

Leave-Aware Multi-Skilled Crew Scheduling: A Bi-Objective Model and an Evaluation of NSGA-II and the Non-Dominated Sorting Firefly Algorithm

Pegah Tahmasebi¹ , Javad Behnamian² 

1. MSc, Industrial Engineering Department, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran, E-mail: pgh.thm@gmail.com
2. Corresponding Author, Professor, Industrial Engineering Department, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran. E-mail: Behnamian@basu.ac.ir

Article Info	ABSTRACT
Article type: Research Article Article history: Received 15 June 2026 Received in revised form 2 September 2026 Accepted 18 September 2026 Published online 1 October 2026 Keywords: airline crew scheduling, multi-skilled crew, leave-request fulfillment, bi-objective modeling, NSGA-II, NSFA, crew rostering	Airline crew scheduling is a challenging optimization problem that affects operational cost, crew utilization, and service reliability. This study addresses multi-skilled cabin crew scheduling through a bi-objective integer programming formulation that considers leave-request fulfillment and crew employment cost. The first objective minimizes denied leave requests, while the second minimizes assignment costs together with penalties for exceeding the allowable leave level and cost savings associated with unused allowance. The model considers three cabin positions, purser, flight attendant, and substitute, and enforces position-specific daily staffing requirements, mutually exclusive daily assignments, and upper bounds on granted leave. Two multi-objective metaheuristics, the Non-dominated Sorting Genetic Algorithm II (NSGA-II), and the Non-dominated Sorting Firefly Algorithm (NSFA) are applied. A instance with 40 multi-skilled crew members and a 28-day planning horizon is evaluated using runtime, number of Pareto solutions, diversity metric (DM), and mean ideal distance (MID). Across five runs, NSGA-II required substantially less computational time, with an average runtime of approximately 12 s, compared with 116 s for NSFA. In contrast, NSFA achieves a higher average DM (0.294 versus 0.037) and a lower average MID (0.0012 versus 0.0045). The results demonstrate an instance-specific trade-off between computational efficiency and the measured characteristics of the obtained Pareto solutions.
Cite this article: Tahmasebi, Pegah. and Behnamian, Javad, (2026)., Leave-Aware Multi-Skilled Crew Scheduling: A Bi-Objective Model and Evaluation of NSGA-II and Non-dominated Sorting Firefly Algorithm. <i>Journal of Engineering Management and Soft Computing</i>, 12 (4). 227-251. DOI: https://doi.org/10.22091/jemsc.2026.17220.1422	
	© The Author(s) retains the copyright. DOI: https://doi.org/10.22091/jemsc.2026.17220.1422
Publisher: University of Qom	

1) Introduction

Air transportation is a highly interconnected service system in which operational decisions directly influence cost, resource utilization, and service reliability. Among the major planning problems faced by airlines, crew scheduling is particularly challenging because crew assignments must simultaneously satisfy operational requirements, position-specific staffing needs, and employment-related policies. The Airline Crew Scheduling Problem (ACSP) concerns the allocation of crew members to scheduled duties over a planning horizon while satisfying a set of operational and contractual requirements (Deveci & Demirel, 2018; Kasirzadeh et al., 2017). In practice, crew scheduling can involve duty-time limitations, rest requirements, qualification restrictions, staffing requirements, and crew preferences. The problem becomes more flexible but also more complex when cabin crew members are multi-skilled and can perform different positions, such as purser, flight attendant, and substitute. In such settings, the assignment of a crew member to one position on a given day affects the availability of that crew member for other positions and, therefore, the overall feasibility and cost of the schedule. Crew-related expenditures also constitute a major component of airline operating costs, which further emphasizes the importance of effective crew assignment decisions (Deveci & Demirel, 2018). The combinatorial nature of crew scheduling makes computationally efficient solution approaches important, particularly as the number of crew members, planning days, positions, and operational requirements increases. Recent studies further indicate that contemporary crew scheduling research is increasingly concerned with practical implementation, crew individuality, operational disruptions, fatigue, and the integration of advanced optimization methods. A recent meta-review of aircrew scheduling identified emerging research challenges related to artificial intelligence, integration, resource sharing, crew individuality, robustness, efficiency, sustainability, and data availability (Sun et al., 2026). Recent airline crew scheduling research has also incorporated fatigue management into multi-objective scheduling and applied NSGA-II to balance operational and crew-related considerations (Ye et al., 2025). In related crew-planning settings, recent studies have emphasized operational implementation, disruption recovery, and multi-skilled optimization. For example, optimization-based crew rostering has been implemented for routine annual planning and daily crew assignment in urban rail transit (Liebchen et al., 2026), while efficient crew re-scheduling procedures have been developed for emergency disruption recovery (Zhao et al., 2026). In addition, recent multi-skilled scheduling research has combined multi-objective optimization, hybrid NSGA-II, and sensitivity analysis to address reliability and disruption effects (Ghasemi et al., 2026). Together, these studies indicate a broader movement toward multi-objective, multi-skilled, implementation-oriented, and disruption-aware scheduling frameworks. Exact optimization methods can provide optimal solutions for appropriately sized instances, but their computational requirements may become prohibitive as problem dimensions grow (Gopalakrishnan & Johnson, 2005; Ozdemir & Mohan, 2001). Consequently, heuristic and metaheuristic approaches have remained important for obtaining high-quality solutions within acceptable computational times (Ernst et al., 2004). In addition to operational feasibility and cost, crew preferences can be relevant to schedule quality. One such preference concerns requested leave days. A scheduling model that treats leave requests only as hard constraints

may restrict the feasible scheduling space, whereas treating them as preferences allows the optimization process to balance leave-request fulfillment against operational and cost considerations. This distinction is particularly relevant when staffing requirements make it impossible to grant all requested leave days.

Motivated by this issue, the present study develops a bi-objective integer programming formulation for multi-skilled cabin crew scheduling. The formulation considers three cabin crew positions—purser, flight attendant, and substitute—and jointly addresses two objectives: minimizing denied leave requests and minimizing total crew employment cost. The leave component represents the number of requested leave days that are not granted, while the cost objective accounts for position-dependent assignment costs and a parameterized leave policy involving an allowable leave level, penalties for exceeding the allowance, and cost savings associated with unused leave allowance. The formulation therefore provides a decision framework in which leave-request fulfillment and assignment cost can be considered simultaneously rather than treating leave requests exclusively as hard feasibility requirements. The main objectives of this study are threefold. First, the study aims to develop a bi-objective integer programming model for multi-skilled cabin crew scheduling that explicitly balances leave-request fulfillment against crew employment cost while accounting for position-specific staffing requirements and a parameterized leave policy. Second, it aims to implement and compare two multi-objective metaheuristic approaches, NSGA-II and NSFA, using a common solution representation and the same scheduling formulation in order to examine their computational efficiency and the characteristics of the non-dominated solutions they generate. Third, the study aims to characterize the practical trade-off between computational efficiency and the characteristics of the obtained non-dominated solution sets by evaluating runtime, number of non-dominated solutions, diversity, and distance measures for the considered scheduling instance. The study makes three main contributions. First, it develops a bi-objective integer programming formulation for multi-skilled cabin crew scheduling that jointly incorporates leave-request fulfillment, position-specific staffing requirements, and a parameterized leave policy within the assignment-cost objective. Second, two multi-objective metaheuristics, the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2000) and the Non-dominated Sorting Firefly Algorithm (NSFA) (Yang, 2009), are applied to the resulting formulation and adapted to its matrix-based crew scheduling representation. Third, a computational evaluation is conducted using runtime (RT), number of Pareto solutions (NPS), diversity metric (DM), and Mean Ideal Distance (MID) to characterize the observed differences between the two solution approaches for the considered scheduling instance. The contribution of the computational study is therefore not to establish the general superiority of one algorithm over the other, but to provide instance-specific evidence on the computational and Pareto-front characteristics obtained by the two approaches. Accordingly, the study addresses the following research questions:

(1) How can leave-request fulfillment and crew employment cost be jointly incorporated into a bi-objective formulation for multi-skilled cabin crew scheduling? (2) How do NSGA-II and NSFA compare in terms of computational time for the considered scheduling instance? (3) How do the two algorithms compare with respect to the number of obtained Pareto solutions,

Pareto-front diversity, and distance from the ideal point? (4) What trade-off between computational time and the measured characteristics of the obtained Pareto solutions can be observed from the computational results?

To investigate these questions, the proposed formulation is evaluated on an instance comprising 40 multi-skilled crew members over a 28-day planning horizon. The instance considers daily staffing requirements for pursers, flight attendants, and substitutes. NSGA-II and NSFA are implemented using the same problem formulation and evaluated under the same experimental setting. The resulting Pareto solutions are assessed using RT, NPS, DM, and MID. The analysis focuses on the observed computational behavior and Pareto-front characteristics of the two algorithms for the considered instance rather than making broader claims regarding their performance across airline operations in general.

The remainder of the paper is organized as follows. Section 2 reviews the relevant literature on airline crew scheduling, multi-skilled crew assignment, crew preferences, and multi-objective solution approaches, and identifies the research context addressed by this study. Section 3 presents the bi-objective integer programming formulation and defines the associated parameters, decision variables, objectives, and constraints. Section 4 describes the solution methodology, including the implementation of NSGA-II and NSFA and the problem-specific solution representation and operators. Section 5 presents the test instance, experimental setup, and computational results. Section 6 concludes the paper and discusses its limitations and directions for future research.

2) Literature Review

The ACSP has attracted sustained attention since airline deregulation in the late 1970s (Gershkoff, 1989). Early studies focused on exact optimization methods (integer programming, branch-and-bound). Gershkoff (1989) examined crew scheduling in US airlines, demonstrating substantial cost savings. Anbil et al. (1991) reported advances at American Airlines. By the 1990s, exact methods had proved inadequate for realistic instances, prompting the development of heuristics. Zeghal and Minoux (2006) proposed a tabu search algorithm for short- and medium-haul routes. Xu et al. (2006) extended this with a dynamic neighborhood tabu search for flight instructor scheduling. Gamache et al. (2007) introduced a graph coloring model for monthly crew scheduling with preferential bidding. Souai and Teghem (2009) integrated crew pairing and rostering using a genetic algorithm. In a similar vein, Hung-Tso et al. (2012) addressed the multi-objective nature of crew rostering by applying goal programming to balance cost and crew satisfaction, an early demonstration of treating crew preferences as a formal objective.

Recent research has shown a substantial diversification of ACSP solution methodologies, particularly through hybrid metaheuristics, machine learning integration, robust optimization, and advanced column-generation approaches. Prior to this recent surge, Gattermann-Itschert et al. (2023) demonstrated that machine learning, specifically random forests, can effectively capture planners' preferences in crew scheduling, paving the way for the learning-based methods that followed.

Recent studies are reviewed to highlight the main methodological developments and emerging research directions in crew scheduling. Wen et al. (2021) provided a comprehensive

review of airline crew scheduling, systematically examining the major problem classes, mathematical models, and solution algorithms developed in the literature. Their review covers key components of airline crew scheduling, including crew pairing, crew rostering, and related integrated and recovery problems, while highlighting the evolution from exact optimization methods to heuristic and metaheuristic approaches. The study provides a structured overview of the state of the art and helps clarify the methodological foundations and remaining challenges in airline crew scheduling research, thereby providing a useful basis for positioning new optimization approaches within the broader ACSP literature. Wen et al. (2024) presented a novel crew pairing approach with accurate characterization of cross-class substitution, distinguishing substitutions among different pairs of crew classes through precise characterization of work time and costs. They developed a branch-and-price solution approach with a novel column-fixing branching strategy, achieving a 0.01% optimality gap on real-world flight schedules. This work is directly relevant to multi-skilled crew scheduling as it formalizes how higher-class crew can substitute lower-class positions. Racette et al. (2025) proposed an accelerated windowing method for the crew rostering problem that combines machine learning (ML) and combinatorial optimization (CO). Their sequential assignment procedure (seqAsg) using a neural network trained by evolutionary algorithm generates initial solutions, which are then re-optimized using branch-and-price with overlapping windows. The method is over ten times faster than the state-of-the-art GENCOL solver while providing solutions that are, on average, less than 1% from optimality.

Khiabani et al. (2024) constructed a new Integrated Aircraft and Crew Recovery model with multi-objective and priority considerations (IACR_MP). They designed a time-variable competitive swarm optimizer with tri-competition mechanisms to balance exploitation and exploration, incorporating three repair schemes to overcome high-constraint features and a self-exploration strategy. Results show significant outperformance of competitors in solving integrated flight recovery problems. Nourmohammadzadeh and Voß (2024) addressed robust airline fleet and crew scheduling using a metaheuristic approach combining particle swarm optimization (PSO), simulated annealing (SA), and mathematical programming of sub-problems. This hybridization represents a promising avenue for handling robustness issues that became particularly important after the COVID-19 pandemic. Eriksson (2024) proposed the GRU-Enhanced Pairing Initializer (GEPI), integrating Gated Recurrent Units into column generation for the crew pairing problem. GEPI generates an initial partial solution using supervised learning before the traditional column generation process. The results show that GEPI-generated pairings reduced flight schedule cost in 34 test cases, demonstrating the potential of neural networks to improve algorithm starting points. Wen and He (2025) introduced the concept of pairing similarity (both flight-based and duty-based similarity) to mitigate drawbacks of the individual crew pairing approach. Their column generation-based algorithm aims to promote team-building opportunities while maintaining cost efficiency. Experiments show that duty-based similarity measures exhibit stronger ability to improve team spirit, with a range where team-building is improved with relatively small cost increases. van Rossum et al. (2024) addressed fairness in crew planning, proposing a rolling horizon approach with penalty-based feedback mechanism and column generation heuristic. Applied to

Netherlands Railways instances with up to 572 unique guards, the method satisfied individual rules for 95.2% of employees on average. While focused on railway crew, the fairness mechanisms are highly transferable to airline contexts. Abbas and Taha Yassen (2025) reviewed 142 papers since 2010, classifying solution methods from exact models to heuristics, hybrid metaheuristics, hyper-heuristics, and ML-driven heuristics. The review underscores the current state of machine learning-based hyper-heuristic optimization in solving crew scheduling issues and provides promising directions for future research. Herekoğlu and Kabak (2024) presented a column generation-based solution approach for the crew recovery problem utilizing a customized deep learning model. Applied to a major European airline, results indicate that the DL-supported model outperforms traditional methods in solution quality and efficiency, indicating the power of learning-based approaches for large-scale optimization problems. Baradaran and Hosseinian (2021) contributed a multi-objective integer formulation for the airline crew scheduling problem and compared NSGA-II with Multi-Objective Differential Evolution (MODE), providing a methodological basis for bi-objective crew scheduling models. Additionally, several other works have contributed to the advancement of crew scheduling. Deng et al. (2022) proposed a hybrid multi-objective metaheuristic combining genetic algorithm, variable neighborhood search, and Aquila optimizer for airline crew rostering with qualification and language constraints. Aggarwal et al. (2023) compared a customized genetic algorithm with column generation for the crew pairing optimization problem, handling over 400,000 pairings. Wen et al. (2022) developed an individual scheduling approach for multi-class cabin crew to address manpower requirement heterogeneity. Schrottenboer et al. (2023) ensured reserve-crew reliability through branch-and-price with robust optimization. Zhou et al. (2020) designed a multi-objective ant colony system (MOACS) that simultaneously optimizes fairness and crew satisfaction. These works further illustrate the methodological diversity that has shaped the field prior to the most recent wave of machine-learning-based methods. The classification of solution methods for the airline crew scheduling problem is presented in Table 1.

Table 1. Classification of Solution Methods

Category	Methodology/Approach	Representative Studies
Exact Methods	Integer programming; branch-and-price	Gershkoff (1989); Anbil et al. (1991); Wen et al. (2024); Schrottenboer et al. (2023)
Classical Metaheuristics	Tabu search; genetic algorithm	Zeghal & Minoux (2006); Xu et al. (2006); Souai & Teghem (2009)
Hybrid Metaheuristics	GA + VNS + Aquila; PSO + SA + mathematical programming; competitive swarm optimization	Khiabani et al. (2024); Nourmohammadzadeh & Voß (2024); Deng et al. (2022)
Machine-Learning-Driven Methods	Random forest; neural networks; GRU; deep learning	Racette et al. (2025); Eriksson (2024); Herekoğlu & Kabak (2024); Gattermann-Itschert et al. (2023)
Robust Optimization	Robust optimization; scenario/uncertainty-oriented approaches	Nourmohammadzadeh & Voß (2024); Schrottenboer et al. (2023)
Multi-Objective Evolutionary/Metaheuristic Methods	NSGA-II; improved NSGA; MOACS; MODE	Zhou et al. (2020); Baradaran & Hosseinian (2021)

The reviewed approaches are grouped into six methodological categories: exact methods, classical metaheuristics, hybrid metaheuristics, machine-learning-driven methods, robust

optimization, and multi-objective evolutionary methods. The classification highlights the methodological diversity of ACSP research, ranging from exact mathematical optimization and conventional metaheuristics to hybrid, learning-enhanced, robust, and multi-objective approaches.

Table 2 summarizes representative studies on airline crew scheduling and related crew-planning problems, covering the methodological developments represented in the literature review. Recent studies have increasingly incorporated machine learning, robust optimization, advanced column-generation procedures, and multi-objective evolutionary algorithms into crew scheduling and related crew-planning problems, reflecting a shift toward data-driven, resilient, and computationally adaptive solution methodologies (Abbas & Taha Yassen, 2025; Racette et al., 2025; Wen et al., 2021). Despite the breadth of existing research, several gaps remain. First, relatively limited attention has been given to formulations that explicitly represent leave-request fulfillment and crew assignment cost as separate, potentially conflicting objectives, particularly in multi-skilled cabin crew settings. Second, to the best of our knowledge, a direct computational comparison of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) and the Non-dominated Sorting Firefly Algorithm (NSFA) for this specific class of crew scheduling problem has not been reported. Third, less attention has been devoted to multi-skilled daily rostering with position-specific demand requirements across cabin roles; related studies on cross-class substitution have largely focused on crew pairing rather than daily rostering. Fourth, the integration of parameterized leave policies that account for excess and unused leave remains relatively limited in the reviewed literature. The present study addresses these gaps through a bi-objective integer programming formulation for multi-skilled cabin crew scheduling. The formulation jointly considers leave-request fulfillment and crew assignment cost while incorporating position-specific daily staffing requirements and a parameterized leave policy involving a leave allowance, penalties for exceeding the allowance, and rewards for unused allowance. Two established multi-objective metaheuristics, NSGA-II and NSFA, are then applied to the resulting formulation. Their performance is examined using runtime, number of Pareto solutions, DM, and MID on one test instance, comprising 40 crew members over a 28-day planning horizon. Accordingly, the computational results should be interpreted as instance-specific evidence rather than as general evidence of algorithmic superiority. The comparison illustrates the observed trade-off between computational time and the characteristics of the obtained Pareto solutions for the considered instance.

Table 2. Representative Studies on Airline Crew Scheduling

Reference	Methodology/Approach	Main Focus
Gershkoff (1989)	Integer programming; branch-and-bound	ACS and cost reduction
Anbil et al. (1991)	Mathematical programming	Large-scale airline crew scheduling
Zeghal & Minoux (2006)	Tabu search	Short- and medium-haul crew scheduling
Xu et al. (2006)	Dynamic Neighborhood Tabu Search	Flight instructor scheduling
Gamache et al. (2007)	Graph coloring	Monthly crew scheduling with preferential bidding
Souai & Teghem (2009)	Genetic algorithm	Integrated crew pairing and rostering
Wen et al. (2021)	Review	Airline crew scheduling models, problem classes, solution algorithms, and research challenges

Table 2. Representative Studies on Airline Crew Scheduling

Reference	Methodology/Approach	Main Focus
Wen et al. (2024)	Branch-and-Price with column-fixing branching	Crew pairing with cross-class substitution
Racette et al. (2025)	Machine learning + combinatorial optimization + branch-and-price	Accelerated crew rostering using sequential assignment and windowing
Khiabani et al. (2024)	Time-variable competitive swarm optimizer with repair schemes	Integrated aircraft and crew recovery
Nourmohammadzadeh and Voß (2024)	PSO + SA + mathematical programming	Robust airline fleet and crew scheduling
Eriksson (2024)	GRU + column generation	Crew pairing initialization
Wen & He (2025)	Column generation with pairing similarity	Individual crew pairing and team-building
van Rossum et al. (2024)	Rolling horizon + penalty-based feedback + column generation heuristic	Fairness in crew planning
Abbas & Taha Yassen (2025)	Systematic review of solution methodologies	Evolution of exact, heuristic, hybrid, hyper-heuristic, and ML-driven approaches
Herekoğlu & Kabak (2024)	Deep learning + column generation	Crew recovery optimization
Deng et al. (2022)	GA + VNS + Aquila optimizer	Multi-objective crew rostering with qualification and language constraints
Aggarwal et al. (2023)	Customized GA + column generation	Large-scale crew pairing optimization
Wen et al. (2022)	Individual scheduling approach	Multi-class cabin crew and heterogeneous manpower requirements
Schrotenboer et al. (2023)	Branch-and-Price + robust optimization	Reliable reserve-crew scheduling
Zhou et al. (2020)	MOACS	Fairness and crew satisfaction
Gattermann-Itschert et al. (2023)	Random forest + optimization	Learning planner preferences in crew scheduling
Hung-Tso et al. (2012)	Goal programming	Multi-goal crew rostering and crew satisfaction
Baradaran & Hosseini (2021)	NSGA-II + Multi-Objective Differential Evolution (MODE)	Multi-objective airline crew scheduling
Present study	Bi-objective integer programming + NSGA-II + NSFA	Multi-skilled cabin crew scheduling with leave-request fulfillment, position-specific demand, and parameterized leave policy

3) Mathematical Model

The problem considered in this study is a multi-skilled daily cabin crew scheduling problem over a finite planning horizon. A set of N crew members is available to cover the daily requirements of three cabin positions: purser, flight attendant, and substitute. Each crew member is assumed to be qualified for all three positions. For each crew member and planning day, the scheduling decision determines either one cabin position or an approved leave status. The input data consist of the position-specific staffing requirement for each day, the leave requests submitted by crew members, the assignment cost associated with each crew-position combination, and the parameters governing the allowable and maximum number of leave days. The model seeks to determine a feasible daily assignment of all crew members while jointly minimizing denied leave requests and total crew employment cost. The cost objective additionally incorporates penalties for leave exceeding the allowable level and cost savings associated with unused leave allowance. Thus, the problem explicitly represents the trade-off between satisfying crew leave preferences and maintaining the required daily staffing levels at minimum assignment-related cost.

The problem is formulated over a planning horizon of T days. The model assumes that each crew member is either assigned to one of the three cabin positions or granted approved leave on each planning day; ordinary off-duty days are not explicitly modeled. The schedule must satisfy the required number of crew members in each cabin position on each day while balancing leave-request fulfillment against crew assignment cost and the associated leave-policy adjustments.

Table 3. Parameters and Decision Variables of the Mathematical Model

Notation	Description	Typical Value
Index		
n	Index of crew members	$= 1, 2, \dots, N$
t	Index of planning days	$= 1, 2, \dots, T$
p	Index of positions (1: purser, 2: flight attendant, 3: substitute)	$= 1, 2, 3$
Parameter		
D_t^p	Required number of crew members in position p on day t	Presented in Table 4
c_{nt}^p	Cost of assigning crew member n to position p on day t	2.0 (purser), 1.5 (flight attendant), 1.0 (substitute)
λ_{nt}	1 if crew member n requests leave on day t , and 0 otherwise. It represents the crew member's leave request and serves as an input reflecting the crew member's leave preference.	Input from crew
AL	Allowed number of leave days without penalty	4 days/month
m	Penalty coefficient for exceeding the leave allowance	0.5 monetary units
l	Cost saving associated with each unused leave day	0.5 monetary units
L^{max}	Maximum allowable leave days for each crew member.	7 days
M	Big-M constant used in the linearization constraints	Big value
Variable		
x_{nt}^p	1 if crew member n is assigned to position p on day t , and 0 otherwise	Binary
y_{nt}	1 if the leave request of crew member n for day t is granted, and 0 otherwise	Binary
Derived Quantity		
L_n	Total number of leave days assigned to crew member n	Integer
Auxiliary Variables		
E_n	Excess leave days beyond the allowed leave level AL for crew member n	Continuous, nonnegative
U_n	Unused portion of the allowed leave level AL for crew member n	Continuous, nonnegative
z_n	1 if $L_n > AL$, and 0 otherwise	Binary

3-1) Objective Functions

The first objective minimizes the number of denied leave requests, i.e., cases in which a crew member requests leave but is assigned to work. Thus, Z_1 counts the total number of requested leave days that are not granted. The model does not require every requested leave to be granted; instead, leave requests are treated as preferences and optimized jointly with staffing requirements and employment cost.

$$\min Z_1 = \sum_{t=1}^T \sum_{n=1}^N \lambda_{nt}(1 - y_{nt}) \quad (1)$$

The second objective minimizes total assignment cost together with the penalty for excess leave and the cost saving associated with unused leave allowance. The first term represents crew assignment costs. E_n represents leave days exceeding the allowed level AL , while U_n represents the unused portion of AL. Thus, the second term penalizes excess leave and the third term provides a cost reduction for unused leave allowance

$$\min Z_2 = \sum_{n=1}^N \sum_{t=1}^T \sum_{p=1}^3 c_{nt}^p x_{nt}^p + m \sum_{n=1}^N E_n - l \sum_{n=1}^N U_n \quad (2)$$

3-2) Constraints

Constraint (3) ensures that each crew member is assigned to exactly one feasible state on each day of the planning horizon. This state can be either one of the three working positions (purser, flight attendant, or substitute) or a leave status. Therefore, the sum of all assignment decisions and the leave decision for each crew member on each day must equal one, guaranteeing a unique and mutually exclusive assignment structure.

$$\sum_{p=1}^3 x_{nt}^p + y_{nt} = 1, \quad \forall n, t \quad (3)$$

The parameter λ_{nt} represents the leave request submitted by crew member n for day t , whereas y_{nt} is the corresponding decision variable indicating whether leave is actually granted. Thus, a requested leave is satisfied when $\lambda_{nt}=1$ and $y_{nt}=1$, whereas it is denied when $\lambda_{nt}=1$ and $y_{nt}=0$. Since Constraint (3) requires each crew member to be either assigned to one of the available positions or placed on leave, $1-y_{nt}$ indicates that the crew member is working. Therefore, $\lambda_{nt}(1-y_{nt})$ equals one exactly when a requested leave is denied.

Constraint (4) ensures that the staffing requirement for each cabin position on every day is satisfied. It requires the number of crew members assigned to each position to be at least equal to the required demand level D_t^p , thereby preventing understaffing and ensuring sufficient crew availability for each position. Any additional assignment beyond the required demand incurs the corresponding assignment cost in Objective (2).

$$\sum_{n=1}^N x_{nt}^p \geq D_t^p, \quad \forall t, p \quad (4)$$

Constraint (5) defines the total number of leave days assigned to each crew member over the planning horizon. It links the binary leave decision variable y_{nt} to the aggregate leave count L_n , ensuring consistency between daily decisions and cumulative leave accounting.

$$L_n = \sum_{t=1}^T y_{nt} \quad \forall n \quad (5)$$

Constraint (6) restricts the total number of leave days for each crew member to not exceed a predefined upper bound L^{max} . This constraint limits the total number of leave days assigned to each crew member and prevents excessive absence over the planning horizon. Note that the

allowed leave level AL determines the threshold for leave-related cost adjustments, whereas L^{max} poses a hard upper bound on the total leave assigned to each crew member.

$$L_n \leq L^{max} \quad \forall n \quad (6)$$

Constraints (7) – (12) linearize the excess-leave and unused-leave terms in Objective (2). E_n represents the amount by which the assigned leave exceeds AL , while U_n represents the unused portion of AL . The binary variable z_n identifies whether the assigned leave is on the excess-leave side or the unused-leave side. In this study, M is set to 4. This value is sufficient because the maximum possible value of E_n is $L^{max} - AL = 7 - 4 = 3$, while the maximum possible value of U_n is $AL = 4$. Therefore, $M = 4$ provides a valid upper bound for both auxiliary variables.

$$E_n \geq L_n - AL \quad \forall n \quad (7)$$

$$E_n \leq Mz_n \quad \forall n \quad (8)$$

$$E_n \leq L_n - AL + M(1 - z_n) \quad \forall n \quad (9)$$

$$U_n \geq AL - L_n \quad \forall n \quad (10)$$

$$U_n \leq M(1 - z_n) \quad \forall n \quad (11)$$

$$U_n \leq AL - L_n + Mz_n \quad \forall n \quad (12)$$

Constraint (13) restricts granted leave to submitted leave requests. Since y_{nt} represents whether the leave request of crew member n on day t is granted, a leave assignment is permitted only when a corresponding request exists. Thus, y_{nt} can equal one only if $\lambda_{nt} = 1$. This constraint ensures consistency between leave requests and granted leave assignments.

$$y_{nt} \leq \lambda_{nt}, \quad \forall n, t \quad (13)$$

Constraint (14) defines the domains of the decision and auxiliary variables. The assignment variables x_{nt}^p , leave variables y_{nt} , and binary indicator z_n are binary, while E_n and U_n are non-negative continuous auxiliary variables used in the linearization of Objective (2)

$$x_{nt}^p \in \{0,1\}, y_{nt} \in \{0,1\}, z_n \in \{0,1\}, E_n \geq 0, U_n \geq 0, \quad \forall n, t, p \quad (14)$$

4) The Proposed Algorithms

The mathematical model presented in Section 3 is a bi-objective integer programming formulation that belongs to the class of NP-hard problems (Ozdemir & Mohan, 2001). For instances with dozens of crew members and planning horizons of several weeks, the search space grows rapidly, making exhaustive enumeration of feasible schedules impractical. Exact optimization methods, such as branch-and-bound or branch-and-cut, may become computationally prohibitive as the problem dimensions increase. Moreover, the presence of two conflicting objectives—leave-request fulfillment and airline cost—requires consideration of a set of Pareto-optimal trade-off solutions rather than a single solution. Therefore, multi-objective metaheuristic algorithms provide a practical approach for obtaining high-quality approximations of the Pareto front within reasonable computational time. Metaheuristic methods, including the Firefly Algorithm and NSGA-II, have been widely applied to complex optimization problems in recent studies published in this journal. The Firefly Algorithm has been successfully employed for index tracking (Sajadi et al., 2022), improvement of algorithmic intelligence and computational speed (Solaymani et al., 2022), and cluster-head selection in

wireless sensor networks in combination with genetic algorithms (Moshtaghi Yazdani & Olyaei Torqabeh, 2023). NSGA-II and related evolutionary optimization approaches have also been applied to multi-objective decision-making problems, including predictive maintenance in supply chains (Pouri, 2025), green supply chain network design under environmental and energy considerations (Fathi et al., 2026), and sustainable supply chain optimization in the cement industry (Fathi Hafshejani et al., 2023). These studies demonstrate the applicability of both algorithmic families to complex optimization problems involving multiple objectives, nonlinear or combinatorial search spaces, and competing decision criteria. In this study, two multi-objective metaheuristics are considered: the well-established Non-dominated Sorting Genetic Algorithm II (NSGA-II) and the Non-dominated Sorting Firefly Algorithm (NSFA). The first essential step in applying either metaheuristic is to define an appropriate solution representation that captures the scheduling decisions. The remainder of this section first describes the solution representation and then presents the computational framework and the specific implementation of NSGA-II and NSFA for the multi-skilled crew scheduling problem

4-1) Solution Representation

A solution to the crew scheduling problem is represented as an $N \times T$ matrix, where N is the number of crew members and T is the number of days in the planning horizon. Each entry in this matrix takes one of four integer values: 1 indicates that the crew member works as a purser on that day, 2 indicates a flight attendant, 3 indicates a substitute, and 0 indicates that the crew member is on leave. For example, a cell value of 3 at position (2,1) means that crew member 2 works as a substitute on day 1 of the planning horizon.

4-2) General Framework

Both NSGA-II and NSFA are population-based multi-objective metaheuristics. Although they employ different search mechanisms, both algorithms iteratively generate and evaluate candidate schedules and retain non-dominated solutions. The common computational framework used in this study is summarized in Algorithm 1. Algorithm-specific search and selection mechanisms are described separately in Sections 4.3 and 4.4.

Algorithm 1. General Computational Framework for the Proposed Multi-Objective Solution Procedure

Input: Population size P , number of generations G

Output: Approximate Pareto front

Step 1: Initialization

Generate an initial population of P candidate schedules.

Repair infeasible schedules to satisfy the model constraints.

Evaluate the two objective functions for each individual using Equations (1) and (2).

Step 2: Evolutionary search

For generation $g = 1, \dots, G$:

Step 2.1: Generate new candidate solutions

Apply the algorithm-specific search operators to the current population to generate new candidate schedules.

For NSGA-II, crossover and mutation operators are applied.

For NSFA, firefly attraction and movement operators are applied.

Repair any infeasible candidate solutions.

Evaluate the two objective functions for all newly generated solutions.

Algorithm 1. General Computational Framework for the Proposed Multi-Objective Solution Procedure**Step 2.2: Combine candidate solutions**

Combine the current population and the newly generated solutions.

Step 2.3: Non-dominated sorting

Classify the combined population into successive Pareto fronts:

$F_1, F_2, \dots, F_k$, where F_1 contains the non-dominated solutions.

Step 2.4: Algorithm-specific diversity and selection

Apply the corresponding environmental selection mechanism of the algorithm.

For NSGA-II, crowding distance is calculated within each Pareto front and used together with the front rank to select the next population.

For NSFA, the firefly-specific attraction, movement, and selection mechanism is applied to preserve promising non-dominated solutions and guide the search toward diverse Pareto-optimal regions.

Step 2.5: Form the next population

Select P solutions according to the corresponding algorithm-specific selection mechanism.

Set the selected solutions as the population for the next generation.

Step 3: Termination

After G generations, extract the non-dominated solutions from the final population.

Step 4: Output

Return the resulting non-dominated solutions as the approximate Pareto front

4-3) NSGA-II

NSGA-II, proposed by Deb et al. (2000), is one of the most widely used multi-objective evolutionary algorithms. It operates on a population of candidate solutions and iteratively improves them over a predefined number of generations. In this study, an initial population of $P = 50$ feasible crew schedules is generated, where each individual is represented by an $N \times T$ matrix, as described in Section 4.1. Each individual is evaluated according to the two objective functions defined in Equations (1) and (2). A core component of NSGA-II is fast non-dominated sorting, which partitions the population into successive Pareto fronts $F_1, F_2, \dots$. The first front F_1 contains the non-dominated solutions, while subsequent fronts contain solutions dominated by solutions in preceding fronts. This ranking mechanism provides the primary criterion for comparing candidate schedules. To maintain diversity along the Pareto front, NSGA-II employs the crowding-distance measure. For each solution within a Pareto front, the crowding distance estimates the density of neighboring solutions in the objective space. Boundary solutions are assigned an infinite crowding distance so that extreme solutions are preferentially preserved. During selection, solutions with lower front rank are preferred, while solutions with larger crowding distance are preferred when they belong to the same front. After the selection stage, offspring solutions are generated using crossover and mutation operators, followed by feasibility repair. The specific parameter settings used in the computational experiments are reported in Section 5.

NSGA-II uses an elitist replacement strategy. The current parent population and the newly generated offspring population are combined, and the resulting population is sorted into Pareto fronts. Complete fronts are transferred to the next generation until the population size would be

exceeded. From the final partially accepted front, solutions with the largest crowding distances are selected until the population size reaches P . This procedure preserves high-quality non-dominated solutions while maintaining diversity across the objective space. The algorithm is executed for a predefined number of generations. At termination, the non-dominated solutions of the final population are extracted as the approximate Pareto front. The specific NSGA-II operators used in this study are summarized in Algorithm 2.

Algorithm 2. Pseudo-Code of NSGA-II

Input: Population size $P = 50$, maximum number of generations $G = 30$

Output: Approximate Pareto front

Step 1: Initialization

Generate P candidate schedules using the $N \times T$ matrix representation.
Apply the feasibility-repair procedure to infeasible schedules.

Step 2: Parent selection and offspring generation

For each generation $g = 1, \dots, G$:
Apply binary tournament selection based on Pareto rank and crowding distance.
Apply two-point matrix crossover to the selected parent schedules.
Apply uniform random mutation to the selected mutant solutions.
Apply the feasibility-repair procedure to the generated offspring.

Step 3: Fitness evaluation

Evaluate Z_1 and Z_2 for all generated solutions using Equations (1) and (2).

Step 4: Environmental selection

Apply non-dominated sorting and crowding distance.
Retain solutions according to Pareto rank and crowding distance.

Step 5: Termination

After $G = 30$ generations, extract the non-dominated solutions as the approximate Pareto front

Crossover Operator – Two-Point Matrix Crossover: In the NSGA-II implementation, as shown in Figure 1, a specialized matrix crossover operator is used to generate offspring from two parent schedules.

$$\begin{array}{ccc}
 \begin{bmatrix} 1 & 0 & 2 & 3 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 3 & 0 & 2 & 1 \\ 2 & 3 & 0 & 1 & 0 & 3 \\ 1 & 3 & 3 & 2 & 0 & 2 \end{bmatrix}_{N \times T} & & \begin{bmatrix} 0 & 1 & 2 & 2 & 1 & 3 \\ 0 & 1 & 2 & 3 & 3 & 1 \\ 3 & 0 & 3 & 0 & 2 & 1 \\ 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 2 & 1 & 3 & 0 & 2 \end{bmatrix}_{N \times T} \\
 \text{Parent 1} & & \text{Parent 2} \\
 \begin{bmatrix} 1 & 0 & 2 & 2 & 1 & 3 \\ 3 & 2 & 2 & 3 & 3 & 1 \\ 1 & 0 & 3 & 0 & 2 & 1 \\ 2 & 3 & 0 & 1 & 2 & 3 \\ 1 & 3 & 1 & 3 & 0 & 2 \end{bmatrix}_{N \times T} & & \begin{bmatrix} 0 & 1 & 2 & 3 & 1 & 2 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 3 & 0 & 3 & 0 & 2 & 1 \\ 2 & 3 & 0 & 1 & 0 & 3 \\ 0 & 2 & 3 & 2 & 0 & 2 \end{bmatrix}_{N \times T} \\
 \text{Offspring 1} & & \text{Offspring 2}
 \end{array}$$

Figure 1. Two-Point Matrix Crossover Operator for Crew Scheduling

Each chromosome is represented as an $N \times T$ matrix, where rows correspond to crew members and columns correspond to planning days. Two cut points are selected, one along the

crew dimension and one along the day dimension. The resulting matrix is divided into four regions, and complementary regions are exchanged between the two parents to generate the offspring. This operator allows combinations of assignment and leave patterns from the two parent schedules to be transferred to the offspring. After crossover, the resulting schedules are checked for feasibility and repaired when necessary.

Mutation Operator – Uniform Random Mutation: Uniform random mutation is applied to the selected mutant solutions. The mutation fraction determines the proportion of the population selected to undergo mutation, whereas the mutation rate determines the probability of modifying each decision variable within a selected mutant solution. When mutation occurs, the current cell value is replaced by one of the three alternative values from $\{0,1,2,3\}$. For the computational experiments, these parameters are set to 0.4 and 0.2, respectively, as reported in Section 5. For example, if a cell has a value of 1, representing a purser assignment, mutation may change it to 0 (leave), 2 (flight attendant), or 3 (substitute). This operator introduces additional variation into the population and helps prevent premature convergence. After mutation, the resulting schedule is checked and repaired if necessary to restore feasibility.

$$\begin{bmatrix} 3 & \dots \\ & \dots \\ 1 & \dots \\ & \dots \end{bmatrix}_{N \times T} \rightarrow \begin{bmatrix} 1 & \dots \\ & \dots \\ 3 & \dots \\ & \dots \end{bmatrix}_{N \times T}$$

Figure 2. Uniform Random Mutation Operator

4-4) Non-Dominated Sorting Firefly Algorithm

The Firefly Algorithm, originally proposed by Yang (2009), is a nature-inspired swarm intelligence metaheuristic based on the attraction behavior of fireflies. In the optimization context, each firefly represents a candidate solution, and less attractive fireflies are moved toward more attractive ones. In this study, the firefly search mechanism is adapted to the discrete multi-objective crew scheduling problem. The population consists of 50 fireflies, and the algorithm is executed for 30 iterations. The base attraction coefficient is set to $\beta = 0.5$, while the randomization coefficient α is generated from a uniform distribution $U(-1,1)$. The decision variables are bounded within $[0,3]$, corresponding to leave, purser, flight attendant, and substitute assignments. For each iteration, firefly i is compared with the other fireflies. In the implemented NSFA, the attraction step uses the assignment-cost objective Z_2 as a local search guide. Specifically, firefly i is moved toward firefly j when $Z_2(j) \leq Z_2(i)$. This cost-based attraction directs the search toward lower-cost regions, while the multi-objective quality of the population is maintained through fast non-dominated sorting and crowding distance during environmental selection. For each such pair, a random number I of decision variables is selected from the $N \times T$ scheduling matrix, where $1 \leq I \leq NT$. The selected positions are randomly determined and modified using the discrete movement rule described below. After movement, each modified decision variable is bounded within the allowable range $[0,3]$. For a selected decision variable, the new value is calculated as

$$x_{i,k}^{new} = x_{j,k} + \text{round}(\beta x_{j,k}) + \text{round}(\alpha e) \quad (15)$$

where $x_{j,k}$ is the current value of decision variable k for firefly j , β represents the attraction coefficient, α is the randomization coefficient, and e denotes a randomly selected integer from $\{1,2,3\}$. The resulting value is then restricted to the interval $[0,3]$ to preserve the discrete scheduling representation.

After all firefly movements are completed, the current population and the newly generated solutions are merged. The combined population is classified using fast non-dominated sorting, and crowding distance is calculated to support diversity preservation (Yang, 2013). The population is then sorted according to Pareto rank and crowding distance, and only the best 50 fireflies are retained for the next iteration. The non-dominated solutions of the final population constitute the approximate Pareto front.

Algorithm 3. Pseudo-Code of NSFA

Input: Population size ($P = 50$), maximum number of iterations ($G = 30$)

Output: Approximate Pareto front

Step 1: Initialization

Generate P candidate fireflies using the $N \times T$ matrix representation.

Evaluate the two objective functions for each firefly using Equations (1) and (2).

Apply the feasibility-repair procedure to infeasible solutions.

Step 2: Firefly movement

For iteration $g = 1, \dots, G$:

For each firefly $i = 1, \dots, P$:

For each firefly $j \neq i$:

If

$$Z_2(j) \leq Z_2(i) \tag{16}$$

then:

1. Randomly select an integer $I \in \{1, \dots, NT\}$.
2. Randomly select I decision-variable positions.
3. For each selected position k , update its value according to

$$x_{i,k}^{new} = x_{j,k} + \text{round}(\beta x_{j,k}) + \text{round}(\alpha e)$$
4. Restrict the updated value to $[0,3]$.

After all movements are completed, the current population and the modified population are merged for subsequent non-dominated sorting and environmental selection.

Step 3: Environmental selection

Merge the current population and the newly generated solutions.

Perform fast non-dominated sorting.

Calculate crowding distance.

Sort the combined population according to Pareto rank and crowding distance.

Retain the best $P = 50$ solutions.

Step 4: Termination

After $G = 30$ iterations, extract the non-dominated solutions of the final population as the approximate Pareto front.

4-5) Feasibility Repair

After initialization, and after each crossover, mutation, or firefly movement, each candidate schedule is checked for feasibility. First, the total number of granted leave days for each crew member is checked against L^{max} ; any excess leave assignments are converted to working

assignments. Second, for each planning day and cabin position, staffing requirements are checked against D_t^p . When a position-specific requirement is violated, eligible leave assignments are converted to the deficient position until the required demand is satisfied. If insufficient leave assignments are available to satisfy the required demand, the candidate schedule is classified as infeasible and is not retained in the population. Finally, Constraints (3), (4), (6), and (13) are rechecked before the candidate solution is accepted.

5) Computational Results

To capture the inherent stochastic variability of the metaheuristic algorithms, each algorithm was executed in five independent runs on the same test instance. A population size of 50 solutions and a maximum of 30 iterations were used for both NSGA-II and NSFA. For NSGA-II, the crossover fraction was set to 0.7, the mutation fraction was set to 0.4, and the mutation rate within the mutation operator was set to 0.2. For NSFA, the base attraction coefficient β was set to 0.5, while the randomization coefficient α was randomly generated from a uniform distribution $U(-1,1)$. The decision variables were bounded within the interval $[0,3]$, corresponding to the four discrete scheduling states: leave, purser, flight attendant, and substitute. All experiments were conducted on a standard desktop computer equipped with an Intel Core i7 processor, 16 GB of RAM, and Windows 11. The algorithms were implemented in Python 3.9 using the NumPy library for numerical computations.

5-1) Test Problem

The computational experiment uses a test instance rather than data obtained from a specific airline company. The instance is constructed to represent plausible multi-skilled cabin crew scheduling conditions. It comprises 40 crew members, all of whom are fully multi-skilled and therefore capable of serving in any of the three cabin positions (purser, flight attendant, or substitute). The planning horizon is 28 days, corresponding to a four-week monthly scheduling period. For each crew member, the number of requested leave days is randomly selected from 1 to 7 days over the 28-day planning horizon. The requested days are then randomly selected without replacement from the 28 planning days. The resulting binary parameter λ_{nt} equals 1 for a requested leave day, and 0 otherwise. Position-specific daily staffing requirements vary across the planning horizon and are reported in Table 4. The leave policy is defined by the following parameters: the allowed number of leave days without penalty (AL) is 4 days per month; the penalty per extra leave day (m) is 0.5 monetary units; the cost saving associated with each unused leave day (l) is 0.5 monetary units; and the maximum allowable leave days per crew member L^{max} is 7 days over the 28-day horizon. Finally, assignment costs are position-dependent and uniform across crew members: 2.0 monetary units per day for a purser, 1.5 for a flight attendant, and 1.0 for a substitute. These values assign the highest cost to the purser position and the lowest cost to the substitute position. Position-specific daily staffing requirements vary across the planning horizon and are reported in Table 4.

Table 4. Daily Crew Demand (D_t^p) for 28 Days

Day	Purser	Flight Attendant	Substitute	Day	Purser	Flight Attendant	Substitute
1	12	8	6	15	12	8	6
2	12	8	6	16	12	8	6
3	13	9	6	17	13	9	6
4	13	8	7	18	13	8	7
5	14	9	6	19	14	9	6
6	12	8	5	20	12	8	5
7	10	7	5	21	10	7	5
8	12	8	6	22	12	8	6
9	12	9	6	23	12	9	6
10	13	9	7	24	13	9	7
11	14	9	6	25	14	9	6
12	13	8	6	26	13	8	6
13	12	8	5	27	12	8	5
14	11	7	5	28	11	7	5

5-2) Performance Metrics

Four complementary performance metrics were employed to evaluate and compare the two algorithms.

- The first metric is the total runtime, measured in seconds from the start of algorithm execution until termination after 30 generations (iterations); lower runtime is preferable as it indicates greater computational efficiency.
- The second metric is the number of non-dominated solutions generated by each algorithm, denoted by |PF|, where a higher number provides the decision maker with a richer set of trade-off alternatives.
- The third metric is the diversity metric, which measures the overall spread of the obtained Pareto front. In this study, DM is calculated as the Euclidean distance between the extreme values of the two objective functions:

$$DM = \sqrt{\left(\max_i Z_{1i} - \min_i Z_{1i}\right)^2 + \left(\max_i Z_{2i} - \min_i Z_{2i}\right)^2} \quad (17)$$

According to the definition adopted in Equation (17), larger DM values indicate a greater spread of the obtained solutions across the objective space.

- The fourth metric is the Mean Ideal Distance (MID), which measures the average normalized Euclidean distance of the obtained Pareto solutions from the ideal point. The MID is calculated as:

$$MID = \frac{1}{|PF|} \sum_{i=1}^{|PF|} \sqrt{\left(\frac{f_{1i} - f_1^{best}}{f_1^{max} - f_1^{min}}\right)^2 + \left(\frac{f_{2i} - f_2^{best}}{f_2^{max} - f_2^{min}}\right)^2} \quad (18)$$

where |PF| represents the number of non-dominated solutions, f_{ji} denotes the value of objective j for solution i , f_j^{best} signifies the ideal value of objective j , and f_j^{max} and f_j^{min} are the maximum and minimum values of objective j , respectively, used

for normalization. A smaller MID indicates that the obtained Pareto solutions are, on average, closer to the ideal point.

5-3) Experimental Results

To evaluate the performance of the two algorithms under identical conditions, five independent runs were conducted for each method on the test instance described before. The detailed numerical results for NSGA-II and NSFA are presented in Table 5.

Table 5. Comparison of NSGA-II and NSFA Results

Run	NSGA-II				NSFA			
	RT (s)	NPS	DM	MID	RT (s)	NPS	DM	MID
1	11.7746	4	0.04295943	0.001007188	113.7296	6	0.2052438	0.001267330
2	11.6303	3	0.02445711	0.001033037	105.0878	8	0.2909038	0.001397370
3	12.1557	7	0.04012107	0.009640362	124.8764	4	0.3014963	0.001151675
4	11.9008	8	0.03395942	0.001007818	110.4861	10	0.4415295	0.001099857
5	11.9295	5	0.04160589	0.009657895	126.1703	3	0.2301562	0.001167604

The table reports RT, NPS, DM, and MID obtained in each independent run. These metrics provide complementary information on computational efficiency, the number of obtained Pareto solutions, the spread of the obtained solutions, and their distance from the ideal point. NSGA-II shows substantially lower computational time across all five runs, with runtimes ranging from 11.63 to 12.16 s. In contrast, NSFA requires between 105.09 and 126.17 s, indicating a considerably higher computational cost for the implemented firefly-based search procedure. The number of non-dominated solutions varies across runs for both algorithms. NSGA-II produces between 3 and 8 solutions, whereas NSFA produces between 3 and 10 solutions. Therefore, based solely on NPS, neither algorithm consistently dominates the other. The DM values are consistently larger for NSFA, ranging from 0.2052 to 0.4415, compared with 0.0245 to 0.0430 for NSGA-II. According to the definition adopted in Equation (17), the larger DM values indicate a greater spread of the obtained solutions across the objective space. Thus, NSFA generates a more widely dispersed set of Pareto solutions in these experiments. Regarding convergence, NSFA obtains MID values between 0.00110 and 0.00140 across all five runs. NSGA-II obtains lower MID values in three of the five runs, whereas NSFA obtains lower MID values in the remaining two runs. Consequently, NSFA exhibits more consistent MID values across independent runs, whereas NSGA-II shows greater run-to-run variability in this metric. Overall, the results indicate a clear trade-off between computational efficiency and the characteristics of the obtained Pareto fronts. NSGA-II is substantially faster, whereas NSFA produces more widely dispersed Pareto solutions and shows more consistent MID values across the five runs. These differences are summarized and compared descriptively in terms of their mean performance across the five runs.

5-4) Comparative Analysis

The comparative analysis examines how the implemented solution mechanisms affect the computational behavior and decision-support characteristics of the scheduling model under controlled and identical experimental conditions. Since both algorithms are applied to the same mathematical formulation, the same test instance, and identical population and termination settings, the differences observed in their performance measures provide an implementation-level basis for examining the effect of the selected solution mechanism. Runtime provides an indication of computational efficiency, while NPS, DM, and MID provide complementary information about the number, spread, and distance characteristics of the obtained non-dominated solutions. Accordingly, the analysis considers not only the computational burden of each implementation but also how the selected search mechanism influences the range and characteristics of the scheduling alternatives available to the decision maker.

As shown in Table 6, the experimental results reveal clear trade-offs between the two algorithms. In terms of runtime, NSGA-II is approximately 9.8 times faster than NSFA, with average runtimes of 11.88 seconds and 116.07 seconds, respectively. This difference is consistent with the additional computational effort required by the firefly-based search, particularly the repeated pairwise comparisons and movement calculations. Therefore, NSGA-II is preferable when computational efficiency is the primary concern. Regarding the number of Pareto solutions, both algorithms produce comparable numbers of non-dominated solutions, with average NPS values of 5.4 for NSGA-II and 6.2 for NSFA. Thus, NPS alone does not provide a strong basis for distinguishing the two algorithms. A substantial difference is observed in diversity. The average DM of NSFA is 0.29387, compared with 0.03662 for NSGA-II, indicating that NSFA generates a substantially wider spread of solutions in the objective space. Specifically, the mean DM of NSFA is approximately 8.0 times that of NSGA-II. This result indicates a stronger diversity of the obtained Pareto solutions under the implemented NSFA procedure. Regarding convergence, NSFA achieves an average MID of 0.001217, compared with 0.004469 for NSGA-II. Thus, the average MID of NSFA is approximately 72.8% lower, indicating that its obtained solutions are, on average, closer to the ideal point. NSGA-II also shows greater variability in MID across the five runs, with particularly higher values in Runs 3 and 5.

Table 6. Mean Performance Comparison (NSGA-II vs. NSFA)

Metric	NSGA-II	NSFA	Relative Difference
RT (s)	11.88	116.07	877.17%
NPS	5.40	6.20	14.81%
DM	0.03662	0.29387	702.46%
MID	0.004469	0.001217	-72.77%

The results highlight a clear trade-off between computational efficiency and the characteristics of the obtained Pareto solutions. NSGA-II is more suitable when computational time is the primary operational constraint, whereas NSFA may be more appropriate when a broader exploration of the trade-off space is preferred and additional computational effort is acceptable. The two algorithms produce comparable numbers of non-dominated solutions, suggesting that their main difference lies in the computational effort and the distribution of the obtained solutions rather than in the number of alternatives alone. From a practical perspective, NSGA-II can therefore be considered a more suitable option for scheduling environments

requiring rapid solution generation or frequent schedule updates. NSFA, in contrast, may be useful in planning settings where the decision maker has more computational time available and wishes to examine a broader range of trade-off alternatives. A hybrid strategy combining the rapid search of NSGA-II with the broader exploration observed for NSFA could also be investigated in future research.

6) Conclusion and Future Research

This paper presented a bi-objective integer programming model for multi-skilled airline crew scheduling in which leave-request fulfillment and crew employment cost are considered simultaneously. The formulation represents three cabin crew positions—purser, flight attendant, and substitute—and incorporates position-specific daily staffing requirements, individual leave requests, an upper bound on granted leave, and a leave allowance with penalties for excess leave and cost savings associated with unused leave allowance. The resulting scheduling problem was solved using two multi-objective metaheuristics, the Non-dominated Sorting Genetic Algorithm II (NSGA-II), and the Non-dominated Sorting Firefly Algorithm (NSFA), using a matrix-based representation of crew assignments over the 28-day planning horizon. The computational results for the considered 40-crew instance show a clear difference in computational effort between the two approaches. NSGA-II required an average of approximately 11.88 seconds, whereas NSFA required approximately 116.07 seconds. Thus, NSGA-II generated the final population and corresponding non-dominated solutions with substantially lower computational time. In contrast, NSFA produced Pareto fronts with considerably larger DM values, with an average of 0.2939 compared with 0.0366 for NSGA-II, indicating a substantially wider range of obtained trade-off solutions under the adopted diversity measure. NSFA also obtained a lower average MID value than NSGA-II (0.001217 versus 0.004469). The number of non-dominated solutions was relatively similar, with mean NPS values of 6.2 for NSFA and 5.4 for NSGA-II. These results indicate that, for the tested instance, the main distinction between the two approaches lies not in the number of solutions obtained, but in the computational effort and the distribution characteristics of the resulting Pareto solutions

6-1) Practical Implications

The computational findings provide a practical basis for selecting between the two implemented solution approaches according to the scheduling context. For situations in which a planner needs to generate a feasible set of alternative schedules within a limited computational budget, NSGA-II is the more appropriate option among the two tested algorithms because its average runtime is substantially lower than that of NSFA (11.88 s versus 116.07 s). This characteristic is particularly relevant when schedules must be generated quickly or updated within a restricted planning time window. In contrast, when computational time is less restrictive and the planner places greater emphasis on exploring a wider range of trade-offs between leave-request fulfillment and crew employment cost, NSFA provides a useful alternative because its average DM is substantially higher than that of NSGA-II (0.2939 versus 0.0366). The similar average number of non-dominated solutions produced by the two algorithms (6.2 for NSFA and 5.4 for NSGA-II) indicates that the practical advantage of NSFA is mainly associated with the greater

spread of the obtained solutions rather than with generating a substantially larger number of alternatives.

Based on these observations, a practical decision rule can be considered for the scheduling process: NSGA-II should be preferred when computational speed is the primary requirement, whereas NSFA can be considered when the planner has a larger computational budget and requires broader exploration of the trade-off space. In an operational implementation, the planner can first specify the available computational time and the relative importance of rapid schedule generation versus the exploration of alternative trade-offs. The algorithm can then be selected according to these priorities. A further practical extension is to develop a sequential solution procedure in which NSGA-II is first used to rapidly generate an initial set of feasible trade-off solutions, followed by NSFA to explore selected regions of the objective space when additional computational time is available. Such a procedure would allow the faster algorithm to support rapid initial schedule generation while using the complementary search behavior of NSFA for further exploration. However, this hybrid procedure has not been implemented or validated in the present study and should therefore be regarded as a direction for future investigation rather than as a demonstrated operational strategy.

6-2) Limitations and Future Research

Several limitations of the present study should be acknowledged. First, the computational evaluation is based on a single synthetic instance with 40 crew members and a 28-day planning horizon. Although the instance represents plausible multi-skilled crew scheduling conditions, a single instance does not provide sufficient evidence to establish the scalability or general comparative superiority of either algorithm. Second, the formulation uses deterministic position-specific demand and focuses on two objectives, namely leave-request fulfillment and crew employment cost. Other operational and contractual features commonly encountered in airline crew scheduling, such as duty-time restrictions, rest requirements, qualification heterogeneity, and operational disruptions, are not explicitly represented in the current formulation. Third, the experimental comparison is based on heuristic results and does not include an exact benchmark or optimality-gap analysis for the main 40-crew instance. Consequently, the reported results characterize the relative computational and Pareto-front behavior of the two implemented algorithms rather than their distance from a known global optimum. Finally, the leave requests and staffing requirements are synthetically generated rather than obtained from an operational airline dataset, which limits the extent to which the numerical findings can be directly generalized to real airline planning environments.

Future research should address these limitations through a staged computational and empirical extension. First, the proposed model should be tested on multiple small-, medium-, and large-scale instances generated from real airline crew data, while preserving the same evaluation protocol for NSGA-II and NSFA. Second, small instances should be solved to optimality using an exact mixed-integer programming solver so that optimality gaps can be reported for the metaheuristic solutions. Third, sensitivity experiments should systematically vary the leave-policy parameters and staffing requirements to determine how changes in operational conditions affect leave-request fulfillment, assignment cost, and the relative

behavior of the two algorithms. Fourth, the model can be extended to incorporate uncertainty in demand and operational disruptions, as well as additional airline-specific constraints such as rest, qualification, and duty-time requirements. Finally, a hybrid strategy can be evaluated in which NSGA-II is used for rapid initial exploration and NSFA is subsequently applied to the selected regions of the objective space.

References

- Abbas, A. K., & Taha Yassen, E. (2025). Crew scheduling problems: A systematic literature review. *IEEE Access*, 13, 190196–190220. <https://doi.org/10.1109/ACCESS.2025.3628669>
- Aggarwal, D., Saxena, D. K., Bäck, T., & Emmerich, M. T. M. (2023). Real-world airline crew pairing optimization: Customized genetic algorithm versus column generation method. In *Evolutionary Multi-Criterion Optimization* (pp. 518–531). Springer. https://doi.org/10.1007/978-3-031-27250-9_37
- Anbil, R., Gelman, E., Patty, B., & Tanga, R. (1991). Recent advances in crew-pairing optimization at American Airlines. *Interfaces*, 21(1), 62–74. <https://doi.org/10.1287/inte.21.1.62>
- Baradaran, V., & Hosseini, A. H. (2021). A multi-objective mathematical formulation for the airline crew scheduling problem: MODE and NSGA-II solution approaches. *Journal of Industrial Management Perspective*, 11(1), 247–269. <https://doi.org/10.52547/jimp.11.1.247>
- Deb, K., Pratap, A., Agrawal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. <https://doi.org/10.1109/4235.996017>
- Deng, B., Ding, R., Li, J., Huang, J., Tang, K., & Li, W. (2023). Hybrid multi-objective metaheuristic algorithms for solving airline crew rostering problem with qualification and language. *Mathematical Biosciences and Engineering*, 20(1), 1460–1487. <https://doi.org/10.3934/mbe.2023066>
- Deveci, M., & Demirel, N. C. (2018). A survey of the literature on airline crew scheduling. *Engineering Applications of Artificial Intelligence*, 74, 54–69. <https://doi.org/10.1016/j.engappai.2018.05.008>
- Eriksson, W. (2024). *Machine learning-enhanced column generation for the crew pairing problem: Leveraging gated recurrent units to generate the initial restricted master problem* [Master's thesis, Chalmers University of Technology]. <https://hdl.handle.net/20.500.12380/309081>
- Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1), 3–27. [https://doi.org/10.1016/S0377-2217\(03\)00095-X](https://doi.org/10.1016/S0377-2217(03)00095-X)
- Fathi Hafshejani, K., Bagheri Sorkhi, M., & Modiri, M. (2023). Integrated hybrid model of sustainable supply chain in cement industry. *Engineering Management and Soft Computing*, 9(1), 1–18. <https://doi.org/10.22091/JEMSC.2021.6422.1144>
- Fathi, M. R., Saffarinia, A., & Pahlevanzadeh, M. J. (2026). Multi-objective green supply chain network design considering carbon emissions and electricity consumption. *Engineering Management and Soft Computing*, 12(1), 14–35. <https://doi.org/10.22091/jemsc.2026.14628.1326>
- Gamache, M., Hertz, A., & Ouellet, J. O. (2007). A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding. *Computers & Operations Research*, 34(8), 2384–2395. <https://doi.org/10.1016/j.cor.2005.09.010>
- Gattermann-Itschert, T., Poreschack, L. M., & Thonemann, U. W. (2023). Using machine learning to include planners' preferences in railway crew scheduling optimization. *Transportation Science*, 57(3), 796–812. <https://doi.org/10.1287/trsc.2022.1196>
- Gershkoff, I. (1989). Optimizing flight crew schedules. *Interfaces*, 19(4), 29–43. <https://doi.org/10.1287/inte.19.4.29>
- Ghasemi, M., Nazari, A., Thiruvady, D., Tavakkoli-Moghaddam, R., Shahabi-Shahmiri, R., & Mirnezami, S.-A. (2026). A multi-skilled resource-constrained project scheduling with reliability constraints. *Computers & Industrial Engineering*, 219, 112154. <https://doi.org/10.1016/j.cie.2026.112154>
- Gopalakrishnan, B., & Johnson, E. L. (2005). Airline crew scheduling: State-of-the-art. *Annals of Operations Research*, 140(1), 305–337. <https://doi.org/10.1007/s10479-005-3975-3>

- Herekoğlu, A., & Kabak, Ö. (2024). Crew recovery optimization with deep learning and column generation for sustainable airline operation management. *Annals of Operations Research*, 342(1), 399–427. <https://doi.org/10.1007/s10479-023-05738-z>
- Hung-Tso, L., Yen-Ting, C., Tsung-Yu, C., & Yi-Chun, L. (2012). Crew rostering with multiple goals: An empirical study. *Computers & Industrial Engineering*, 63(2), 483–493. <https://doi.org/10.1016/j.cie.2012.04.013>
- Kasirzadeh, A., Saddoune, M., & Soumis, F. (2017). Airline crew scheduling: Models, algorithms, and data sets. *EURO Journal on Transportation and Logistics*, 6(2), 111–137. <https://doi.org/10.1007/s13676-015-0080-x>
- Khiabani, A., et al. (2024). A time-varying competitive swarm optimizer for integrated flight recovery with multi-objective and priority considerations. *Computers & Industrial Engineering*, 190, 110019. <https://doi.org/10.1016/j.cie.2024.110019>
- Liebchen, C., Türk, C., & Vornhusen, B. (2026). Railway crew rostering: Tailored optimization solutions in daily use in Berlin. *Urban Rail Transit*, 12, 428–450. <https://doi.org/10.1007/s40864-026-00272-0>
- Moshtaghi Yazdani, N., & Olyaei Torqabeh, M. H. (2022). Optimizing cluster head selection in wireless sensor networks using firefly and genetics algorithm. *Engineering Management and Soft Computing*, 8(2), 193–211. <https://doi.org/10.22091/jemsc.2020.5695.1139>
- Nourmohammadzadeh, A., & Voß, S. (2025). Robust airline fleet and crew scheduling: A metaheuristic approach. In *Learning and Intelligent Optimization* (pp. 290–304). Springer. https://doi.org/10.1007/978-3-031-75623-8_23
- Ozdemir, H. T., & Mohan, C. (2001). Flight graph based genetic algorithm for crew scheduling in airlines. *Information Sciences*, 133(3–4), 165–173. [https://doi.org/10.1016/S0020-0255\(01\)00083-4](https://doi.org/10.1016/S0020-0255(01)00083-4)
- Pouri, K. (2025). Predictive maintenance in multi-objective supply chains by combining machine learning and evolutionary optimization. *Engineering Management and Soft Computing*, 11(2), 179–193. <https://doi.org/10.22091/jemsc.2025.12514.1264>
- Racette, P., Quesnel, F., Lodi, A., & Soumis, F. (2025). *Accelerated windowing for the crew rostering problem with machine learning*. arXiv. <https://doi.org/10.48550/arXiv.2503.00160>
- Sajadi, S. J., Sajadi, S. M., & Sabzevari, M. (2022). Index tracking problem solving using metaheuristic firefly algorithm. *Engineering Management and Soft Computing*, 7(1), 127–146. <https://doi.org/10.22091/jemsc.2016.585>
- Schrotenboer, A. H., Wenneker, R., Ursavas, E., & Zhu, S. X. (2023). Reliable reserve-crew scheduling for airlines. *Transportation Research Part E: Logistics and Transportation Review*, 178, 103283. <https://doi.org/10.1016/j.tre.2023.103283>
- Solaymani, F., Salehpour, N., & Nazari Farokhi, M. (2022). Present a new method for increasing the intelligence and speed of the Firefly algorithm. *Engineering Management and Soft Computing*, 8(1), 1–20. <https://doi.org/10.22091/jemsc.2018.753.1033>
- Souai, N., & Teghem, J. (2009). Genetic algorithm-based approach for the integrated airline crew-pairing and rostering problem. *European Journal of Operational Research*, 199(3), 674–683. <https://doi.org/10.1016/j.ejor.2007.10.065>
- Sun, X., Leng, Y., Wu, G., Yan, S., & Wandelt, S. (2026). AIRCREWSCHED: A meta review of aircrew scheduling and its research challenges. *Transportation Research Part E: Logistics and Transportation Review*, 205, 104525. <https://doi.org/10.1016/j.tre.2025.104525>
- Wen, X., Chung, S. H., Choi, T. M., & Fu, X. (2024). Airline cabin crew pairing with accurate characterization of cross-class substitution: A branch-and-price approach. *Transportation Research Part B: Methodological*, 190, 103084. <https://doi.org/10.1016/j.trb.2024.103084>
- Wen, X., Chung, S. H., Ji, P., & Sheu, J. B. (2022). Individual scheduling approach for multi-class airline cabin crew with manpower requirement heterogeneity. *Transportation Research Part E: Logistics and Transportation Review*, 163, 102763. <https://doi.org/10.1016/j.tre.2022.102763>
- Wen, X., & He, Y. (2025). Do costs and team-building conflict in airline crew scheduling? An individual crew pairing approach. *Transportation Research Part E: Logistics and Transportation Review*, 202, 104329. <https://doi.org/10.1016/j.tre.2025.104329>

- Wen, X., Sun, X., Sun, Y., & Yue, X. (2021). Airline crew scheduling: Models and algorithms. *Transportation Research Part E: Logistics and Transportation Review*, 149, 102304. <https://doi.org/10.1016/j.tre.2021.102304>
- Xu, J., Sohoni, M., McCleery, M., & Bailey, T. G. (2006). A dynamic neighborhood based tabu search algorithm for real-world flight instructor scheduling problems. *European Journal of Operational Research*, 169(3), 978–993. <https://doi.org/10.1016/j.ejor.2004.08.023>
- Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *Stochastic Algorithms: Foundations and Applications* (LNCS 5792, pp. 169–178). Springer. https://doi.org/10.1007/978-3-642-04944-6_14
- Yang, X. S. (2013). Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, 29(2), 175–184. <https://doi.org/10.1007/s00366-012-0254-1>
- Ye, B., Qin, F., & Zhang, J. (2025). Research on airline crew scheduling model for fatigue management. *Aerospace*, 12(12), 1116. <https://doi.org/10.3390/aerospace12121116>
- Zeghal, F. M., & Minoux, M. (2006). Modeling and solving a crew assignment problem in air transportation. *European Journal of Operational Research*, 175(1), 187–209. <https://doi.org/10.1016/j.ejor.2004.11.028>
- Zhao, M., Yang, S., Yang, X., & Wu, J. (2026). An efficient column generation approach for crew re-scheduling and recovery in urban rail transit systems under emergency conditions. *Expert Systems with Applications*, 299(Part B), 129993. <https://doi.org/10.1016/j.eswa.2025.129993>
- Zhou, S., Zhan, Z., Chen, Z., Kwong, S., & Zhang, J. (2021). A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11), 6784–6798. <https://doi.org/10.1109/TITS.2020.2994779>