operational planning problems, including patient admission scheduling and operating room management [20]-[22].

Patient admission scheduling studies address uncertainty in arrivals, length of stay, and resource availability using mixed-integer programming, fuzzy decision models, evolutionary optimization, and robust heuristics [7], [8], [20]-[24]. Similarly, operating room and surgical scheduling research focuses on uncertainty in surgery durations, downstream ward capacity, and emergency disruptions, often employing robust optimization, hybrid metaheuristics, reinforcement learning, and decomposition-based approaches [22]-[26]. Recent works emphasize integrated scheduling across healthcare layers, demonstrating that decisions in one layer can significantly affect feasibility and performance in others [27]-[31].

In this context, Cooperative Coevolutionary Algorithms have emerged as a promising paradigm for complex optimization problems characterized by interacting subcomponents. Instead of evolving a single population to optimize all objectives simultaneously, cooperative coevolution decomposes the problem into multiple subpopulations that evolve under specialized objectives while cooperating through contextual fitness evaluation [32], [33]. Surveys and recent studies report successful applications of cooperative coevolution in large-scale optimization, multi-objective scheduling, and uncertainty-aware problems, particularly in workflow scheduling and distributed systems [32]-[36]. Nevertheless, its application to NSP remains relatively limited compared to classical GA-based approaches [5], [6], [13].

Compared to existing NSP approaches, most studies focus on enhancing monolithic optimization frameworks through hybridization, local search, or adaptive penalty mechanisms. While these methods improve solution quality and convergence speed, they generally do not address the underlying structural limitation of competing constraint objectives within a single population. In contrast, the proposed cooperative approach introduces a fundamentally different optimization structure by decoupling feasibility and quality objectives into interacting subpopulations, enabling more stable and modular optimization behavior. Based on the reviewed literature, three research gaps are identified. First, existing

NSP metaheuristics often struggle with constraint interference as realism and preference complexity increase under monolithic penalty-based optimization [5], [6], [18].

Although hybrid and memetic methods enhance local search capability and convergence speed, they typically operate within a single evolving population, where feasibility enforcement and solution refinement compete within the same optimization process. Second, integrated healthcare scheduling highlights the need for structured yet cooperative optimization algorithms capable of managing interdependent decision layers without destabilizing feasibility [20], [27], [29], [31]. Third, the convergence behavior and population-level stability of cooperative coevolution in NSP settings remain underexplored under controlled experimental conditions [32], [33], [34], [35], [36]. This study addresses these gaps by proposing and evaluating a cooperative coevolutionary approach for NSP that separates feasibility-driven and quality-driven optimization into interacting subpopulations while maintaining a unified solution representation. By employing contextual cooperative evaluation, the proposed approach enables controlled interaction between subpopulations through fitness evaluation without direct chromosome exchange, aiming to maintain feasibility, preserve structured diversity, and provide a foundation for future extensions toward integrated healthcare scheduling problems. The main novelty of this work lies in the contextual cooperative evaluation strategy, which allows independent optimization processes to interact without disrupting solution structure, thereby reducing constraint interference and improving optimization stability. This design also provides a structured and extensible foundation for future integration with other healthcare scheduling components, such as operating theatre and patient admission scheduling.

## 2. Method

### 2.1. Problem Formulation

This subsection presents the mathematical formulation of the NSP considered in this study. The problem definition is first introduced, followed by the hard constraints that ensure feasibility and the soft constraints that capture schedule quality. Finally, the objective function used in both the Classical GA and the proposed Coop-NSP approach is defined.

#### 2.1.1. Nurse Scheduling Mode and Schedule Representation

Let $N = \{1,2,3,\ldots n\}$ denote the set of nurses and $D = \{1,2,3\ldots d\}$ denote the set of days in the planning horizon starting from Monday to Sunday. Let $S = \{0,1,2,3,4\}$ represent the set of working shifts, where 0,1,2,3,4 indicated morning (AM), afternoon (PM), night (N), post-night (PN), and off shift, respectively. The post-night (PN) shift represents mandatory recovery time following a night shift and is included explicitly in the shift set to ensure compliance with hospital labour regulations and fatigue management policies. Each nurse may be assigned at most one shift per day, including the OFF shift, thereby preventing overlapping or conflicting duties. This assumption reflects standard rostering practice in hospital environments and simplifies feasibility checking at the gene level. The nursing workforce is categorized into three role-based subsets: sister nurses $N_s$, senior nurses $N_{sen}$, and healthcare assistants $N_{ha}$(HCA).

These role distinctions are essential for modelling coverage requirements and role-dependent constraints, as different nurse categories are subject to different operational rules and responsibilities. A binary decision variable is defined as: $x_{n,d,s} = 1$, if nurse $n$, assigned to shift $s$ on day $d$. While 0, if assigned otherwise. A complete schedule is represented as a three-dimensional nurse, day, shift assignment structure. In both the Classical GA and the proposed cooperative approach, a chromosome encodes a full weekly roster using a nurse-by-day representation, where each gene specifies the shift assigned to a nurse on a given day. Each gene takes one of the allowed shift values, resulting in a consistent and interpretable encoding that directly mirrors practical nurse rosters. Fig. 1 illustrates the chromosome structure for a single nurse over a seven-day planning horizon, which is extended across all nurses to form a complete weekly roster.

This nurse-by-day representation provides a compact yet expressive structure for capturing role-based coverage requirements, working-hour regulations, and night-to-post-night sequencing constraints. By

associating each gene with a single nurse-day assignment, constraint violations can be detected and penalized locally, supporting stable evolutionary search behavior.

| Chromosome | | | | | | | |
|---|---|---|---|---|---|---|---|
| Nurse | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 |
| N1 (Sister): | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} |
| N2 (Senior Nurse): | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} |
| N3 (Healthcare Assistant): | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} | {0...4} |

**Fig. 1.** Chromosome representation for a weekly schedule, where each gene denotes a nurse's shift assignment.

### 2.1.2. Hard and Soft Constraints

The Nurse Scheduling Problem considers both hard and soft constraints. Hard constraints represent mandatory requirements derived from hospital regulations and workforce policies, where any violation results in an infeasible schedule. Soft constraints reflect preference and quality aspects, such as workload balance and fairness, which improve schedule quality but are not strictly required. To prioritize feasibility, hard constraint violations are assigned significantly higher penalty weights than soft constraints. The complete set of constraints and their corresponding penalty values for both the Classical GA and Coop-NSP are summarized in Table 1.

**Table 1.** Hard Constraints Used in Classical and Cooperative NSP.

| ID | Constraint Description | Penalty |
|---|---|---|
| HC1 | Shift coverage: For each day, each shift, must have at least one senior nurse and one HCA | 700 |
| HC2 | OFF-day assignment: Sister must be OFF on both weekend days (Saturday and Sunday); Senior nurses and HCAs must have exactly one OFF day per week. | 600 |
| HC3 | Minimum weekly working hours: Each nurse must work at least 40 hours per week. | 300 |
| HC4 | Night to post-night: Any night shift (N) must be followed by a post-night (PN) shift on the following day. | 18 |
| SC1 | Sister shift preference: Sister nurses should work only AM or PM shifts on weekdays. | 30 |
| SC2 | Role balance constraint: Each shift best have minimum two Senior Nurse and two HCA. | 55 |
| SC3 | Preferred weekly working hours: Weekly working hours for each nurse should lie within the preferred range of 47–52 hours. | 20 |
| SC4 | Night shift distribution: Senior nurses and HCAs should be assigned at least one night shift per week. | 15 |

The penalty weights were determined through preliminary tuning to achieve a balance between feasibility enforcement and solution quality. During early generations, limited violations may occur to maintain diversity. However, the evolutionary process progressively favors solutions with fewer hard constraint violations, guiding the population toward feasible schedules before refining quality through soft constraints.

### 2.1.3. Objective Function

The Cooperative Nurse Scheduling Problem is formulated using a contextual cooperative objective function involving two interacting subpopulations. Both subpopulations encode complete weekly nurse schedules using the same chromosome representation. However, each subpopulation is optimized under a specialized objective aligned with either feasibility or schedule quality. The first subpopulation focuses on minimizing hard constraint violations and is evaluated using the hard constraint penalty function $HC(H_i)$, where $H_i$ denotes a schedule from the hard constraint subpopulation. The second subpopulation focuses on minimising soft constraint penalties and is evaluated using the soft constraint penalty function $SC(S_j)$, where $S_j$ denotes a schedule from the soft constraint subpopulation. Cooperation between the two subpopulations is incorporated through contextual fitness evaluation, as defined in the following equations.

$$F(H_i) = HC(H_i) + SC(S_{best}) \quad (1)$$

$$F(S_j) = HC(H_{best}) + SC(S_j) \quad (2)$$

where $H_{best}$ and $S_{best}$ denote the best-performing individuals from the hard and soft subpopulations, respectively, based on their specialised objectives. At each generation, the final realised schedule is obtained by pairing $H_{best}$ and $S_{best}$and evaluating the combined penalty:

$$F(x) \quad = HC(x) + SC(x) \tag{3}$$

This formulation enables feasibility and schedule quality to be optimized under specialized objectives while maintaining global consistency through cooperative interaction.

### 2.2. Proposed Algorithm

This section describes the optimization approaches used to solve the NSP, including a Classical Genetic Algorithm (GA) and the proposed Cooperative Nurse Scheduling (Coop-NSP). Both approaches employ identical genetic operators and parameter settings to ensure a fair comparison.

#### 2.2.1. Classical Genetic Algorithm

The Classical Genetic Algorithm (GA) serves as a baseline method using a single population. Each chromosome is evaluated using the unified objective function in Equation (3), which combines both hard and soft constraint penalties. Standard genetic operators, including tournament selection, one-point crossover, and mutation, are applied.

#### 2.2.2. Cooperative Nurse Scheduling (Proposed)

The proposed Cooperative Co-Evolutionary Nurse Scheduling Problem (Coop-NSP), which decomposes the optimization process into two interacting subpopulations corresponding to hard and soft constraints. The approach separates feasibility-driven and quality-driven objectives into two independently evolving subpopulations. The hard-constraint subpopulation focuses on minimizing feasibility violations, while the soft-constraint subpopulation focuses on improving schedule quality. Both subpopulations evolve independently using standard genetic operators. Each subpopulation evolves independently using standard genetic operators, including selection, crossover, and mutation, with no chromosome exchange between subpopulations. As shown in Pseudocode 1, cooperation is introduced solely during fitness evaluation through a contextual evaluation mechanism. Individuals from one subpopulation are evaluated in the context of the current best solution from the complementary subpopulation, ensuring global schedule consistency while maintaining independent optimization processes. Through this mechanism, improvements in one subpopulation influence the evolutionary pressure in the other only when compatible solutions emerge, promoting gradual alignment between feasibility-oriented and quality-oriented objectives over successive generations.

**Pseudocode 1** Cooperative Co-Evolutionary NSP

**Initialize** hard population $x$
**Initialize** soft population $y$
**Evaluate** $HC(H_i)$ for all $H_i \in X$
**Evaluate** $SC(S_j)$ for all $S_j \in Y$
**For** *generation* = 1 **to** *MAX_GEN* **do**
  **Evolve** hard subpopulation using hard penalties
  **Evolve** soft subpopulation using soft penalties
  **Identify** best hard chromosome $H_{best}$
  **Identify** best soft chromosome $S_{best}$
  **Evaluate** cooperative fitness:
    **For** each hard chromosome, $H_i$:
      $F(H_i) = HC(H_i) + SC(S_{best})$
    **For** each soft chromosome, $S_j$:
      $F(S_j) = HC(H_{best}) + SC(S_j)$
  **Evaluate** final combined schedule:
      $F(x) \ = HC(x) + SC(x)$
  Record best and average cooperative penalty
**end for**

In this cooperative framework, collaborator selection is performed using a deterministic strategy in which the current best-performing chromosome from the complementary subpopulation is selected as

the representative collaborator during evaluation. This ensures a stable and consistent reference for fitness assessment across generations. The cooperation mechanism is intentionally maintained in a structured and non-adaptive manner, where interaction occurs only through contextual fitness evaluation rather than dynamic parameter adjustment or chromosome exchange. This design avoids additional computational overhead while preserving convergence stability and interpretability of the optimization process. The cooperative optimization strategy is motivated by prior work on cooperative and decomposition-based evolutionary algorithms applied to complex scheduling and optimization problems. Cooperative coevolution has demonstrated improved scalability and stability in environments with interacting decision components, including large-scale optimization, workflow scheduling and distributed systems [37], [38].

Extensions incorporating diversity preservation, cluster-based decomposition, and hybrid evolutionary mechanisms further support the effectiveness of cooperative algorithms under uncertainty [39], [40]. In healthcare scheduling, integrated and cooperative approaches have been increasingly explored to coordinate multiple operational layers, highlighting the importance of decomposed optimization structures that preserve feasibility while enabling controlled interaction between subproblems [41].

### 2.3. Experimental Setup

Experiments were conducted based on the NSP formulation described in Section 2.1. The experimental configuration reflects realistic hospital rostering conditions and is designed to enable a fair and consistent comparison between optimization strategies. The problem instance consists of 500 nurses assigned over a 7-day planning horizon, resulting in a total of 3,500 decision variables. Each chromosome therefore represents a complete weekly schedule with a size of 500 by 7. This configuration reflects a large-scale real-world scheduling problem where both feasibility and solution quality must be maintained under complex constraint interactions. Key dataset characteristics and algorithm parameters are summarized in Table 2. This corresponds to a high-dimensional combinatorial optimization problem, reinforcing the need for advanced evolutionary strategies such as cooperative co-evolution.

**Table 2.** Algorithm Parameters.

| Parameter | Population Size | Selection Method | Crossover Operator | Mutation Operator | Max Generations | Runs |
|---|---|---|---|---|---|---|
| Value | 100 (hard) 100 (soft) | Tournament (k = 2) | One-point (0.7) | Random (0.3) | 15,000 | 30 |

Both the Classical GA and the proposed Coop-NSP were evaluated under identical experimental settings using the same chromosome representation, genetic operators, penalty formulation, and termination criteria. The only difference lies in the optimization strategy: GA optimizes all constraints within a single population, whereas Coop-NSP decomposes optimization into two interacting subpopulations targeting hard and soft constraints. Cooperation is introduced through contextual fitness evaluation as defined in equations (1) and (2). Each algorithm was executed for 30 independent runs with different random seeds, and performance was evaluated using best-of-generation and average population penalties to assess solution quality, convergence behavior, and population stability.

## 3. Results and Discussion

This subsection presents the experimental results obtained from comparing the proposed Coop-NSP with a Classical GA. Both approaches were evaluated under identical problem formulations, penalty models, and algorithmic parameters.

### 3.1. Convergence Behaviour

As shown in Fig. 2, the Classical GA exhibits rapid convergence during the early stages of evolution. Both the best-of-generation and average penalty curves decrease sharply within the first few thousand generations, indicating strong selection pressure and efficient exploitation of the search space. After this initial phase, the curves stabilize and remain closely aligned, reflecting rapid population uniformity once

feasible solutions are identified. The narrow and persistent gap between the best and average penalties in later generations suggests that most individuals in the population converge toward similar solution structures. While this behavior is effective for quickly producing high-quality solutions in standalone NSP instances, it also indicates reduced population diversity, which limits further exploration of alternative feasible configurations as optimization progresses.

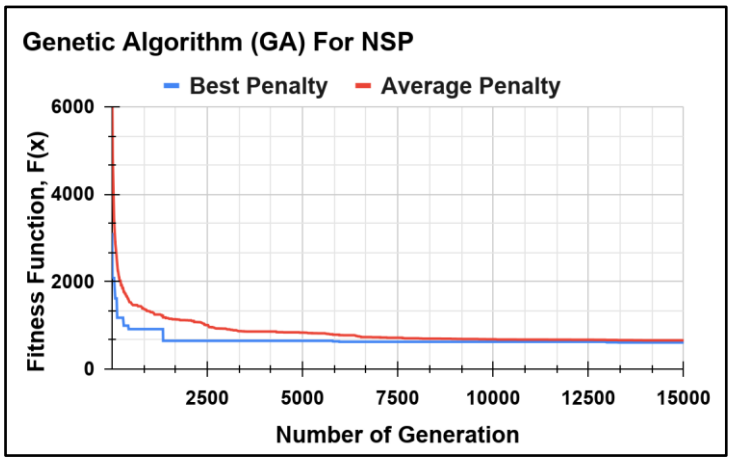



**Fig. 2.** Convergence curves of the Classical GA showing best-of-generation and average population penalty values across 15,000 generations for the NSP.

In contrast, the convergence behavior of the proposed Coop NSP, as shown in Fig. 3, follows a distinctly different pattern. Instead of smooth monotonic convergence, Coop NSP exhibits a stepwise convergence profile characterized by extended plateau phases followed by abrupt reductions in penalty values. These plateau phases indicate periods during which improvements achieved within one subpopulation, either the hard constraint or soft constraint group, do not immediately result in global performance gains. Overall improvement occurs only when compatible solutions emerge in the complementary subpopulation through contextual fitness evaluation.

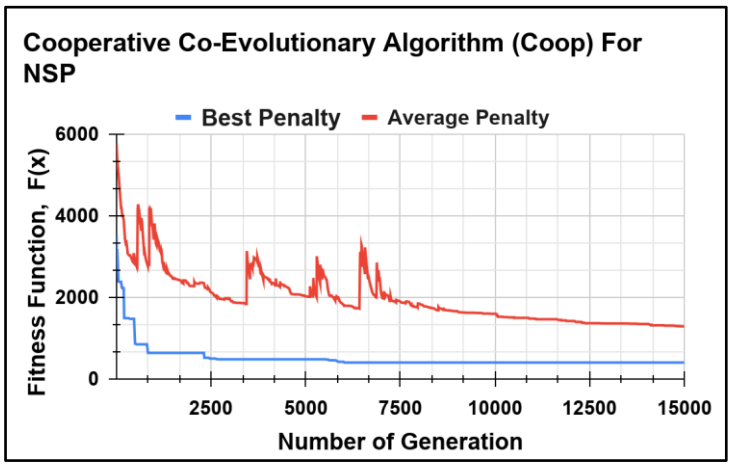



**Fig. 3.** Convergence curves of the proposed Cooperative Co-evolutionary (Coop) showing best-of-generation and average contextual penalty values across 15,000 generations under cooperative interaction between hard and soft constraint subpopulations for NSP.

Despite slower initial progress compared to the Classical GA, the average penalty curve in Coop-NSP remains consistently separated from the best-of-generation curve throughout the evolutionary process. This sustained separation indicates the preservation of structured population diversity arising from independent subpopulation evolution. Furthermore, the absence of oscillatory or unstable behavior indicates that cooperative interaction provides a controlled optimization process, enabling gradual refinement of schedule quality while maintaining feasibility. This behavior indicates that the proposed

approach maintains a balance between exploration and exploitation through structured population diversity and controlled cooperative interaction.

### 3.2. Statistical Performance Analysis

Table 3 summarizes the statistical performance of both optimization approaches over 30 independent runs, including mean, standard deviation, and execution time. To strengthen the robustness of the comparison, additional statistical measures are incorporated, including 95% confidence intervals, non-parametric significance testing, multiple comparison correction, and effect size evaluation.

**Table 3.** Statistical Significance, Confidence Interval, and Effect Size Analysis Over 30 Runs.

| Statistic | Best Penalty | | Average Penalty | | Execution Time | |
|---|---|---|---|---|---|---|
| *Algorithm* | *GA* | *Coop* | *GA* | *Coop* | *GA* | *Coop* |
| Mean | 651.42 | 768.72 | 880.08 | 2169.37 | 20.53s | 26.27s |
| Std. Dev. | 34.86 | 197.63 | 37.59 | 146.39 | 0.17s | 0.35s |
| 95% CI Lower | 638.40 | 694.92 | 866.04 | 2114.71 | 20.47s | 26.14s |
| 95% CI Upper | 664.44 | 842.51 | 894.12 | 2224.04 | 20.59s | 26.40s |
| Wilcoxon W | 93 | 93 | 0 | 0 | – | – |
| p-value | 0.0032 | 0.0032 | < 0.0001 | < 0.0001 | – | – |
| Adjusted p-value (Bonferroni) | 0.0064 | 0.0064 | < 0.0001 | < 0.0001 | – | – |
| Effect Size (Cohen's d) | 0.83 | 0.83 | 12.06 | 12.06 | – | – |

The statistical analysis confirms that the differences between the Classical GA and Coop-NSP are statistically significant. For best penalty, the Wilcoxon signed-rank test yields $p = 0.0032$ (adjusted $p = 0.0064$), indicating a significant difference with a large effect size ($d = 0.83$). For average penalty, the difference is highly significant ($p < 0.0001$) with a very large effect size ($d = 12.06$), reflecting substantial divergence in population behavior between the two optimization strategies. To further examine distributional characteristics, boxplot visualizations are presented in Fig. 4. The results show that the Classical GA produces a more concentrated distribution of penalty values, indicating consistent convergence across runs. In contrast, Coop-NSP exhibits a wider spread, reflecting increased variability and diversity in optimization outcomes. This behavior is attributed to the cooperative optimization mechanism, where the separation of hard and soft constraint subpopulations preserves diverse solution structures. While this enables Coop-NSP to occasionally achieve competitive best-case solutions, it also results in higher variability across runs compared to the more stable convergence behavior of the Classical GA.

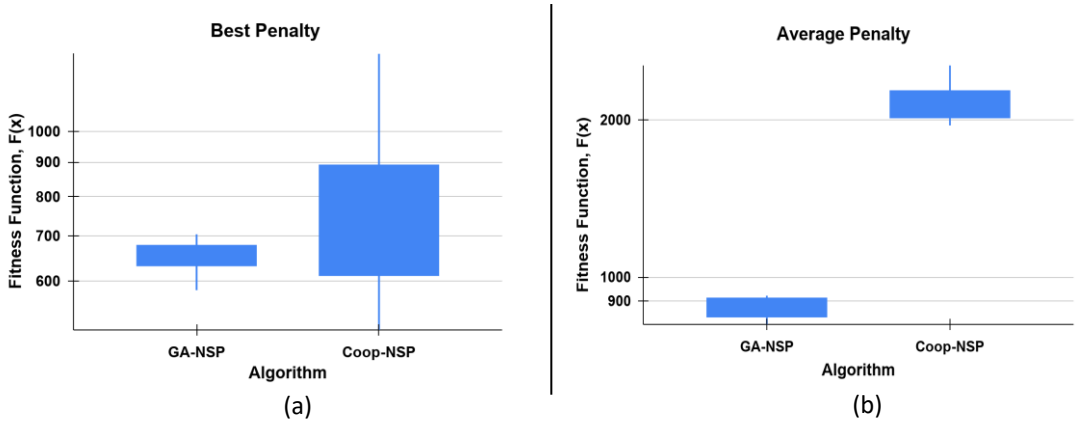



**Fig. 4.** Boxplot distributions of (a) best penalty and (b) average penalty obtained over 30 independent runs for Classical GA and Coop-NSP.

### 3.3. Discussion

This study presented a comparative evaluation of a Classical Genetic Algorithm (GA) and a Cooperative Co-Evolutionary approach for the Nurse Scheduling Problem under identical experimental

conditions. The Classical GA demonstrated rapid convergence, lower final penalty values, and reduced computational cost for standalone NSP instances, confirming the effectiveness of monolithic evolutionary optimization when feasibility and quality constraints are tightly coupled and evaluated jointly. In contrast, the proposed Coop-NSP exhibited a distinct optimization behavior characterized by stepwise convergence and sustained population diversity, resulting from contextual cooperative interaction between hard constraint and soft constraint subpopulations. Although this cooperative approach converges more slowly when applied to the NSP in isolation, the results highlight important structural advantages of cooperative co-evolution, particularly in maintaining feasibility and supporting stable optimization under specialized evolutionary pressures.

Rather than aiming to replace classical evolutionary strategies for isolated nurse scheduling tasks, the proposed cooperative algorithm establishes a structured and extensible optimization foundation that is well suited for more complex healthcare scheduling environments involving multiple interacting decision components. This structured formulation is particularly relevant for real-world healthcare operations, where scheduling decisions are interdependent across multiple domains. The cooperative framework provides a foundation for coordinating nurse rostering with related processes such as operating theatre scheduling and patient admission planning, enabling more holistic and operationally consistent decision-making. The observed optimization behavior of the proposed Coop-NSP aligns with findings reported in integrated and hybrid scheduling studies, where structured or decomposed optimization algorithms prioritize feasibility preservation and coordination stability over aggressive convergence speed [42], [43]. Similar trade-offs between convergence efficiency and solution robustness have been reported in integrated patient admission and resource scheduling problems, particularly when uncertainty and downstream constraints are considered [44]. In nurse rostering, hybrid evolutionary algorithms combining global search with problem-specific strategies have also demonstrated improved feasibility handling under complex constraint structures, supporting the suitability of modular evolutionary designs for realistic scheduling environments [45].

From a computational perspective, the proposed cooperative framework introduces additional overhead due to the use of two interacting subpopulations and contextual fitness evaluation. This increases computational cost compared to a classical GA but improves search diversity and reduces premature convergence, leading to more stable optimization behavior across runs. However, the study is conducted on a single controlled NSP configuration, which limits generalizability across different problem scales and constraint conditions. Although statistical validation has been performed, the findings should be interpreted within this scope. Further evaluation on diverse datasets is required to confirm robustness. Overall, the approach reflects a trade-off between efficiency and robustness, where improved stability and extensibility may come at the expense of real-time applicability. From a scalability perspective, the modular structure of the cooperative framework supports extension to larger nurse populations and more complex scheduling environments. By distributing optimization across interacting components rather than a single population, the model can better handle increasing problem size and complexity while maintaining global consistency. In practical deployment settings, factors such as parameter sensitivity, computational resources, and robustness under dynamic operational conditions must be considered. The diversity-preserving nature of the cooperative framework enhances adaptability to disruptions such as staffing changes or emergency demand, supporting its applicability in real-world healthcare scheduling systems.

## 4. Conclusion

This study examined the effectiveness of a cooperative co-evolutionary formulation for solving the Nurse Scheduling Problem (NSP) and compared its performance with a classical Genetic Algorithm (GA) under identical experimental conditions. Unlike conventional evolutionary approaches that optimize all constraints within a single population using a unified fitness function, the proposed approach separates feasibility-driven and quality-driven objectives into two interacting subpopulations that cooperate through contextual fitness evaluation. The results demonstrate that while the Classical GA achieves faster convergence and lower overall penalty values for standalone NSP instances, the cooperative

approach maintains higher population diversity and exhibits more stable optimization behavior under interacting constraints. Although the cooperative approach incurs higher computational cost, its structured optimization framework provides advantages in maintaining feasibility and managing constraint interactions, making it more suitable for complex scheduling environments involving multiple interdependent objectives. Future work will extend the proposed framework to larger-scale and integrated healthcare scheduling scenarios, including operating theatre and patient admission planning, as well as further validation across diverse problem instances to strengthen robustness and generalizability.

## Acknowledgment

The authors gratefully acknowledge the support of Universiti Teknikal Malaysia Melaka and the Ministry of Higher Education Malaysia under the Fundamental Research Grant Scheme [FRGS/1/2023/ICT02/UTEM/02/3]. This research was also supported by the Universiti Teknikal Malaysia Melaka Kesidang Scholarship, which funded the first author's postgraduate studies.

## Declarations

**Author contribution.** M.F.M.N. contributed to the conceptualization, methodology, algorithm development, implementation, analysis, and manuscript preparation. Z.I.M.Y. provided supervision, methodological guidance, validation, and manuscript review. H.B. contributed co-supervision, scheduling domain expertise, validation, and manuscript review. A.D. provided hospital scheduling domain expertise and validated practical feasibility.

**Funding statement.** The Universiti Teknikal Malaysia Melaka (UTeM) Kesidang Scholarship supported the first author's postgraduate studies.

**Conflict of interest.** The authors declare no conflict of interest.

**Declaration on the use of AI-assisting tools.** Generative AI tools were used solely for language refinement and editorial assistance. All scientific content was developed and validated by the authors, who take full responsibility for the work.

**Additional information.** No additional information is available for this paper.

## References

[1] A. Jan, M. Yamamoto, and A. Ohuchi, "Search Algorithms for Nurse Scheduling with Genetic Algorithms," Springer, Boston, MA, 2002, pp. 149–161. doi: 10.1007/978-1-4615-0819-9_9.

[2] A. C. Adamuthe and R. Bichkar, "Hybrid Genetic Algorithmic Approaches for Personnel Timetabling and Scheduling Problems in Healthcare," in *International Journal of Computer Applications®*, IJCA, 2011, p. 8. [Online] Available at: https://www.ijcaonline.org/proceedings/ictsm/number2/2785-108/.

[3] C. M. Ngoo, S. L. Goh, S. N. Sze, N. R. Sabar, S. Abdullah, and G. Kendall, "A Survey of the Nurse Rostering Solution Methodologies: The State-of-the-Art and Emerging Trends," *IEEE Access*, vol. 10, pp. 56504–56524, 2022, doi: 10.1109/ACCESS.2022.3177280.

[4] P.-S. Chen and Z.-Y. Zeng, "Developing two heuristic algorithms with metaheuristic algorithms to improve solutions of optimization problems with soft and hard constraints: An application to nurse rostering problems," *Appl. Soft Comput.*, vol. 93, no. August, p. 106336, Aug. 2020, doi: 10.1016/j.asoc.2020.106336.

[5] N. Nico, N. Charibaldi, and Y. Fauziah, "Comparison of Memetic Algorithm and Genetic Algorithm on Nurse Picket Scheduling at Public Health Center," *Int. J. Artif. Intell. Robot.*, vol. 4, no. 1, pp. 9–23, May 2022, doi: 10.25139/ijair.v4i1.4323.

[6] H. İ. Koruca, M. S. Emek, and E. Gulmez, "Development of a new personalized staff-scheduling method with a work-life balance perspective: case of a hospital," *Ann. Oper. Res.*, vol. 328, no. 1, pp. 793–820, Sep. 2023, doi: 10.1007/s10479-023-05244-2.

[7] M. R. Hassani and J. Behnamian, "A scenario-based robust optimization with a pessimistic approach for nurse rostering problem," *J. Comb. Optim.*, vol. 41, no. 1, pp. 143–169, Jan. 2021, doi: 10.1007/s10878-020-00667-0.

[8] P. Rerkjirattikal, V.-N. Huynh, S. Olapiriyakul, and T. Supnithi, "A Goal Programming Approach to Nurse Scheduling with Individual Preference Satisfaction," *Math. Probl. Eng.*, vol. 2020, no. 1, pp. 1–11, Nov. 2020, doi: 10.1155/2020/2379091.

[9] E. C. Yagmur and A. Sarucan, "Nurse scheduling with opposition-based parallel harmony search algorithm," *J. Intell. Syst.*, vol. 28, no. 4, pp. 633–647, Oct. 2021, doi: 10.1515/JISYS-2017-0150/XML.

[10] C.-C. Lin, J.-R. Kang, and T.-H. Hsu, "A Memetic Algorithm with Recovery Scheme for Nurse Preference Scheduling," *J. Ind. Prod. Eng.*, vol. 32, no. 2, pp. 83–95, Feb. 2015, doi: 10.1080/21681015.2014.997815.

[11] X. Ma *et al.*, "A Survey on Cooperative Co-Evolutionary Algorithms," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 421–441, Jun. 2019, doi: 10.1109/TEVC.2018.2868770.

[12] Q. Xiao, J. Zhong, L. Feng, L. Luo, and J. Lv, "A Cooperative Coevolution Hyper-Heuristic Framework for Workflow Scheduling Problem," *IEEE Trans. Serv. Comput.*, vol. 15, no. 1, pp. 150–163, Jan. 2022, doi: 10.1109/TSC.2019.2923912.

[13] S. Qin, D. Pi, Z. Shao, and Y. Xu, "A Cluster-Based Cooperative Co-Evolutionary Algorithm for Multiobjective Workflow Scheduling in a Cloud Environment," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 3, pp. 1648–1662, Jul. 2023, doi: 10.1109/TASE.2022.3183681.

[14] C. Wang *et al.*, "Cooperative co-evolutionary differential evolution algorithm applied for parameters identification of lithium-ion batteries," *Expert Syst. Appl.*, vol. 200, no. August, p. 117192, Aug. 2022, doi: 10.1016/j.eswa.2022.117192.

[15] M. Blocho, T. Jastrzab, and J. Nalepa, "Cooperative co-evolutionary memetic algorithm for pickup and delivery problem with time windows," in *GECCO 2022 Companion - Proceedings of the 2022 Genetic and Evolutionary Computation Conference*, Association for Computing Machinery, Inc, Jul. 2022, pp. 176–179. doi: 10.1145/3520304.3528782;TOPIC:TOPIC:CONFERENCE-COLLECTIONS.

[16] W.-J. Ou, X.-L. Shi, and W.-N. Chen, "A Distributed Cooperative Co-Evolutionary Algorithm Based on Ring Network for Distributed Large-Scale Optimization," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, Oct. 2022, pp. 3018–3025. doi: 10.1109/SMC53654.2022.9945430.

[17] L. Sun, L. Lin, H. Li, and M. Gen, "Cooperative Co-Evolution Algorithm with an MRF-Based Decomposition Strategy for Stochastic Flexible Job Shop Scheduling," *Mathematics*, vol. 7, no. 4, p. 318, Mar. 2019, doi: 10.3390/math7040318.

[18] A. Tayyab and U. Saif, "A Two-Stage Genetic Artificial Bee Colony Algorithm for Solving Integrated Operating Room Planning and Scheduling Problem With Capacity Constraints of Downstream Wards," *IEEE Access*, vol. 10, pp. 131109–131127, 2022, doi: 10.1109/ACCESS.2022.3229709.

[19] J.-J. Wang, Z. Dai, A.-C. Chang, and J. J. Shi, "Surgical scheduling by Fuzzy model considering inpatient beds shortage under uncertain surgery durations," *Ann. Oper. Res.*, vol. 315, no. 1, pp. 463–505, Aug. 2022, doi: 10.1007/s10479-022-04645-z.

[20] M. Arab Momeni, A. Mostofi, V. Jain, and G. Soni, "COVID19 epidemic outbreak: operating rooms scheduling, specialty teams timetabling and emergency patients' assignment using the robust optimization approach," *Ann. Oper. Res.*, vol. 359, no. 3, pp. 1–31, May 2022, doi: 10.1007/s10479-022-04667-7.

[21] H. Xu, Y. Fang, C.-A. Chou, N. Fard, and L. Luo, "A reinforcement learning-based optimal control approach for managing an elective surgery backlog after pandemic disruption," *Health Care Manag. Sci.*, vol. 26, no. 3, pp. 430–446, Sep. 2023, doi: 10.1007/s10729-023-09636-5.

[22] A. L. Bolaji *et al.*, "A room-oriented artificial bee colony algorithm for optimizing the patient admission scheduling problem," *Comput. Biol. Med.*, vol. 148, no. September, p. 105850, Sep. 2022, doi: 10.1016/j.compbiomed.2022.105850.

[23] R. Hosseini Rad, S. Baniasadi, P. Yousefi, H. Morabbi Heravi, M. Shaban Al-Ani, and M. Asghari Ilani, "Presented a Framework of Computational Modeling to Identify the Patient Admission Scheduling Problem in the Healthcare System," *J. Healthc. Eng.*, vol. 2022, no. 1, pp. 1–15, Nov. 2022, doi: 10.1155/2022/1938719.

[24] S. Ashraf, N. Rehman, S. Abdullah, B. Batool, M. Lin, and M. Aslam, "Decision support model for the patient admission scheduling problem based on picture fuzzy aggregation information and TOPSIS methodology," *Math. Biosci. Eng.*, vol. 19, no. 3, pp. 3147–3176, 2022, doi: 10.3934/mbe.2022146.

[25] L. Zhou, N. Geng, Z. Jiang, and S. Jiang, "Integrated Multiresource Capacity Planning and Multitype Patient Scheduling," *INFORMS J. Comput.*, vol. 34, no. 1, pp. 129–149, Jan. 2022, doi: 10.1287/ijoc.2020.1048.

[26] B. Spratt and E. Kozan, "An integrated rolling horizon approach to increase operating theatre efficiency," *J. Sched.*, vol. 24, no. 1, pp. 3–25, Feb. 2021, doi: 10.1007/s10951-020-00655-6.

[27] M. Squires, X. Tao, S. Elangovan, R. Gururajan, X. Zhou, and U. R. Acharya, "A novel genetic algorithm based system for the scheduling of medical treatments," *Expert Syst. Appl.*, vol. 195, no. June, p. 116464, Jun. 2022, doi: 10.1016/j.eswa.2021.116464.

[28] J. Zhao and H. Wen, "Dynamic planning with reusable healthcare resources: application to appointment scheduling," *Flex. Serv. Manuf. J.*, vol. 34, no. 4, pp. 859–878, Dec. 2022, doi: 10.1007/s10696-021-09411-0.

[29] M. I. Restrepo, L.-M. Rousseau, and J. Vallée, "Home healthcare integrated staffing and scheduling," *Omega*, vol. 95, no. September, p. 102057, Sep. 2020, doi: 10.1016/j.omega.2019.03.015.

[30] X. Zhan, Z. Xie, and D. Yao, "An Integrated Scheduling Algorithm Based on a Process End Time-Driven and Long-Time Scheduling Strategy," *Symmetry (Basel).*, vol. 14, no. 10, p. 2106, Oct. 2022, doi: 10.3390/sym14102106.

[31] A. Sadeghi and M. J. Sadeghi, "Manpower Planning Scheduling in hospital operating rooms," *Int. J. Integr. Care*, vol. 22, no. S3, p. 7, Nov. 2022, doi: 10.5334/ijic.ICIC22264.

[32] A. Salami, B. Afshar-Nadjafi, and M. Amiri, "A Two-Stage Optimization Approach for Healthcare Facility Location- Allocation Problems With Service Delivering Based on Genetic Algorithm," *Int. J. Public Health*, vol. 68, p. 1605015, Feb. 2023, doi: 10.3389/ijph.2023.1605015.

[33] N. Larrain, S. Wang, T. Stargardt, and O. Groene, "Cooperation Improvement in an Integrated Healthcare Network: A Social Network Analysis," *Int. J. Integr. Care*, vol. 23, no. 2, p. 32, Jun. 2023, doi: 10.5334/ijic.6519.

[34] S. M. Nagarajan, G. G. Devarajan, A. S. Mohammed, T. V. Ramana, and U. Ghosh, "Intelligent Task Scheduling Approach for IoT Integrated Healthcare Cyber Physical Systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2429–2438, Sep. 2023, doi: 10.1109/TNSE.2022.3223844.

[35] A. Ala, F. E. Alsaadi, M. Ahmadi, and S. Mirjalili, "Optimization of an appointment scheduling problem for healthcare systems based on the quality of fairness service using whale optimization algorithm and NSGA-II," *Sci. Rep.*, vol. 11, no. 1, p. 19816, Oct. 2021, doi: 10.1038/s41598-021-98851-7.

[36] J. Wang, Z. Dai, J. Chang, J. (Junmin) Shi, and H. Liu, "Robust surgical scheduling for nonoperating room anesthesia (NORA) under surgical duration uncertainty," *Decis. Sci.*, vol. 55, no. 3, pp. 262–280, Jun. 2024, doi: 10.1111/deci.12584.

[37] A. Nikfarjam, A. Moosavi, A. Neumann, and F. Neumann, "Computing High-Quality Solutions for the Patient Admission Scheduling Problem Using Evolutionary Diversity Optimisation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13398 LNCS, no. 1, Springer Science and Business Media Deutschland GmbH, 2022, pp. 250–264. doi: 10.1007/978-3-031-14714-2_18.

[38] V. Junet *et al.*, "A decision support system based on artificial intelligence and systems biology for the simulation of pancreatic cancer patient status," *CPT Pharmacometrics Syst. Pharmacol.*, vol. 12, no. 7, pp. 916–928, Jul. 2023, doi: 10.1002/psp4.12961.

[39] N. A. Ab Rashid and S. Ya'acob, "Predictive Analytics on Scheduled Surgery Cases in Forecasting the Operating Theatre Utilisation," *Open Int. J. Informatics*, vol. 9, no. 1, pp. 45–52, Jun. 2021, doi: 10.11113/oiji2021.9n1.20.

[40] P. Zhang, J. F. Bard, D. J. Morrice, and K. M. Koenig, "Extended open shop scheduling with resource constraints: Appointment scheduling for integrated practice units," *IISE Trans.*, vol. 51, no. 10, pp. 1037–1060, Oct. 2019, doi: 10.1080/24725854.2018.1542544.

[41] O. Stan, C. Avram, I. Stefan, and A. Astilean, "Integrated innovative solutions to improve healthcare scheduling," in *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, IEEE, May 2016, pp. 1–6. doi: 10.1109/AQTR.2016.7501301.

[42] Y. Fu, H. Wang, and M. Huang, "Integrated scheduling for a distributed manufacturing system: a stochastic multi-objective model," *Enterp. Inf. Syst.*, vol. 13, no. 4, pp. 557–573, Apr. 2019, doi: 10.1080/17517575.2018.1545160.

[43] A. M. Turhan and B. Bilgen, "Mixed integer programming based heuristics for the Patient Admission Scheduling problem," *Comput. Oper. Res.*, vol. 80, no. April, pp. 38–49, Apr. 2017, doi: 10.1016/j.cor.2016.11.016.

[44] S. Ceschia and A. Schaerf, "Dynamic patient admission scheduling with operating room constraints, flexible horizons, and patient delays," *J. Sched.*, vol. 19, no. 4, pp. 377–389, Aug. 2016, doi: 10.1007/s10951-014-0407-8.

[45] X. Zhuo, H. Huang, Z. Cai, and H. Hu, "An hybrid evolutionary algorithm with scout bee global search strategy for Chinese nurse rostering problems," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, May 2015, pp. 769–775. doi: 10.1109/CEC.2015.7256969.