The Aeronautical Journal (2026), **130**, pp. 2010–2035
doi:10.1017/aer.2026.10146

ROYAL AERONAUTICAL SOCIETY

RESEARCH ARTICLE

Matheuristic strategy development and benchmarking for explainable and efficient aircraft departure sequencing

Kadir Dönmez and Oguzhan Sahin

Aircraft Maintenance Department, Samsun University, Türkiye
Corresponding author: Kadir Dönmez; Email: kadir.donmez@samsun.edu.tr

Received: 20 March 2025; **Revised:** 30 January 2026; **Accepted:** 30 January 2026

Keywords: Aircraft departure sequencing; Air traffic management; Matheuristics; Metaheuristic algorithms; Mixed-integer linear programming

Abstract

This paper develops and compares four distinct constraint-aware and adaptive matheuristic approaches based on tabu search, simulated annealing, bacterial foraging optimisation and multi-ant colony optimisation. By integrating metaheuristic search with a mixed-integer linear programming (MILP) model, a hybrid framework is designed to optimise aircraft departure sequencing with rigorous precision. The core novelty lies in how this integration reconciles the exactness of mathematical programming with the interpretability of metaheuristics. Standard MILP models offer precise solutions but function as opaque 'black boxes', while standard metaheuristics operate via understandable moves but often rely on simplified feasibility checks. The proposed solution merges these strengths, employing adaptive metaheuristics to generate candidate sequences using interpretable moves while a restricted MILP acts as a high-fidelity evaluator. Fixing the sequence enables the MILP to focus solely on optimising continuous timings and holds. This ensures that every candidate step is strictly feasible and optimally timed, turning the solution process into a sequence of human-readable queue adjustments backed by the numerical precision of an exact solver. In a calibrated Antalya Airport case study with 40 aircraft, all algorithms attain the MILP optimum or remain within 1% of it, achieving a 29% reduction in hold fuel compared to first-come-first-served (FCFS). In denser 50-aircraft scenarios, the approach maintains high solution quality within feasible time limits, proving that explainability does not require sacrificing computational efficiency.

Nomenclature

A-CDM	Airport Collaborative Decision Making
A-SMGCS	Advanced Surface Movement Guidance and Control System
A320	Airbus A320 (Medium category in this study)
A333	Airbus A330-300 (Heavy category in this study)
AC	Aircraft (problem instance size, e.g. 40–50 AC)
ACO	Ant Colony Optimisation
ADS-B	Automatic Dependent Surveillance–Broadcast
AI	Artificial Intelligence
AIP	Aeronautical Information Publication
ATC	Air Traffic Control
ATCo	Air Traffic Controller
ATM	Air Traffic Management
BFOA	Bacterial Foraging Optimisation Algorithm
C550	Cessna 550 Citation II (Light category in this study)
CPS	Constrained Position Shifting
CPU	Central Processing Unit
DMAN	Departure Manager

© The Author(s), 2026. Published by Cambridge University Press on behalf of Royal Aeronautical Society.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

DSP	Departure Sequencing Problem
DSS	Decision Support System
EASA	European Union Aviation Safety Agency
EI	Emission Index (mass of pollutant per mass of fuel, typically g/kg fuel)
ETA	Estimated Time of Arrival
ETD	Estimated Time of Departure
FAA	Federal Aviation Administration
FCFS	First-Come-First-Served
GDSAA	General Directorate of State Airports Authority (Türkiye)
H/M/L	Heavy / Medium / Light aircraft categories
HC	Hydrocarbons (pollutant)
HMI	Human–Machine Interface
IATA	International Air Transport Association
ICAO	International Civil Aviation Organization
kg	kilogram
kt	knot (nautical miles per hour)
LNS	Large Neighbourhood Search
LP	Linear Programming
LTAI	Antalya Airport (ICAO location indicator)
MACO	Multi-Ant Colony Optimisation
MILP	Mixed-Integer Linear Programming
MPS	Maximum Position Shift
NOx	Nitrogen oxides (pollutant)
NP-hard	Non-deterministic Polynomial-time hard
R&D	Research and Development
RAM	Random Access Memory
SA	Simulated Annealing
STD	Scheduled Time of Departure
TS	Tabu Search
TSAT	Target Start-up Approval Time
TTOT	Target Take-Off Time

1.0 Introduction

The exponential expansion of international trade, coupled with the aggressive growth of both legacy and low-cost carriers, has led to a surge in global air traffic that strains the capacity of modern aviation infrastructure [1]. Within this congested environment, efficient air traffic management (ATM) is a necessity for operational safety and sustainability [2]. Among the various subsystems of ATM, the management of runway capacity, specifically aircraft scheduling, has emerged as a critical bottleneck.

In daily operations, arrival flights typically take precedence over departures. This inherent prioritisation often forces departing aircraft to wait on the ground, leading to queue spillbacks and substantial delays [3]. Consequently, the departure sequencing problem (DSP) becomes the focal point of airport efficiency. However, optimising the DSP is complex as it requires the simultaneous coordination of aircraft types, sizes, taxi speeds and critical ground phases from pushback to takeoff [4]. The process involves a rigid layer of constraints including wake turbulence separation [5], air traffic control (ATC) time windows, constrained position shifting (CPS) for airline equity and specific precedence rules [6–8].

Historically, research into solving the DSP has been polarised. On one side, exact algorithms like MILP are favoured for their ability to model detailed constraints and provide theoretically optimal solutions [9–11]. However, exact methods are computationally expensive for large-scale NP-hard instances [12, 13] and generally function as 'black boxes' that do not reveal the operational logic behind the schedule, creating a trust barrier for human controllers [14–16]. On the other side, heuristics and metaheuristics offer speed and understandable steps but traditionally rely on simplified feasibility checks. These simplifications often fail to capture system-wide dynamics such as optimal ground-holding strategies, potentially leading to sub-optimal plans [17–19].

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

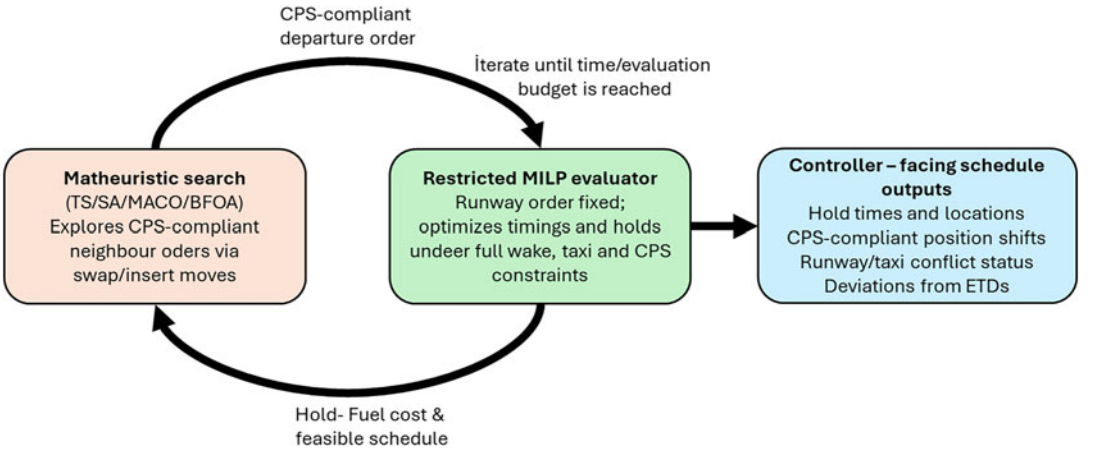


Figure 1. *Schematic of the matheuristic loop. The metaheuristic proposes a CPS-compliant departure order; the restricted MILP returns optimal timings and holds for that fixed order; the resulting hold-fuel objective is fed back to guide the next move.*

To resolve this dichotomy, this study introduces a hybrid MILP-in-the-loop matheuristic framework inspired by recent advances in explainable metaheuristics [20, 21] as illustrated in Fig. 1. We develop four specific constraint-aware and adaptive variants of tabu search (TS), simulated annealing (SA), multi-ant colony optimisation (MACO) and bacterial foraging optimisation (BFOA). Unlike standard implementations, these algorithms are engineered with adaptive mechanisms to efficiently explore the search space using interpretable moves while a restricted MILP evaluator guarantees rigorous validation. This aligns with findings that controllers prefer tools that match their mental models [22], bridging the gap between numerical precision and operational explainability.

2.0 Literature review

Matheuristics, a blend of mathematical programming and heuristic methods, have gained prominence in solving complex combinatorial optimisation problems across various domains [23]. This section provides a comprehensive overview of matheuristics in general routing and scheduling applications before focusing on specific implementations in the aviation domain and identifying the gaps addressed by this study.

2.1 Matheuristics in routing, scheduling and logistics

The integration of exact solvers within heuristic frameworks has proven particularly effective in logistics. Archetti and Speranza [24] classify these approaches into decomposition, improvement and column generation-based methods. Solyal and Süral [25] addressed the inventory routing problem using a matheuristic that sequentially solves different MILPs, demonstrating that this hybrid approach could outperform existing algorithms and provide new best solutions. Similarly, Boschetti and Maniezzo [26] applied a matheuristic to city logistics by combining a dual ascent procedure with a randomised constructive heuristic, which proved effective in real-world scenarios. Vadseth et al. [27] presented an iterative matheuristic for inventory routing that emphasised the maximum level replenishment policy and outperformed state-of-the-art exact methods on large benchmark instances. Furthermore, Moura et al. [28] introduced a hybrid approach for the combined bin packing and vehicle routing problem, demonstrating significant improvements in operational efficiency under real-world constraints.

In the broader context of transportation networks, Chouman and Crainic [29] developed a cutting-plane matheuristic that efficiently computes tight lower bounds, surpassing current best methods in

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

the literature. Villegas et al. [30] proposed a hybrid method for truck and trailer routing using a set-partitioning formulation that outperformed leading methods and highlighted the robustness of matheuristics. Additionally, Angelelli et al. [31] focused on freight forwarder services and validated their method on real data instances, showing its capability to offer high-quality solutions for large-scale problems within reasonable computing times.

In industrial scheduling, matheuristics often bridge the gap between theory and practice. Dang et al. [32] tackled parallel machine scheduling using a genetic algorithm combined with linear programming, showing significant improvements over current practices. Fanjul-Peyro et al. [33] proposed strategies for unrelated parallel machine scheduling that significantly outperformed mathematical models on medium-sized instances. Della Croce and Salassa [34] introduced a variable neighbourhood search-based matheuristic for nurse rostering and demonstrated that hybrid methods can handle complex resource constraints better than pure solvers. Finally, Karlsson et al. [35] presented a matheuristic for large-scale avionic scheduling in integrated modular systems, effectively solving large instances and providing valuable managerial insights.

2.2 Matheuristics in aviation operations

Within the aviation sector, matheuristics have been successfully adapted to various ground and airside operations. Silva et al. [36] addressed the aircraft landing problem (ALP) using a relax-and-fix approach that proved efficient for instances with a large number of aircraft and offered significant improvements over existing mathematical formulations. In maintenance operations, Zhang et al. [37] combined ant colony optimisation with set partitioning to handle flexible cruise times. Their approach significantly improved aircraft utilisation and demonstrated high accuracy when validated with Bureau of Transportation Statistics data. Karsu and Solyal [38] employed a matheuristic for gate assignment to minimise passenger walking distances, outperforming existing algorithms in computational tests. Delgado and Mora [39] applied column generation-based heuristics to air-cargo recovery and significantly improved profit margins by allowing flexible cargo assignment policies. Kyriakidou [40] addressed aircraft landing scheduling using variable neighbourhood search combined with interior-point methods, showing substantial improvements over traditional methods through statistical analysis.

2.3 The Gap in DSP: From black-box to explainable optimisation

Despite these successes, there is currently no matheuristic that embeds an exact MILP evaluator directly within the DSP as existing work relies predominantly on pure metaheuristics. Studies by Atkin et al. [41], Caprì and Ignaccolo [42] and Benlic et al. [43] typically enforce feasibility through forward timing rules or ad-hoc repairs rather than through an exact optimisation layer. More recently, constraint-guided search frameworks have been introduced to inject structural information [44], yet these still largely lack the rigorous guarantees of a MILP for complex ground-holding couplings.

Conversely, adoption barriers persist for exact methods due to their opacity. Even when a stand-alone MILP offers a detailed solution, the intermediate reordering logic remains hidden inside the branch-and-bound tree, forcing controllers to accept a 'black box' output without insight into the operational decisions that led to it. Recent work highlights that such opacity limits trust [15, 45] and that controllers prioritise tools aligning with their mental models [22]. To bridge this divide, this study draws on principles of explainable metaheuristics [20, 21], aiming to combine the transparency of trajectory-based search with the precision of mathematical programming.

3.0 Contribution of the study

To address the identified gap, this paper introduces a structured MILP-in-the-loop framework. Unlike previous studies that rely on simplified feasibility checks, we embed a restricted MILP as a high-fidelity

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

evaluator within four enhanced metaheuristics. This architecture allows us to rigorously value candidate solutions while maintaining a transparent search process.

The main contributions of the study are:

- **MILP-in-the-loop matheuristic framework:** We propose a hybrid architecture where adaptive metaheuristics generate candidate sequences and a restricted MILP performs the evaluation. This novel design guarantees that every proposed solution strictly adheres to wake-turbulence, separation and ground-holding constraints, overcoming the feasibility limitations of standard heuristics (see Fig. 1).
- **Constraint-aware and explainable search:** By restricting the MILP to evaluation rather than full search, we expose the decision process. The framework operates via interpretable neighbourhood operators (e.g. CPS-compliant swaps, priority inserts), creating a traceable optimisation log where specific moves are directly linked to hold-fuel savings, addressing the transparency requirements of ATM systems.
- **Systematic benchmarking of adaptive algorithms:** We develop and rigorously compare four metaheuristic variants (enhanced TS, SA, MACO, BFOA) under a unified MILP evaluator. This controlled environment isolates the search efficiency of each algorithm, providing clear insights into their convergence behaviours under identical operational constraints.
- **Scalability and real-world validation:** Validated on calibrated data from Antalya Airport, the framework demonstrates superior scalability compared to stand-alone exact methods. In high-density scenarios where full MILP solvers struggle to close the optimality gap, the proposed algorithms converge to near-optimal solutions within tactical time limits, proving viability for real-time decision support.
- **Decision-support ready outputs:** Beyond sequencing, the model generates granular outputs including exact hold times, queue times, delays and disaggregated fuel and emission estimates (HC, CO, NOx). These metrics are designed to be directly consumable by departure manager (DMAN) tools for environmentally conscious planning.

4.0 Methodology

This section details the proposed matheuristic framework designed to solve the DSP. The central architectural principle is the decoupling of the *combinatorial* decision (runway sequencing) from the *continuous* decision (timing and holding). Four enhanced metaheuristics – enhanced TS, SA, MACO and BFOA – are responsible for exploring the discrete space of CPS-compliant departure orders, while a restricted MILP model acts as a sub-routine to optimise timings for each proposed order.

4.1 Evaluation of fixed sequences

The core innovation of our search strategy is the replacement of heuristic objective functions with an exact MILP evaluator. For any candidate permutation generated by the metaheuristic, we solve a simplified version of the DSP model where the binary ordering variables are fixed. This transforms the NP-hard sequencing problem into a computationally tractable linear programming (LP) or reduced MILP problem focusing solely on resource allocation (Fig. 1).

The evaluation protocol proceeds as follows:

1) The metaheuristic generates a candidate neighbour order using a constraint-aware move (e.g. a CPS-compliant swap).
2) This candidate order is passed to the restricted MILP, where the binary variables defining the sequence are fixed.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

3) The restricted MILP optimises the remaining continuous variables: reaching times to each point $\{t_i^p\}$ and hold durations $\{s_i^p\}$, ensuring optimal distribution of ground holds across the taxiway network.
4) The MILP returns the optimal objective value (total hold fuel) and feasibility status, which are fed back to the metaheuristic to guide the search.
5) Steps 1–4 repeat until the time or iteration budget is exhausted.

Frequency of evaluation: In each iteration, the metaheuristic filters candidates using quick feasibility checks (e.g. CPS bounds). Only promising candidates trigger a restricted-MILP solve. To ensure computational efficiency compatible with real-time operations, the MILP solver is warm-started from the previous solution, and the total number of MILP calls is capped by a global budget.

Computing environment: All experiments were performed on a Windows workstation equipped with an 11th Gen Intel® Core™ i7-1185G7 CPU (3.00 GHz) and 64 GB of RAM. The algorithms were implemented in Python 3.11 using Pyomo and solved with *Gurobi Optimizer* (v12.0.3). Default solver parameters were used, except for the TimeLimit, which was kept consistent across all runs to ensure comparability.

4.2 MILP model

We formulate the DSP on a time–space representation of the airport surface. The model minimises total ground-hold fuel while enforcing runway wake separations, multi-point ground conflict separations and CPS. Below we define all sets, indices, parameters and decision variables before presenting the constraints.

Sets and indices:

- $i, j \in I$: set of aircraft.
- P: set of surface/operation points (pushback, conflict points, runway); for clarity we partition $P = P^{\text{conf}} \cup P^{\text{rwy}} \cup P^{\text{oth}}$.
- $P_i \subseteq P$: ordered path of aircraft i (only the points actually traversed by i). We write $q \prec_i p$ if point q precedes p on i's path.
- O: wake-turbulence types (Heavy/Medium/Light); $\tau_i \in O$ denotes the type of aircraft i.

Parameters:

- α_i [s]: system entry (ready) time of aircraft i.
- γ_i [s]: prescribed gate/pushback delay for i (treated as a parameter).
- θ_i^p [s]: nominal taxi time for i to reach point $p \in P_i$ from the previous point on i's path.
- $\bar{s}_i^p$ [s]: maximum allowable local ground hold of i at point $p \in P_i$.
- Δ_p [s]: required time separation at conflict point $p \in P^{\text{conf}}$ (point-dependent).
- W_{τ_i,τ_j} [s]: wake separation at the runway when a type-τ_i aircraft is followed by a type-τ_j aircraft.
- ϕ_i [kg/s]: hold-fuel rate parameter used to evaluate ground holds of aircraft i.
- MPS: constrained position-shifting limit (max allowed deviation from scheduled order).
- $n = |I|$, $M > 0$: a sufficiently large constant (big-M).

Decision variables:

- t_i^p [s]: time aircraft i reaches point $p \in P_i$.
- s_i^p [s]: local ground-hold (delay) of aircraft i at point $p \in P_i$.
- $x_{ij}^p \in \{0, 1\}$: ordering at point p (1 if i is before j at p), defined only for $p \in P_i \cap P_j$.
- π_i^p: position of aircraft i at point $p \in P_i$; π_i^{std} scheduled position; π_i^{fin} final (runway) position.
- F_i [kg]: fuel consumed by aircraft i due to ground holds.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

Initialization and propagation on paths

$$t_i^0 = \alpha_i + \gamma_i \qquad \forall i \in I \tag{1}$$

$$t_i^p = t_i^0 + \sum_{q \prec_i p} \left(\theta_i^q + s_i^q\right) \qquad \forall i \in I,\ \forall p \in P_i \tag{2}$$

$$0 \quad \leq \quad s_i^p \leq \bar{s}_i^p \qquad \forall i \in I,\ \forall p \in P_i \tag{3}$$

Positions, CPS, and consistency

$$\pi_i^p = n - \sum_{\substack{j \in I \\ j \neq i}} x_{ij}^p \qquad \forall i \in I, \quad \forall p \in P_i \tag{4}$$

$$\pi_i^p \ \leq \ \pi_i^q \qquad \forall i \in I,\ \forall p, q \in P_i \colon\ p \prec_i q \tag{5}$$

$$\left|\pi_i^{\text{fin}} - \pi_i^{\text{std}}\right| \ \leq \ \text{MPS} \qquad \forall i \in I \tag{6}$$

Conflict and runway separations (only where aircraft meet)

$$t_j^p - t_i^p \ \geqslant \ \Delta_p - M\left(1 - x_{ij}^p\right) \qquad \forall p \in P^{\text{conf}},\ \forall i \neq j \colon\ p \in P_i \cap P_j \tag{7}$$

$$t_i^p - t_j^p \ \geqslant \ \Delta_p - M\, x_{ij}^p \qquad \forall p \in P^{\text{conf}},\ \forall i \neq j \colon\ p \in P_i \cap P_j \tag{8}$$

$$t_j^p - t_i^p \ \geqslant \ W_{\tau_i,\tau_j} - M\left(1 - x_{ij}^p\right) \qquad \forall p \in P^{\text{rwy}},\ \forall i \neq j \colon\ p \in P_i \cap P_j \tag{9}$$

$$t_i^p - t_j^p \ \geqslant \ W_{\tau_j,\tau_i} - M\, x_{ij}^p \qquad \forall p \in P^{\text{rwy}},\ \forall i \neq j \colon\ p \in P_i \cap P_j \tag{10}$$

Hold Fuel accounting and objective

$$F_i \ = \ \phi_i \sum_{p \in P_i} s_i^p \qquad \forall i \in I \tag{11}$$

$$\min \quad \sum_{i \in I} F_i \tag{12}$$

Model overview: Equation (1) sets each aircraft's pushback time from ready time and a fixed gate delay (parameter). Equation (2) propagates times along the *aircraft-specific* path P_i, summing nominal taxi and optimised holds only over points actually traversed. Bounds in Equation (3) cap local holds. Positions π are derived from order binaries x at the points where aircraft meet (Equation (4)), kept nondecreasing along each path (Equation (5)), and restricted by CPS (Equation (6)) to stay close to the scheduled order. Conflict-point separations use the point-dependent parameter Δ_p (Equations (7) and (8)); runway separations use the wake matrix W_{τ_i,τ_j} (Equations (9) and (10)). Fuel is linear in holds (Equation (11)), and the objective minimises total hold fuel (Equation (12)). Emissions (HC/CO/NOx) are reported post-solve from F_i and type-specific indices. Importantly, if an aircraft does not traverse a point (e.g. different apron), that point is not in P_i, so timing and separation constraints never apply spuriously. The big-M terms appear only in disjunctive separations and do not affect propagation or hold-fuel accounting.

4.3 Tabu search (TS)

The TS is a metaheuristic that explores the solution space by iteratively moving from one potential solution to another, avoiding cycles through a tabu list [46–48]. Our enhanced TS incorporates custom features such as dynamic, reactive tabu tenure adjustment, priority pairs based on ETD values, CPS compliance and a small elite pool with path-relinking/micro-scale large neighbourhood search (LNS) to improve its effectiveness in aircraft sequencing. Dynamic tabu tenure is updated reactively: it increases

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

when the search stagnates and may be relaxed again after diversification steps. Priority pairs are grouped based on estimated time of departure (ETD) values, guiding the search towards feasible solutions that respect temporal constraints. CPS compliance ensures that position shifts for each aircraft do not exceed a certain limit, maintaining fairness and minimising controller workload. The TS also maintains a small elite pool of best sequences and periodically applies path-relinking and a micro-scale LNS between the current and elite solutions, which intensifies the search without violating CPS.

The system initialises the search state, best solution and parameters, generates a neighbourhood of CPS-compliant candidates using allowed swap/insert moves, and evaluates only a small top-k subset of them using the MILP model based on heuristic scores, in order to respect the runtime budget. Moves are checked against the tabu list and the aspiration criterion, and the tabu list is updated to prevent cycling back to recent moves. The system adjusts tabu tenure reactively and performs random CPS-compliant shuffles, together with occasional elite-based path-relinking/micro-LNS, to escape local optima. The search stops when a global time limit, a MILP evaluation budget or the maximum number of non-improving iterations is reached. The pseudo-code of the enhanced TS approach is given in Algorithm 1.

4.4 Simulated annealing (SA)

The SA algorithm is a simulation technique that draws inspiration from metallurgy, where materials are heated and cooled to achieve a stable crystalline structure [49]. It uses a temperature parameter to accept worse solutions and escape local optima. Custom enhancements include an adaptive temperature schedule, CPS-compliant moves and priority-based moves, which balance exploration and exploitation effectively.

The enhanced SA system involves initialisation, generation of new CPS-compliant solutions, evaluation via the restricted MILP, acceptance criteria, temperature update and stagnation check. It balances exploration and exploitation through an adaptive temperature schedule, avoiding premature convergence. CPS-compliant moves ensure feasible and practical solutions, even under complex operational constraints. The system also performs a shuffle if stagnation is detected to escape local optima. In practice, the SA loop is additionally capped by a global wall-clock time limit and a maximum number of MILP evaluations to keep its runtime comparable with the other metaheuristics. The pseudo-code of the enhanced SA approach is given in Algorithm 2.

4.5 Multi-ant colony optimisation (MACO)

Ant colony optimisation (ACO) is a method inspired by ants' behaviour, where they deposit pheromones on paths to reinforce efficient routes [50]. Multi-ACO uses one or more colonies, each searching independently but sharing information periodically, to enhance global search capability [51]. In our implementation, the number of colonies is kept small, which is sufficient to maintain diversity while staying within the restricted-MILP evaluation budget. Custom enhancements include multiple colonies running in parallel when needed, dynamic pheromone updating and periodic inter-colony communication to propagate good solutions across the search space. These features enhance diversity and reduce the likelihood of premature convergence.

The initialisation process involves setting up multiple colonies, each with its own pheromone matrix and initial state. Ants in a colony use pheromone information and heuristics like CPS compliance and priority pairs to construct solutions. Each constructed sequence is then evaluated either by the restricted MILP or a fast-timing surrogate, depending on the configuration. The pheromone update is a system where pheromones are updated based on the quality of a solution to guide future ants toward better solutions. Inter-colony communication involves sharing the best solutions among colonies to facilitate the spread of beneficial solutions. The global best update updates the global best solution across all colonies. The pseudo-code of the enhanced MACO approach is given in Algorithm 3.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

Algorithm 1 Enhanced Tabu Search

```
Input: InitialState, MaxIterations, InitialTabuTenure, MaxNoImprovement, PriorityPairs,
TimeLimit, MaxEvaluations
Set CurrentSolution = InitialState Set
BestSolution = InitialState
Set ElitePool = {InitialState}
Set TabuList = Ø
Set NoImprovementCounter = 0
Set Iteration = 0
Set TabuTenure = InitialTabuTenure
Set StartTime = CurrentTime()
Set NumEvaluations = 0
while (Iteration < MaxIterations) and (NoImprovementCounter < MaxNoImprovement) and
(CurrentTime() − StartTime < TimeLimit) and (NumEvaluations < MaxEvaluations) do
    Iteration++
    Neighbourhood = GenerateNeighborSolutions(CurrentSolution, PriorityPairs)
    Candidates = SelectTopK(Neighbourhood)
    ImprovedThisIteration = False
    for each Candidate in Candidates do
      if MoveIsTabu(Candidate. Move) == False or IsAspirationCriterionMet(Candidate,
      BestSolution) then
        HoldFuelConsumption = EvaluateUsingMILP(Candidate)
        NumEvaluations++
        if HoldFuelConsumption < BestSolution. Cost
        then
          BestSolution = Candidate
          ImprovedThisIteration = True
          UpdateElitePool(BestSolution, ElitePool)
        end if
        CurrentSolution = Candidate
        AddMoveToTabuList(Candidate. Move, TabuTenure)
        break
      end if
    end for
    if ImprovedThisIteration then
      NoImprovementCounter = 0
    else
    NoImprovementCounter++
  end if
  if TabuListSize > TabuTenure then
RemoveOldestMoveFromTabuList()
end if
If NoImprovementCounter == TabuTenure then
   PerformCPSCompliantShuffle(CurrentSolution)
   TabuTenure = AdjustTabuTenure(TabuTenure)
   NoImprovementCounter = 0
end if
If ApplyPathRelinking() then
   CurrentSolution = PathRelinkingAndMicroLNS(CurrentSolution, ElitePool)
end if
end while
Return BestSolution
```

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

Algorithm 2. Enhanced Simulated Annealing

```
Input: InitialState, Tmax, Tmin, Alpha, MaxStagnation, MaxNoImprovement, TimeLimit, MaxEvaluations
Set CurrentSolution = InitialState
Set BestSolution = InitialState
Set Temperature = Tmax
Set Iteration = 0
Set StagnationCounter = 0
Set NoImprovementCounter = 0
Set StartTime = CurrentTime()
Set NumEvaluations = 0
while (Temperature > Tmin) and (NoImprovementCounter < MaxNoImprovement) and (CurrentTime() − StartTime < TimeLimit) and (NumEvaluations < MaxEvaluations) do
 Iteration++
 NewSolution = GenerateCPSCompliantMove(CurrentSolution)
 NewCost = EvaluateUsingMILP(NewSolution)
 NumEvaluations++
 DeltaCost = NewCost − CurrentSolution. Cost
 if (NewCost < CurrentSolution. Cost) or (Random(0, 1) < exp(-DeltaCost/Temperature)) then
  CurrentSolution = NewSolution
  if NewCost < BestSolution. Cost then
   BestSolution = NewSolution
   StagnationCounter = 0
   NoImprovementCounter = 0
  else
   StagnationCounter++
   NoImprovementCounter++
  end if
 else
   StagnationCounter++
   NoImprovementCounter++
 end if
 Temperature = UpdateTemperature(Tmax, Tmin, Iteration, Alpha)
 if StagnationCounter >= MaxStagnation then
   PerformCPSCompliantShuffle(CurrentSolution)
   StagnationCounter = 0
end if
end while
Return BestSolution
```

4.6 Bacterial foraging optimisation (BFOA)

BFOA is a method that uses bacteria's foraging behaviour to find optimal solutions [52]. It involves tumbling and swimming bacteria to find the most nutritious regions. BFOA's custom enhancements include CPS compliance, adaptive swimming length and dynamic reproduction and elimination-dispersion phases. These features ensure feasible solutions and maintain diversity through adaptive reproduction and elimination-dispersion phases (see Ref. [53] for further information about BFOA).

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

Algorithm 3. Enhanced Multi-Ant Colony Optimisation

```
Input: InitialState, NumColonies, NumAnts, MaxIterations, EvaporationRate,
PheromoneIntensity
Initialise Colonies[NumColonies] with InitialState
Set GlobalBestSolution = InitialState
for Iteration = 1 to MaxIterations do
  for each Colony in Colonies do
    for each Ant in Colony do
      Solution = ConstructSolutionUsingPheromonesAndHeuristics(Colony)
      Evaluate Solution using MILP
      if Solution is better than Colony. BestSolution then
        Colony. BestSolution = Solution
        UpdatePheromones(Solution,Colony)
      end if
    end for
  end for
  ShareGlobalBestAmongColonies(Colonies, GlobalBestSolution)
  for each Colony in Colonies do
    if Colony. BestSolution is better than GlobalBestSolution then
      GlobalBestSolution = Colony. BestSolution
    end if
  end for
end for
Return GlobalBestSolution
```

The process involves initialising the bacteria population with feasible solutions and setting parameters for different phases. Each bacterium moves randomly through the solution space, performing a tumble, and then swims toward improved solutions. Similar to the other metaheuristics, BFOA uses the same CPS-compliant swap/insert operators and ETD-based priority pairs when generating new candidate sequences. Reproduction involves retaining and cloning the most efficient bacteria, allowing the rest to be repurposed to concentrate on the most promising areas of the solution space. The elimination-dispersion method introduces diversity by randomly eliminating and redistributing certain bacteria. The global best update updates the global best solution for all bacteria. The enhanced BFOA prevents premature convergence and ensures solutions respect operational constraints like CPS and priority pairs, making them suitable for real-world implementation. The pseudo-code of the enhanced BFOA approach is given in Algorithm 4.

4.7 Custom features

In our matheuristic approach, several custom features are shared across all algorithms, while others are tailored to specific metaheuristics based on their inherent structure and operational characteristics. Priority pair selection, CPS constraints and hold-fuel-weighted swap logic are universally relevant and applied across all four metaheuristics. In contrast, features like probabilistic acceptance are used only where they are particularly effective (e.g. in SA), while random shuffling is incorporated into algorithms like TS and SA to handle stagnation effectively. This selective application of features ensures that each algorithm operates optimally, leveraging the strengths of its design while remaining adaptable to the DSP.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

Algorithm 4. Enhanced Bacterial Foraging Optimisation

```
Input: InitialState, NumBacteria, NumEliminationDispersion, NumReproduction,
NumChemotaxis, SwimLength, StepSize
Initialise Bacteria[NumBacteria] with InitialState
Set GlobalBestSolution = InitialState
for each Elimination-Dispersion Phase do
  for each Reproduction Phase do
    for each Chemotaxis Step do
      for each Bacterium in Bacteria do
        NewSolution = PerformTumble(Bacterium)
        NewCost = EvaluateUsingMILP(NewSolution)
        if NewCost < Bacterium. BestCost then
          Bacterium. Position = NewSolution
          Bacterium. BestCost = NewCost
          SwimCount = 0
          while SwimCount < SwimLength do
            NewSolution = PerformSwim(Bacterium)
            NewCost = EvaluateUsingMILP(NewSolution)
            if NewCost < Bacterium. BestCost then
              Bacterium. Position = NewSolution
              Bacterium. BestCost = NewCost
              SwimCount++
            else
              break
            end if
          end while
        end if
      end for
    end for
    Sort Bacteria by Cost
    Keep best half and clone them; discard the worst half
  end for
  for each Bacterium do
    if Random(0, 1) < DispersionProbability then
    Randomly initialise new position
  end if
 end for
end for
for each Bacterium do
  if Bacterium. BestCost < GlobalBestSolution. Cost then
    GlobalBestSolution = Bacterium
 end if
end for
Return GlobalBestSolution
```

Priority pair selection: First, priority pairs are determined based on the ETD closeness between aircraft. Aircraft with ETDs within a specified threshold (ETD_threshold = 90 s) are considered priority pairs. This strategy ensures that the algorithm focuses on swaps between aircraft with close departure times, which are likely to have a more significant impact on the overall solution. When no priority pairs remain, the algorithm selects two random aircraft to either swap or insert, maintaining diversity in

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

solution space exploration. The same priority-pair generator is shared across TS, SA, MACO and BFOA so that all algorithms focus first on ETD-critical swaps.

Swap and insert operations: At each step, the algorithm selects either a swap or an insert move. A swap exchanges the positions of two aircraft in the sequence, whereas an insert removes an aircraft from its current position and reinserts it at another position.

CPS compliance: Both swap and insert operations respect CPS constraints, ensuring that the new positions of the aircraft stay within a ± 3 position range relative to their ETD-based positions. This approach helps maintain a feasible sequence while staying close to the FCFS order.

Hold-fuel weight-based swap logic: To further refine the swaps, all metaheuristics use a hold-fuel weight-based swap probability when proposing swap moves. Aircraft with higher hold-fuel weights are given priority in swaps, as these have a more substantial impact on hold-fuel consumption and overall solution quality. The swap probability is computed as the ratio of the fuel weight of the first aircraft to the combined hold-fuel weights of both aircraft:

$$\text{swap_probability} = \frac{\text{holdfuel_weight}_a}{\text{holdfuel_weight}_a + \text{holdfuel_weight}_b}.$$

This logic ensures that more hold-fuel-efficient decisions are prioritised across all neighbourhood generators. The hold-fuel weight data for each aircraft type is obtained from the International Civil Aviation Organisation (ICAO) engine emissions databank [54], providing accurate hold-fuel consumption estimates.

Selection of aircraft for swap: Two aircraft, denoted a and b, are selected either randomly or based on priority pairs. Their hold-fuel weights are calculated based on their respective aircraft types, and the probability of performing a swap is then determined by the above formula. If a random number generated is less than the calculated swap probability, the swap is executed. Otherwise, the algorithm either performs a swap or an insert operation as part of its regular process.

Probabilistic acceptance of solutions for SA: The SA algorithm employs a probabilistic acceptance function based on the **Boltzmann distribution** to determine whether to accept a newly generated sequence. This function allows worse solutions to be accepted with a probability that decreases as the temperature drops, helping the algorithm escape local minima. The acceptance probability is computed as:

$$P = e^{-\Delta E/T}$$

where ΔE is the change in energy (hold-fuel consumption) between the current and the new solution, and T is the current temperature. The probabilistic acceptance mechanism helps explore a wider solution space, increasing the chances of escaping local optima and finding the global optimum.

Stagnation and random shuffle: If the algorithm detects stagnation (i.e. no improvement over a set number of iterations), it performs a CPS-compliant random shuffle to explore new regions of the solution space. This operation swaps aircraft positions within the CPS constraints to ensure feasibility while introducing larger changes to the sequence. Random shuffling is a less-frequent operation compared to regular swaps but is crucial in preventing the algorithm from being stuck in local optima. While the regular swap makes small incremental changes, the random shuffle introduces more substantial variations when necessary.

Termination conditions: All metaheuristics incorporate three key termination criteria: a wall-clock time limit (e.g. 600 s), a cap on the total number of restricted-MILP evaluations per run and a maximum number of iterations without improvement. These criteria ensure that the algorithms run within a practical duration, maintain a fair MILP budget relative to the standalone MILP and prevent infinite loops if no improvement is observed. Once any of these conditions is met, the algorithm terminates and returns the best solution found.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

Table 1. *Hold–fuel rate and emission factors used*

Type	Hold–fuel (kg/s)	Emissions (g/kg fuel): HC/CO/NOx
A333 (H)	0.227	0.92/15.05/4.77
A320 (M)	0.104	4.40/44.10/3.20
C550 (L)	0.026	40.00/97.00/2.63

Nominal taxi speed 15 kt (≈7.72 m/s) for all types.

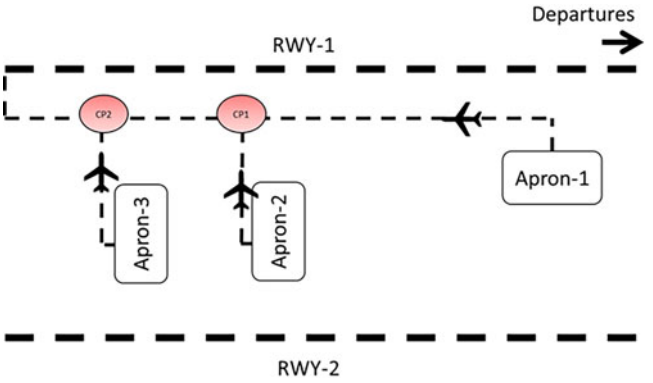


Figure 2. *Ground layout of case airport.*

5.0 Case study

Antalya Airport (LTAI), Turkey's third busiest airport, consistently experiences high delay rates exceeding 70%, with over 400 scheduled departures daily, as evidenced by ADS-B data from Flightradar [55]. This issue largely stems from the prioritisation of arriving flights in ATM protocols. To mitigate further delays, precise scheduling of departures is essential. Serving both domestic and international flights, Antalya Airport faces significant traffic peaks, particularly during the summer season, as reported by GDSAA Turkey [56]. The airport infrastructure includes two parallel runways, adjacent taxiways and three active aprons, as illustrated in Fig. 2 [57].

In this study, we focus on the departure operations at Antalya Airport, specifically on a single runway. As presented in Fig. 2, the airport's ground operations involve two primary conflict points, in addition to the queue of aircraft awaiting takeoff. The first conflict point is at the intersection of aircraft from Apron-1 and Apron-2, while the second involves aircraft from all three aprons. Our analysis includes three types of aircraft: the Airbus A330-300 (A333), the Airbus A320 and the Cessna 550 Citation II, representing high (H), medium (M) and low (L) categories for hold-fuel consumption and emission calculations, based on the ICAO engine emission data bank [54], respectively summarised in Table 1. Although emission factors are listed to characterise the aircraft types, detailed emission reporting is considered out of the scope of this study; given that emissions are directly calculated as a linear function of fuel consumption, the analysis focuses primarily on fuel metrics.

The model tests various scheduled departure times (STD) for 40 and 50 aircraft per hour. Departure time distributions, taxi speeds and apron allocations were determined using a uniform distribution to closely replicate real-world operational conditions.

The case study evaluates the effectiveness of a matheuristic approach in optimising departure sequencing at an airport with complex ground operations. The airport, which serves both domestic and international flights, faces traffic peaks during the summer season. The study modeled the airport's departure operations using a single runway and ground layout, incorporating hold-fuel consumption and emission calculations from three different aircraft categories based on ICAO engine emission data bank [58]. The aircraft classifications include heavy category (Airbus A330-300), medium category (Airbus A320) and light category (Cessna 550 Citation II) to represent these categories.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

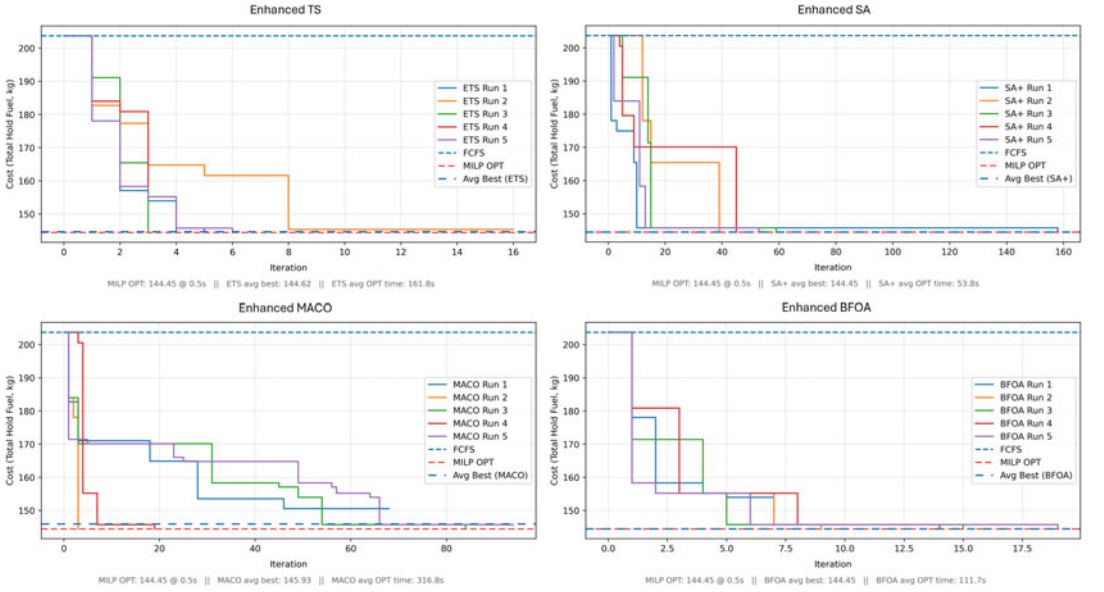


Figure 3. *Algorithm results for each run for sample 1 (40 AC scenario). For fair benchmarking, a uniform 600 s wall-clock time limit was imposed on both the individual metaheuristic runs and the standalone MILP solver. Iteration counts, therefore reflect method-dependent iteration costs.*

6.0 Results

We optimise a *static batch* of 40–50 departures at once. The 'horizon' is the time window that spans this batch, from the earliest ready time to the latest (plus fixed taxi segments). Thus, 40–50 is the *number of aircraft per solve*, not an aircraft-per-hour rate. We initially solved scenarios involving 40 aircraft for an hour horizon, using an ETD sample for each algorithm. The ETD scenario was tested with five runs per algorithm to comprehensively evaluate their solution performance. FCFS and optimal results were also obtained and presented to provide a benchmark for comparison.

Figure 3 shows the iterative convergence results of the four algorithms for Sample 1, which includes 40 AC demand with a 600-s solution time limit.

The convergence patterns illustrated in Fig. 3 validate the robustness of the proposed hybrid matheuristic framework across diverse search strategies. In general, algorithms with aggressive exploitation characteristics, specifically TS and BFOA, demonstrated rapid convergence, locking into near-optimal regions within the very first iterations. In contrast, SA and MACO exhibited a more gradual convergence profile, reflecting their inherent exploration mechanisms; probabilistic acceptance in SA and pheromone-based learning in MACO before stabilising at high-quality solutions. Despite these distinct search trajectories, all four methods successfully converged to the vicinity of the global optimum, confirming the effectiveness of the restricted MILP evaluator in guiding the optimisation process.

Unless otherwise stated, each metaheuristic is executed five *independent* times on the *same* DSP instance. All inputs and settings are fixed across runs: ETDs, airport layout and parameters, CPS limit (± 3), algorithm hyperparameters and the common MILP-based evaluator (same solver and wall-clock budget). The only difference between runs is the random seed driving stochastic choices (e.g. selection of CPS-compliant swap/insert moves and occasional shuffles in TS/SA, acceptance draws in SA, path construction and pheromone sampling in MACO and chemotaxis/dispersion choices in BFOA).

Figures 3 and 4 highlight the superior performance of the proposed algorithms against the FCFS baseline (203.7 kg). The enhanced SA and enhanced BFOA algorithms achieved the theoretical optimal value of 144.45 kg in all five runs, demonstrating 100% consistency. The enhanced TS algorithm also performed exceptionally well, with a best solution matching the optimal (144.45 kg) and an average

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

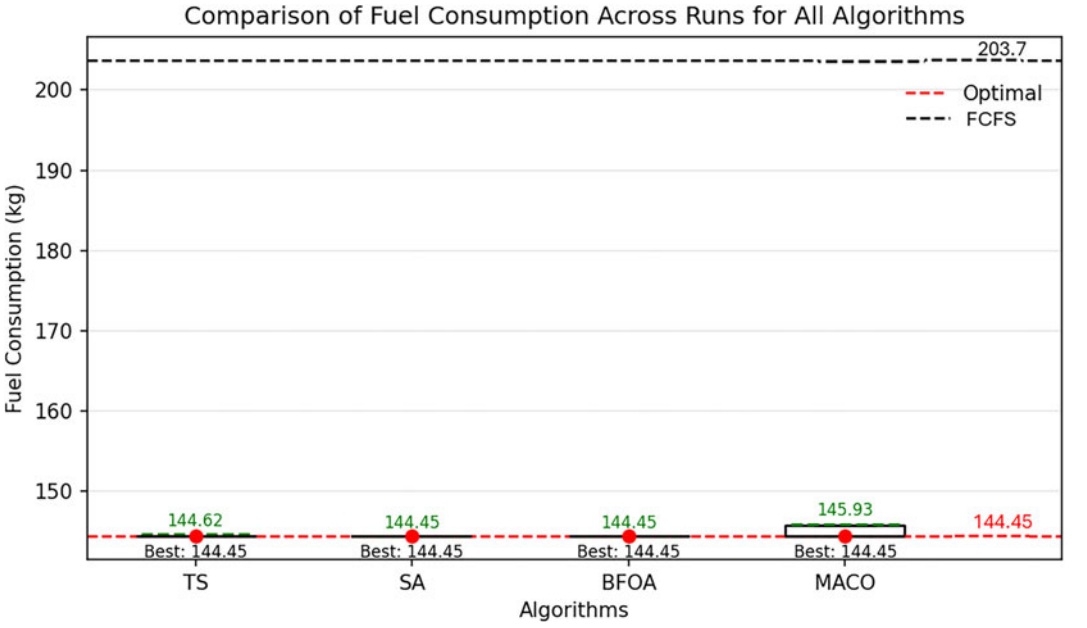


Figure 4. *Box-plot of algorithm results (hold-fuel consumption) for 40 AC sample. For fair benchmarking, a uniform 600 s wall-clock time limit was imposed on both the individual metaheuristic runs and the standalone MILP solver.*

hold fuel of 144.62 kg (only 0.11% deviation). The enhanced MACO achieved an average of 145.93 kg (approx. 1.0% deviation) and successfully found the optimal solution in its best run.

Regarding computational efficiency, the algorithms demonstrated the capability to converge to optimal solutions well within the imposed 600-s (10-min) time limit. The average times to reach the best solutions were 53.8 s for SA, 111.7 s for BFOA, 161.8 s for TS and 316.8 s for MACO. Although these durations exceed the sub-second solve time of a standalone MILP for this specific small-scale instance, they remain highly compatible with the pre-tactical planning horizon (typically spanning 15 min to several hours). In this operational context, a computation time of 1 to 5 min is negligible, especially given the trade-off: unlike the 'black-box' nature of a direct MILP solver, the iterative matheuristic process provides a transparent, traceable log of decision-making (e.g. specific swaps), which is essential for explainability in real-world ATM systems. Overall, all four algorithms provided a reduction of approximately 29.1% compared to the FCFS baseline, proving both the robustness and operational feasibility of the matheuristic evaluator.

To evaluate the limits of the proposed approach, we solved a high-density 50-aircraft scenario under a strict 600-s time limit. The FCFS baseline for this instance incurred a total hold-fuel consumption of 974.44 kg. The classical MILP solver, despite failing to prove global optimality (optimality gap 40.31%), established a strong 'best incumbent' solution of 520.10 kg.

Figure 5 illustrates the performance of the matheuristic algorithms. Among them, the MACO achieved the closest result to the MILP benchmark with a best solution of 539.62 kg (gap 3.75%), followed by the TS at 545.18 kg (gap 4.82%). SA and BFOA algorithms lagged further behind, with gaps of approximately 14.8% and 10.9%, respectively.

This experiment highlights a critical computational trade-off. In the 40-aircraft scenario, the 'restricted MILP evaluator' was fast enough to allow extensive search exploration. However, in the 50-aircraft scenario, the computational cost of solving a MILP sub-problem at *every* iteration increased significantly. This overhead limited the total number of iterations the metaheuristics could perform within the 600-s budget, preventing them from surpassing the standalone MILP's incumbent . Consequently,

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

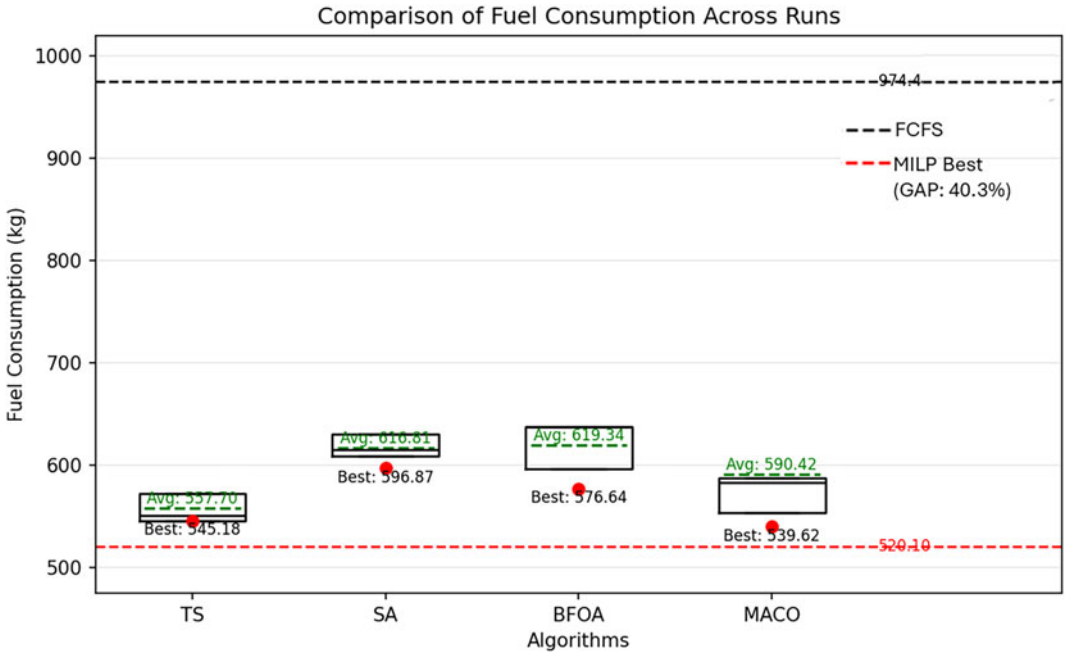


Figure 5. *Box-plot of algorithm results (hold-fuel consumption) for 50 AC sample comparing against FCFS baseline. For fair benchmarking, a uniform 600 s wall-clock time limit was imposed on both the individual metaheuristic runs and the standalone MILP solver.*

for very high-density instances, the pure matheuristic approach faces a scalability challenge due to the cumulative cost of its hybrid evaluator.

6.1 Sensitivity to evaluator time limit

To assess how much the matheuristics benefit from a larger MILP evaluator budget, we repeated the 50–aircraft experiments by doubling the time limit from 600 s (Fig. 5) to 1,200 s (Fig. 6). Overall, all four algorithms obtain lower fuel values when given more time, but the gains are incremental compared with the large improvement they already achieve over the FCFS baseline (974.44 kg). With a 600 s limit, the best solutions found by TS, SA, BFOA and MACO lie between 545.18 kg and 539.62 kg, i.e. roughly 40–45% below FCFS and within about 4–6% of the MILP optimum at 520.10 kg. When the limit is increased to 1200 s, TS benefits the most: its best run improves to 525.32 kg (about 46% below FCFS and only ≈1% above the MILP optimum), and its average fuel drops by roughly 4%. SA and BFOA also tighten their distributions, reducing average fuel by about 6–7% and bringing their best runs to within 3–6% of the optimum. MACO shows smaller marginal gains in the average (around 2%), and due to stochastic variability its single best run is slightly worse than under the 600 s setting, although it remains close to the other methods. These results indicate that a moderate evaluator budget (600 s) already captures most of the achievable savings, while a larger budget mainly sharpens the solutions and allows at least one algorithm (TS) to approach the offline MILP optimum, without changing the qualitative behaviour or interpretability of the matheuristic search.

6.2 Operational traceability and explainability analysis

A major barrier to the adoption of automated decision-support tools in ATM is the black-box character of traditional optimisation. A stand-alone MILP solver returns an optimal sequence and timing vector, but

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

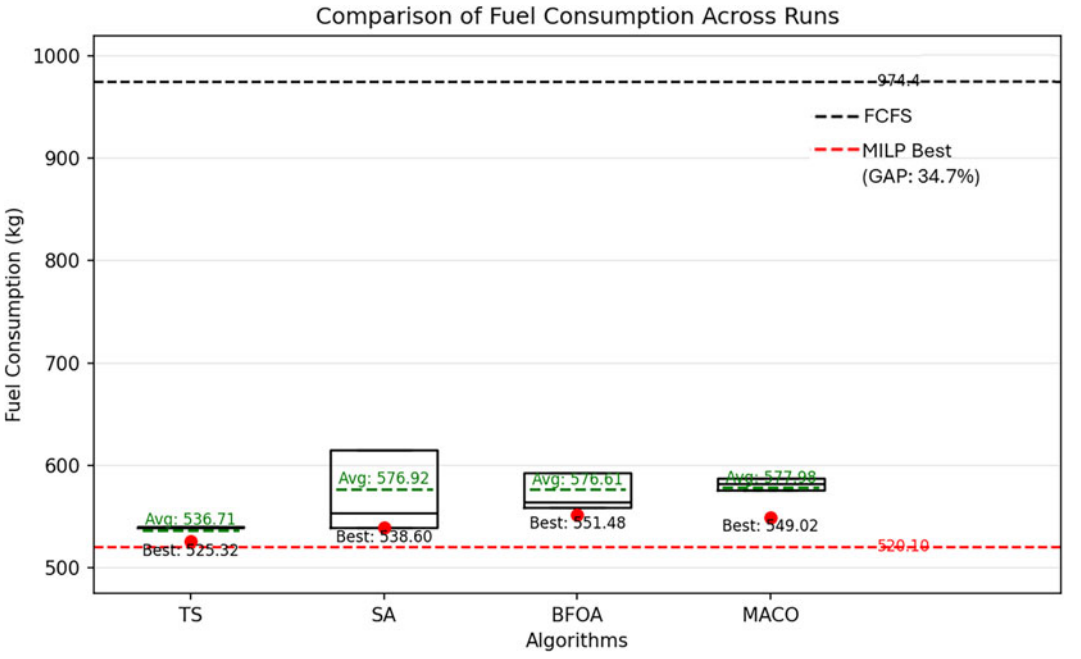


Figure 6. *Comparison of hold-fuel consumption across runs for the 50–aircraft instance. For fair benchmarking, a uniform 1,200 s wall-clock time limit was imposed on both the individual metaheuristic runs and the standalone MILP solver.*

the intermediate reordering logic (which aircraft is moved where, and why) remains hidden inside the branch-and-bound tree. Human controllers, however, must be able to relate a recommended sequence to tangible operational actions such as swapping two departures or slightly reordering flows from different aprons.

The proposed matheuristic framework directly addresses this gap. Each metaheuristic step is an explicitly logged, constrained neighbourhood move (swap or insert) applied to the FCFS queue, and every candidate sequence is then evaluated by the restricted MILP. This design has two important consequences. First, *every* cost value shown to the user is numerically precise, because it comes from solving the full MILP with the runway order fixed. Second, the search trajectory itself becomes a sequence of interpretable edits to the departure list that can be traced back to aircraft attributes such as wake-turbulence category, apron of origin and ETA. In other words, the solution process is no longer an opaque branch-and-bound trace, but a chain of small, human-readable queue adjustments whose marginal impact on fuel and delay is exactly quantified.

To illustrate this property, we examine a representative run of the TS algorithm in the 50-aircraft scenario. Starting from the FCFS baseline (974.44 kg), TS performs a series of CPS-compliant moves that gradually reduce the objective. Table 2 reports the first nine accepted moves. For each step we log the total hold fuel after the move, the type of neighbourhood operation, the queue indices involved and the marginal saving Δ relative to the previous best. Although each move directly touches only two positions (for swaps) or one block (for inserts), its effect propagates to three or four neighbouring aircraft through wake-separation and taxi-out interactions. Figure 7 visualises these savings.

By cross-referencing the queue indices in Table 2 with the scenario data (ETA, wake-turbulence type, apron), we can interpret the dominant sources of savings. Aircraft types are encoded as Type 1 (Heavy), Type 2 (Medium) and Type 3 (Light), and the wake-turbulence separation matrix SEP (i, j) enforces a 120 s separation whenever a heavier aircraft is followed by a lighter one (pairs $(1, 2)$, $(1, 3)$ and $(2, 3)$), and 60 s otherwise.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

Table 2. *Operational traceability log for a TS run in the 50-aircraft scenario (first nine accepted moves)*

Step	Total fuel (kg)	Action type	Queue indices	Marginal saving (Δ kg)
0	974.44	*FCFS baseline*	–	–
1	840.15	Insert	(32, 33)	−134.29
2	736.25	Swap	(36, 37)	−103.90
3	710.57	Swap	(19, 20)	−25.68
4	689.60	Swap	(6, 8)	−20.97
5	651.42	Insert	(17, 20)	−38.18
6	631.04	Swap	(1, 3)	−20.38
7	559.62	Swap	(41, 42)	−71.42
8	545.20	Swap	(15, 17)	−14.42
9	545.18	Swap	(15, 16)	−0.02

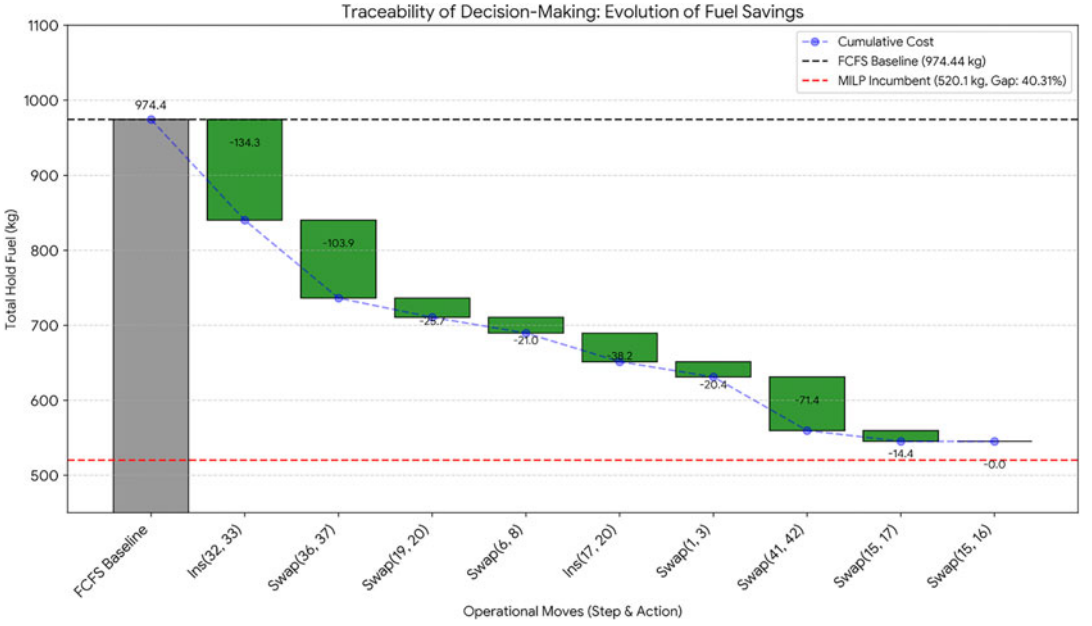


Figure 7. *Waterfall chart of the TS run corresponding to Table 2. Starting from the FCFS baseline (*974.44 *kg), each bar represents an accepted neighbourhood move, with its height equal to the marginal saving achieved by the restricted MILP evaluator. The sequence of interpretable swaps and inserts gradually closes the gap towards the best MILP incumbent (*520.10 *kg), providing a transparent, step-by-step explanation of how fuel savings emerge from specific operational adjustments to the departure queue.*

Wake-turbulence driven moves. The two largest improvements (Steps 2 and 7) align closely with this separation logic. In Step 2 the algorithm swaps the aircraft at queue indices 36 and 37: a Type 1 (Heavy) with ETA 4,700 s from Apron 2, and a Type 2 (Medium) with ETA 4,755 s from the same apron. In the FCFS order the local pattern is *Medium* → *Medium* → *Heavy* → *Medium* → *Heavy*, which contains a penalising Heavy–Medium pair requiring 120 s separation. After the swap the pattern becomes *Medium* → *Medium* → *Medium* → *Heavy* → *Heavy*, replacing the Heavy–Medium pair by Medium–Heavy and Heavy–Heavy, both of which require only 60 s. Although the move explicitly exchanges only two neighbours, it simultaneously shortens the required separation over a block of three consecutive pairs, and the MILP evaluator reports a global saving of 103.9 kg of hold fuel.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

A similar mechanism appears in Step 7, which swaps the aircraft at indices 41 and 42: a Type 3 (Light) from Apron 1 with ETA 4,963 s and a Type 1 (Heavy) from Apron 2 with ETA 5,101 s. Around these positions, the FCFS queue forms a *Heavy (index 40)* → *Light (41)* → *Heavy (42)* → *Light (43)* pattern. This configuration pays the maximum 120 s separation twice (Heavy–Light at 40–41 and Heavy–Light at 42–43). After the swap we obtain *Heavy* → *Heavy* → *Light* → *Light*, in which only one Heavy–Light pair remains and the other transitions use the cheaper 60 s standard. Again, this two-aircraft swap reconfigures the wake interaction of four aircraft at once, and the MILP quantifies the resulting reduction as 71.42 kg.

Apron and ETA-driven moves. Other moves are mainly driven by taxi-out interactions and merge ordering between aprons, rather than by wake alone. In Step 1, for instance, an insert move involving queue positions 32 and 33 reorders a Medium aircraft from Apron 3 (ETA 4,554 s) and a Light aircraft from Apron 1 (ETA 4,582 s) deep in the queue. This pair sits near a merge of different apron flows; by slightly re-timing which aircraft enters the main taxiway first, the algorithm removes a local gridlock that was causing several aircraft behind them to idle. The resulting saving of 134.29 kg cannot be explained by the wake matrix alone, but it is fully captured by the MILP, which models both separation and taxi network congestion. A similar pattern occurs in Step 6, where swapping positions 1 and 3 reshapes the head of the queue by reordering Heavy and Light departures coming from different aprons. The individual aircraft move by only two positions, consistent with the CPS limits, yet the MILP reports a meaningful 20.38 kg saving due to smoother initial merging and reduced stop-and-go taxiing.

Incremental refinements. The remaining steps (3, 4, 5, 8 and 9) contribute smaller but still interpretable improvements (typically 10–40 kg each). These moves fine-tune local precedence relations and merge orders while respecting CPS bounds, often turning near-ties in ETA into sequences that better exploit natural time gaps or cluster similar wake types together. Importantly, even these incremental changes are not heuristic guesses: for each candidate sequence the restricted MILP recomputes the full set of departure times, holds and separation-induced delays, ensuring that the reported marginal gains are exact and operationally feasible.

Overall, this analysis shows how the proposed matheuristic turns a mathematically exact but opaque optimisation process into a sequence of small, understandable decisions ('swap these two departures', 'insert this aircraft slightly later') whose effects on fuel and delay can be inspected and discussed with controllers. This level of traceability is essential for building trust in automated Decision Support System (DSS) and for enabling human-in-the-loop use, where operators can approve, reject or adapt individual moves while retaining the numerical guarantees of an exact MILP evaluator.

7.0 Discussion and conclusion

This study tackled the DSP by proposing a matheuristic framework that combines four enhanced metaheuristics (TS, SA, MACO, BFOA) with a *restricted* MILP model. Instead of letting the MILP search over all permutations, the metaheuristics explore the discrete space of CPS-compliant departure orders using constraint-aware neighbourhoods (swaps and inserts), while the MILP is used as an *evaluator* that optimises continuous timings and holds for each fixed sequence. This MILP-as-oracle design preserves the exact treatment of wake-turbulence separation and complex taxi interactions, but at a computational cost that remains manageable for pre-tactical planning horizons.

The 40-aircraft experiments show that the proposed matheuristic architecture can fully exploit this evaluator. All four algorithms reached the MILP optimum or remained within about 1% in every run, delivering an average reduction of roughly 29% in ground-hold fuel compared with the FCFS baseline. Enhanced SA and enhanced BFOA were particularly consistent, repeatedly attaining the optimal solution despite operating in a discrete, highly constrained search space. These findings challenge the perception that population-based or stochastic methods are ill-suited for detailed sequencing tasks; when supplied with an exact MILP evaluator and CPS-compliant moves, they become reliable high-quality optimisers rather than coarse heuristics.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

The 50-aircraft case highlights a different aspect: the limits of a monolithic exact approach. The standalone MILP solver produced a strong best incumbent of 520.10 kg but was unable to close the optimality gap (40.31%) within the 600 s time limit. Under the same wall-clock budget, the matheuristics – especially MACO and TS – consistently generated solutions within about 5% of this incumbent. In other words, once the instance becomes large enough that even the exact solver can no longer certify optimality, the hybrid methods provide competitive fuel performance while retaining the operational structure imposed by CPS and wake constraints.

The time-limit sensitivity analysis (600 s vs. 1,200 s total evaluator budget; see Fig. 6) further clarifies the trade-off. Doubling the CPU time shifts the entire distribution of objective values closer to the MILP incumbent and tightens the spread across runs. TS benefits most strongly, with both its mean and best-of-run hold-fuel consumption approaching the MILP benchmark, while SA, BFOA and MACO exhibit smaller but still visible gains. From an operational perspective this suggests that a 600 s budget already captures a large fraction of the attainable improvement, and that longer runs are mainly justified in pre-tactical planning contexts where additional computation is available.

Perhaps the most distinctive feature of the proposed framework is its *operational explainability*. Because each metaheuristic move is a logged CPS-compliant swap or insertion and each candidate is scored by the same restricted MILP, the solution process becomes a traceable sequence of small, interpretable edits to the departure queue rather than an opaque branch-and-bound path. The traceability analysis in Section 6.2 showed how a handful of local swaps (e.g. Heavy–Medium and Heavy–Light patterns) can be directly linked to specific reductions in wake-induced holds and taxi-gridlock, with marginal savings precisely quantified in kilograms of fuel. This level of transparency is crucial for human-in-the-loop adoption: controllers can understand *why* a given sequence is recommended and can discuss, accept or adapt individual moves without losing the guarantees provided by the MILP evaluator.

7.1 Application of model results in real operations

Beyond global performance indicators, the restricted MILP produces detailed aircraft-level outputs (Table 3), including start and take-off times (BEGINX, TOUCHT), queueing delays (QDELAY), final sequence position (POS), location-specific holds (hold0/1/2) and associated emissions (HC, CO, NOx). These fields are directly compatible with the information needs of pre-tactical tools such as a DMAN operating within the airport collaborative decision making (A-CDM) framework. Given a set of off-block intentions, the model can be used to generate target take-off times (TTOT) and target start-up approval times (TSAT) that are both fuel-efficient and CPS-compliant, and that can be refreshed in pre-tactical time frame on the order of tens of minutes as conditions change [59].

Implications for air traffic controllers (ATCos). For controllers, the main value of the framework lies in its ability to translate a complex optimisation into a small number of interpretable and schedulable moves.

- *Where and when to delay aircraft:* Explicit hold durations at named locations (e.g. hold1, hold2) allow ATCos to see *where* delay is absorbed and to verify that separation standards are met without overusing runway-adjacent holding points.
- *Position shifting within CPS limits:* The POS and CPS structure make it clear which flights have been advanced or deferred relative to FCFS and by how many positions, supporting fairness and equity discussions with airlines.
- *Final sequence validation:* Because every candidate is evaluated by a full MILP, controllers know that the proposed ordering is globally feasible with respect to separation and taxi network interactions, rather than being the result of a local repair heuristic.

These properties enable the design of DSS interfaces where ATCos can inspect the proposed queue, highlight individual swaps or inserts and immediately see the effect on fuel, delay and emissions, instead

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

Table 3. *Sample aircraft-level outputs from the restricted-MILP evaluator for a single optimised sequence*

Aircraft	BEGINX (s)	TOUCHT (s)	HOLDFUEL (kg)	QDELAY (s)	POS	inttime1 (s)	inttime2 (s)	hold1 (s)	hold2 (s)	hold0 (s)	HC_EI (kg)	CO_EI (kg)	NOx_EI(kg)
1	3,440	3,933	0.00	0	26	3,744	3,789	0	0	0	0.00	0.00	0.00
2	1,625	2,118	0.00	0	5	1,929	1,974	0	0	0	0.00	0.00	0.00
3	3,299	4,053	6.76	261	27	3,603	3,648	0	0	0	270.40	655.71	17.78
4	1,908	2,427	5.90	26	7	2,212	2,257	0	0	0	5.43	88.79	28.14
5	2,001	2,607	11.74	86	9	2,332	2,377	27	0	0	51.67	517.88	37.58
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
50	2,925	3,627	21.72	198	22	3,229	3,285	0	11	0	95.55	957.68	69.49

of being asked to trust a purely numerical 'optimal' solution. The conflict-visualisation work in Ref. [60] demonstrates how similar outputs from the same Antalya model can be mapped onto taxiway layouts and runway timelines, further reinforcing situational awareness.

Implications for airport and airline operators. For airport and airline stakeholders, the aircraft-level outputs support both tactical decisions and longer-term performance monitoring.

- *Hold-fuel management and costs:* HOLDFUEL values quantify the additional fuel burned due to ground holding and queueing, enabling airlines to estimate cost impacts and airports to evaluate the benefit of improved sequencing policies.
- *Emission monitoring:* HC, CO and NOx emission estimates along the taxi-out phase provide a basis for environmental reporting, sustainability key performance indicators (KPIs) and scenario comparisons (e.g. evaluating alternative cost-weightings between fuel and delay).
- *Delay and bottleneck analysis:* QDELAY and POS information can be aggregated over time to identify recurrent choke points in the ground operation and to inform infrastructure or procedural investments.

Overall, the proposed matheuristic approach provides numerically precise, operationally explainable schedules that can feed directly into decision-support tools for ATCos, airport operators and airlines. It thereby bridges the long-standing gap between exact but opaque optimisation and fast but oversimplified heuristic rules.

8.0 Future work

The present work focuses on departure scheduling for a single runway and a one-hour traffic sample calibrated for Antalya Airport. Several extensions are worth pursuing to increase realism and operational impact.

Antalya typically operates its parallel runways in a segregated fashion during busy periods, with departures and arrivals predominantly assigned to different runways; hence, arrivals do not interleave with the departure stream optimised in this study (see Fig. 31 in Ref. [61]). When mixed-mode operations are required in future work, the existing formulation can be extended by treating arrivals as exogenous runway-blocking windows: for each landing ℓ with touchdown time τ_ℓ, we reserve an interval $\left[\tau_\ell - \delta^{\text{in}}, \tau_\ell + \delta^{\text{out}}\right]$ and constrain departure touchdown times to lie outside these windows via standard disjunctive (big-M) constraints. This preserves CPS and wake-separation rules while explicitly accounting for arrival interference. A natural next step is to validate such a mixed-mode extension on busier hubs where arrival–departure interleaving is unavoidable.

A key limitation of the current study is the lack of access to proprietary surface-movement data for Antalya Airport, which prevents a direct comparison between optimised sequences and realised departure streams, taxi-out profiles and fuel consumption. Securing access to A-CDM logs, advanced surface movement guidance and control system (A-SMGCS) trajectories or automatic dependent surveillance–broadcast (ADS-B) data would enable (i) statistical calibration of taxi-time and fuel models, (ii) retrospective replay of historical traffic under alternative sequencing policies and (iii) out-of-sample validation of the matheuristic recommendations against real operational outcomes.

In this paper, all metaheuristic variants are evaluated using the same restricted MILP; 'pure' TS/SA/MACO/BFOA configurations without the MILP have not been reported. This is intentional: our comparison principle is that every candidate order should be scored by the *same* feasibility and timing engine so that all methods are assessed under an identical, operationally valid objective. Removing the MILP would require designing an alternative evaluator – typically a deterministic forward-propagation model plus conflict-repair rules – which would change the scoring function and undermine strict comparability with FCFS and the MILP-backed matheuristics. A fair MILP-free benchmark would need (i) a precisely specified propagation model, (ii) explicit repair policies for separation violations, (iii) identical

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

wall-clock budgets and (iv) additional metrics such as feasibility rate and violation duration. Developing such baselines, and combining the current restricted-MILP approach with other metaheuristics (e.g. genetic algorithms or large neighbourhood search), is a promising avenue for future research.

While we reported a first time-limit sensitivity experiment, a more systematic analysis of model parameters remains to be done. Future work should examine the effect of CPS bounds, wake-separation standards, emission cost weights and uncertainty in ETD/taxi-time predictions on both solution quality and algorithm behaviour. Scenario-based or stochastic variants of the restricted MILP could be used to hedge against variability in demand or surface conditions. Similarly, multi-objective formulations that trade off fuel, delay, fairness (CPS) and emissions could be explored, with the metaheuristics operating on weighted sums or Pareto-based dominance criteria.

Finally, the current framework optimises static one-hour snapshots. In real operations, departure streams evolve as new flight information arrives, aircraft miss slots or runway configuration changes. Embedding the restricted-MILP matheuristics in a rolling-horizon controller that periodically re-optimises the sequence while preserving stability (e.g. through additional position-locking constraints) would bring the method closer to a deployable DMAN module. Coupling this with the explainable move-by-move outputs developed in this paper would enable human-in-the-loop controllers to monitor and selectively apply recommended adjustments in real time.

In summary, the proposed matheuristic framework offers a balanced compromise between exactness, computational tractability and operational explainability for departure sequencing. Building on these foundations through richer data, extended models and online control architectures has the potential to deliver more efficient, sustainable and resilient airport operations.

Acknowledgment. This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Declaration of generative AI and AI-assisted technologies in the writing process. During the preparation of this work the author(s) used ChatGPT in order to improve language and readability, with caution. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] International Air Transport Association (IATA). Global outlook for air transport. Online report. [Online]. https://www.iata.org/en/iata-repository/publications/economic-reports/global-outlook-for-air-transport—december-2023—report/

[2] Yousefzadeh Aghdam, M., Kamel Tabbakh, S.R., Mahdavi Chabok, S.J. and Kheyrabadi, M. Optimization of air traffic management efficiency based on deep learning enriched by the long short-term memory (LSTM) and extreme learning machine (ELM), *J. Big Data*, 2021, **8**, (54), pp 1–18.

[3] Hong, Y. Deterministic and stochastic optimization for aircraft arrival sequencing and scheduling under uncertainty, 2018, PhD dissertation.

[4] Balakrishnan, H. and Chandran, B. Efficient and equitable departure scheduling in real-time: New approaches to old problems, *Proceedings of the 7th USA/Europe Air Traffic Management Research and Development Seminar*, Barcelona, Spain, 2007.

[5] Federal Aviation Administration (FAA). (2014) Order JO 7110.65S: Air traffic control. [Online]. https://www.faa.gov/documentlibrary/media/order/jo_7110.65v.pdf (Accessed 12 September 2024).

[6] Balakrishnan, H. and Chandran, B.G. Algorithms for scheduling runway operations under constrained position shifting, *Oper. Res.*, 2010, **58**, (6), pp 1650–1665.

[7] Balakrishnan, H. and Chandran, B.G. Scheduling aircraft landings under constrained position shifting, *AIAA Guidance, Navigation, and Control Conference and Exhibit,* AIAA, 2006, pp 2006–6320.

[8] Simaiakis, I. and Balakrishnan, H. A queuing model of the airport departure process, *Transp. Sci.*, 2016, **50**, (1), pp 94–109.

[9] Cecen, R.K. and Çetek, C. Optimising aircraft arrivals in terminal airspace by mixed integer linear programming model, *Aeronaut. J.*, 2020, **124**, (1278), pp 1129–1145.

[10] Dönmez, K. Evaluation of the trade-off between ground delays and intersecting departures under various pilot acceptance rate scenarios, *Transp. Res. Rec.*, 2023, **2677**, (7), pp 733–746.

[11] Solveling, G., Solak, S., Clarke, J.-P. and Johnson, E. Runway operations optimization in the presence of uncertainties, *J. Guid. Control Dyn.*, 2011, **34**, (5), pp 1373–1382.

[12] Ikli, S., Mancel, C., Mongeau, M., Olive, X. and Rachelson, E. The aircraft runway scheduling problem: A survey, *Comput. Oper. Res.*, 2021, **132**, p 105336.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

[13] Sun, B., Wei, M. and Jing, B. Optimal model for the aircraft arrival and departure scheduling problem with fuzzy runway incursion time, *Math. Biosci. Eng.*, 2021, **18**, (5), pp 6724–6738.
[14] Goerigk, M. and Hartisch, M. A framework for inherently interpretable optimization models, *Eur. J. Oper. Res.*, 2023, **310**, (3), pp 1312–1324.
[15] Aigner, K.-M., Goerigk, M., Hartisch, M., Liers, F. and Miehlich, A. A framework for data-driven explainability in mathematical optimization, 2023. [Online]. https://arxiv.org/abs/2308.08309
[16] Lumbreras, S. and Ciller, P. Interpretable optimization: Why and how we should explain optimization models, *Appl. Sci.*, 2025, **15**, (10), p 5732.
[17] Gotteland, J.-B., Durand, N., Page, E. and Alliot, J.-M. Aircraft ground traffic optimization, 4th USA/Europe Air Traffic Management Research and Development Seminar, 2001.
[18] Wang, X., Brownlee, A.E.I., Weiszer, M., Woodward, J.R., Mahfouf, M. and Chen, J. A chance-constrained programming model for airport ground movement optimisation with taxi time uncertainties, *Transp. Res. Part C: Emerg. Technol.*, 2021, **132**, p 103382.
[19] García, J., Berlanga, A., Molina, J.M. and Casar, J.R. Optimization of airport ground operations integrating genetic and dynamic flow management algorithms, *AI Commun.*, 2005, **18**, (2), pp 143–164.
[20] Almeida, J., Soares, J., Lezama, F., Limmer, S., Rodemann, T. and Vale, Z. A systematic review of explainability in computational intelligence for optimization, *Comput. Sci. Rev.*, 2025, **57**, p 100764.
[21] Fyvie, M., McCall, J.A.W., Christie, L.A., Brownlee, A.E.I. and Singh, M. Towards explainable metaheuristics: Feature extraction from trajectory mining, *Expert Syst.*, 2025, **42**, (1), p e13494.
[22] Westin, C., Borst, C. and Hilburn, B. Automation transparency and personalized decision support: Air traffic controller interaction with a resolution advisory system, *IFAC-PapersOnLine*, 2016, **49**, (19), pp 201–206. 13th IFAC Symposium on Analysis, Design, and Evaluation of Human-Machine Systems HMS 2016.
[23] Boschetti, M.A. and Maniezzo, V. Matheuristics: Using mathematics for heuristic design, *4OR*, 2022, **20**, (2), pp 173–208.
[24] Archetti, C. and Speranza, M.G. A survey on matheuristics for routing problems, *EURO J. Comput. Optim.*, 2014, **2**, (4), pp 223–246.
[25] Solyal, O. and Süral, Ö. An effective matheuristic for the multivehicle inventory routing problem, *Transp. Sci.*, 2022, **56**, (4), pp 1044–1057.
[26] Boschetti, M.A. and Maniezzo, V. A set covering based matheuristic for a real-world city logistics problem, *Int. Trans. Oper. Res.*, 2015, **22**, (1), pp 169–195.
[27] Vadseth, S.T., Andersson, H. and Stålhane, M. An iterative matheuristic for the inventory routing problem, *Comput. Oper. Res.*, 2021, **131**, p 105262.
[28] Moura, A., Pinto, T., Alves, C. and de Carvalho, J.V. A matheuristic approach to the integration of three-dimensional Bin packing problem and vehicle routing problem with simultaneous delivery and pickup, *Mathematics*, 2023, **11**, (3), p 713.
[29] Chouman, M. and Crainic, T.G. A cutting-plane matheuristic for service network design with design-balanced requirements, *Transp. Sci.*, 2014, **49**, (1), pp 99–113.
[30] Villegas, J.G., Prins, C., Prodhon, C., Saldanha-da Gama, F.R.W. and Santos-Pereira, N. A matheuristic for the truck and trailer routing problem, *Eur. J. Oper. Res.*, 2013, **230**, (2), pp 231–244.
[31] Angelelli, E., Archetti, C. and Peirano, L. A matheuristic for the air transportation freight forwarder service problem, *Comput. Oper. Res.*, 2020, **123**, p 105002.
[32] Dang, Q.-V., van Diessen, T., Martagan, T. and Adan, I. A matheuristic for parallel machine scheduling with tool replacements, *Eur. J. Oper. Res.*, 2021, **291**, (2), pp 640–660.
[33] Fanjul-Peyro, L., Perea, F. and Ruiz, R. Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources, *Eur. J. Oper. Res.*, 2017, **260**, (2), pp 482–493.
[34] Della Croce, F. and Salassa, F. A variable neighborhood search based matheuristic for nurse rostering problems, *Ann. Oper. Res.*, 2014, **218**, (1), pp 185–199.
[35] Karlsson, E., Rönnberg, E., Stenberg, A. and Uppman, H. A matheuristic approach to large-scale avionic scheduling, *Ann. Oper. Res.*, 2021, **302**, (2), pp 425–459.
[36] da Silva, L.C., Ueno, F.Y., dos Santos, M.O. and de Carvalho, A.C.P.L.F. A new modeling for the aircraft landing problem and matheuristic approach to solve the problem with a large number of aircraft, *Pesqui. Operacional*, 2023, **43**, pp 1–30.
[37] Zhang, Q., Chan, F.T.S., Chung, S.-H. and Fu, X. A matheuristic for aircraft maintenance routing problem incorporating cruise speed control, *Expert Syst. Appl.*, 2024, **242**, p 122711.
[38] Karsu, Ö. and Solyal, O. A new formulation and an effective matheuristic for the airport gate assignment problem, *Comput. Oper. Res.*, 2023, **151**, p 106073.
[39] Delgado, F. and Mora, J. A matheuristic approach to the air-cargo recovery problem under demand disruption, *J. Air Transport Manag.*, 2021, **90**, p 101939.
[40] Kyriakidou, A. A matheuristic approach for solving the aircraft landing scheduling problem, PhD dissertation, University of Macedonia, 2020.
[41] Atkin, J.A.D., Burke, E.K., Greenwood, J.S. and Reeson, D. A metaheuristic approach to aircraft departure scheduling at London Heathrow airport, in *Computer-aided Systems in Public Transport*, Lecture Notes in Economics and Mathematical Systems, vol. 600, Springer, 2008, Berlin, Heidelberg, pp 235–252.
[42] Capr, S. and Ignaccolo, M. Genetic algorithms for solving the aircraft-sequencing problem: The introduction of departures into the dynamic model, *J. Air Transp. Manag.*, 2004, **10**, (5), pp 345–351.
[43] Benlic, U., Brownlee, A.E.I. and Burke, E.K. Heuristic search for the coupled runway sequencing and taxiway routing problem, *Transp. Res. Part C: Emerg. Technol.*, 2016, **71**, pp 333–355.

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press

[44] Riahi, V., Newton, M.A.H., Polash, M.M.A., Su, K. and Sattar, A. Constraint guided search for aircraft sequencing, *Expert Syst. Appl.*, 2019, **118**, pp 440–458.
[45] Hurter, C., Degas, A., Guibert, A., Durand, N., Ferreira, A., Cavagnetto, N., Islam, M.R., Barua, S., Ahmed, M.U., Begum, S., Bonelli, S., Cartocci, G., Di Flumeri, G., Borghini, G., Babiloni, F. and Aricó, P. Usage of more transparent and explainable conflict resolution algorithm: Air traffic controller feedback, *Transp. Res. Procedia*, 2022, **66**, pp 270–278. 34th Conference of the European Association for Aviation Psychology.
[46] Glover, F. Tabu search—Part I, *ORSA J. Comput.*, 1989, **1**, (3), pp 190–206.
[47] Glover, F. Tabu search—Part Ii, *ORSA J. Comput.*, 1990, **2**, (1), pp 4–32.
[48] Cecen, R.K. and Çetek, C. A two-step approach for airborne delay minimization using pretactical conflict resolution in free-route airspace, *J. Adv. Transp.*, 2019, Article ID 4 805 613, **2019**, 17 pp.
[49] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. Optimization by simulated annealing, *Science*, 1983, **220**, (4598), pp 671–680.
[50] Dorigo, M., Di Caro, G. and Gambardella, L.M. Ant algorithms for discrete optimization, *Artif. Life*, 1999, **5**, (2), pp 137–172.
[51] Ting, C.-J. and Chen, C.-H. A multiple ant colony optimization algorithm for the capacitated location routing problem, *Int. J. Prod. Econ.*, 2013, **141**, (1), pp 34–44.
[52] Passino, K.M. Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Syst. Mag.*, 2002, **22**, (3), pp 52–67.
[53] Niu, B., Fan, Y., Tan, L., Rao, J. and Li, L. A review of bacterial foraging optimization Part I: Background and development, in Huang, D.-S., McGinnity, T., Heutte, L. and Zhang, X.-P. (Eds.), Advanced Intelligent Computing Theories and Applications. ICIC 2010, Communications in Computer and Information Science, vol. 93, Springer, 2010, Berlin, Heidelberg, pp 743–753.
[54] ICAO Aircraft Engine Emissions Databank. Online database, hosted by the European Union Aviation Safety Agency (EASA), International Civil Aviation Organization (ICAO), 2024, https://www.easa.europa.eu/en/domains/environment/icao-aircraft-engine-emissions-databank (Accessed 15 November 2025).
[55] Flightradar24. (2023) Flight radar 24. [Online]. https://www.flightradar24.com/
[56] GDSAA Turkey. (2024) Antalya airport. [Online]. https://www.dhmi.gov.tr/Sayfalar/Havalimani/Antalya/GenelBilgiler.aspx
[57] AIP Türkiye. (2023) Aeronautical information publication. [Online]. https://www.dhmi.gov.tr/Sayfalar/aipturkey.aspx
[58] International Civil Aviation Organization (ICAO). Icao aircraft engine emissions databank, 2023. [Online]. https://www.easa.europa.eu/en/domains/environment/icao-aircraft-engine-emissions-databank (Accessed 12 September 2024).
[59] Royal Schiphol Group and LVNL (Air Traffic Control the Netherlands). A-CDM Manual, Schiphol Airport, The Netherlands, Amsterdam.
[60] Dönmez, K. Airport ground optimizer (ago): A decision support system initiative for air traffic controllers with optimization and decision-aid algorithms, *J. Air Transp. Manag.*, 2024, **119**, p 102648.
[61] Dönmez, K. Efficiency-based airport ground conflict detection and resolution: Integrating stand hold strategy with constrained position shifting, *Int. J. Aeronaut. Space Sci.*, 2025, **26**, pp 1912–1939.

Cite this article: Dönmez K. and Sahin O. (2026). Matheuristic strategy development and benchmarking for explainable and efficient aircraft departure sequencing. *The Aeronautical Journal*, **130**, 2010–2035. https://doi.org/10.1017/aer.2026.10146

https://doi.org/10.1017/aer.2026.10146 Published online by Cambridge University Press