CEAS Aeronautical Journal
https://doi.org/10.1007/s13272-023-00708-4

ORIGINAL PAPER

Constraint reconciliation optimization to minimize primary and reactionary delay in a research network strategic tool

Jan Berling[1] · **Alexander Lau**[2] · **Volker Gollnick**[1]

Received: 21 December 2022 / Revised: 27 October 2023 / Accepted: 18 December 2023

Abstract

When regulated network capacities are violated, flights may be delayed to meet time-based slot restrictions. In addition, reactionary delays occur when time buffers to subsequent flight legs are exceeded. This work aims to minimize both—primary and reactionary delays in a dynamic simulation of the European Air Transportation System. The slot allocation process is solved by minimizing network delay with a binary optimization approach instead of using the current first-planned–first-served principle. The new module presented in this study, called constraint reconciliation and optimization (CRO), is applied within EUROCONTROL's Research Network Strategic Tool (R-NEST). The results are compared to delays generated by R-NEST's computer-aided slot allocation (ISA–CASA). In simulations of tactical air traffic flow management (ATFM) operations, time is iterated over the day and flight plans are updated with random and propagated delays. The computational complexity of all possible delay- and slot-entry-permutations is reduced by the method of column generation for solving large linear problems. Three validation scenarios that contain flight plans, rotation margins, sector configurations, regulations, and deterministic or stochastic delays are evaluated. In the most realistic scenario, compared to ISA–CASA, CRO reduces primary and reactionary delays by up to 16% while achieving low levels of airspace sector and airport overloads and fast computation times.

Keywords Demand Capacity Balancing · Reactionary Delay · Optimization · Air Traffic Flow Management · European Air Transportation System · Fairness

Abbreviations

AIRAC	Aeronautical information regulation and control
APT	Airport
ATC	Air traffic control
ATFCM	Air traffic flow and capacity management
ATFM	Air traffic flow management
ATM	Air traffic management
ATS	Air transportation system
AU	Airspace user
BIP	Binary integer problem
EATMN	European ATM network
EOBT	Estimated off-block time
ETO	Estimated time over
EUROCONTROL	European Organisation for the Safety of Air Navigation
CODA	Central office for delay analysis
CRO	Constraint reconciliation and optimization
CTO	Calculated time over
CTOT	Calculated take-off time
DCB	Demand–capacity–balancing
DDR2	Demand data repository
FMP	Flow management position
FPFS	First-planned–first-served
CASA	Computer-aided slot allocation
MPR	Most penalizing regulation
NM	Network management
R-NEST	Research network strategic tool
SESAR	Single European Sky Air Traffic Management Research
SCIP	Solving constraint integer programs
SIT1	Slot issue time

✉ Jan Berling
jan.berling@tuhh.de

[1] Institute of Air Transportation Systems, Hamburg University of Technology (TUHH), Blohmstr. 20, 21079 Hamburg, Germany

[2] Air Transportation Systems, German Aerospace Center (DLR), Blohmstr. 20, 21079 Hamburg, Germany

Published online: 25 April 2024

1 Introduction

Traffic in the European Air Transportation System (ATS) frequently threatens to exceed declared sector and airport capacities. In the (pre-) tactical air traffic flow and capacity management (ATFCM) process, demand–capacity imbalances are resolved by allocating calculated take-off times (CTOT). These are selected according to demand–capacity–balancing (DCB) constraints along the planned flight route of an AFTCM-restricted flight. Given the potentially large number of imbalances and their spatial distribution within the European Air Traffic Management (ATM) network, numerous ways of CTOT assignments to adhere to given capacity constraints exist. In Europe, tactical DCB delay is allocated by the computer-aided slot allocation (CASA) heuristic based on the first-planned–first-served (FPFS) principle, which aims to meet achievable flow rates and fairness objectives accepted by airlines. However, CASA generates solutions without considering reactionary subsequent delay, which occurs when a flight arrives late for subsequent flight legs.

Smaller ATFCM scenarios have been modeled by Odoni [1] and solved with LP optimization approaches by Lulli and Odoni [2]. To allow operational resolution throughout a day, computation times are reduced by the method of Column Generation to reduce LP optimization problems of several thousand decision variables, as applied to ATFM by Kaufhold et al. [3]. Bertsimas and Stock [4] resolved large-scale scenarios containing several thousand flights and several hundred network entities, while other continental scenarios are resolved with Column Generation by, e.g., Balakrishnan and Chandran [5] and by Lau et al. [6] with rolling time horizons to generate time-dependent sub-problems. Optimization and FPFS have been studied by Barnhart et al. [7] and Ruiz et al. [8]. Propagated delay has been studied by Ivanov et al. [9], Montlaur and Delgado [10], and Campanelli et al. [11].

The presented work was performed with and on behalf of DLR Institute of Air Transport within the project SESAR PJ09. We have developed the constraint reconciliation and optimization (CRO) as a tool for minimizing primary and reactionary subsequent delay in EUROCONTROL's Research Network Strategic Tool (R-NEST) [12]. In this paper, the performance evaluation of CRO in comparison to the built-in CASA emulation called ISA–CASA [13] is presented as well as the principal setup of the CRO tool. Results for different weighting of primary and reactionary delays are obtained by varying a cost coefficient.

R-NEST is a model-based simulation tool for European ATFCM including a most-realistic representation of trajectories, regulations, rotation margins, and random delays (see Fig. 1). During a dynamic simulation of the European ATS, reactionary and random departure delays push flights out of their estimated departure slots. Consequently, air traffic must be reconciled iteratively over the course of the day. Airport and en-route (airspace) regulations are specified by flow rates that must be respected within regulation periods. These flow rates represent a coordinated number of flights that can be safely handled in a given time period. To implement this, a slot allocation list is created for each regulation, consisting of time-equidistant slots, matching rates, and durations [13]. To satisfy the flow rates in all time-intervals, flights can only enter assigned so called traffic volumes with a valid entry slot. Those

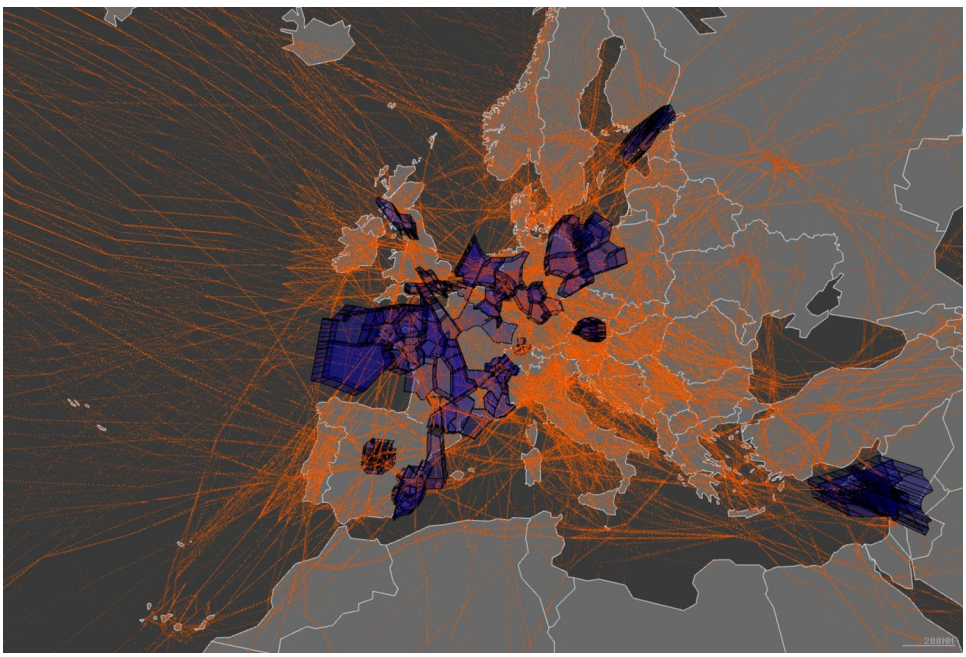

Fig. 1 R-NEST Screenshot of regulated traffic volumes (ATC capacity and staffing reason only) and trajectories, 2th June 2017

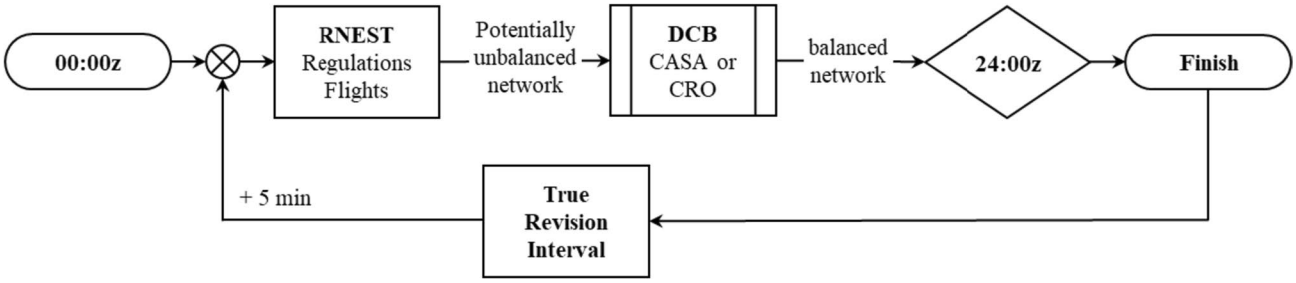

Fig. 2 True revision interval: R-NEST calls the DCB-module every 5 min to resolve imbalances

traffic volumes represent specific traffic flows, e.g., along a defined route within the dedicated airspace sector. When a flight enters multiple regulations, fixed entry times ("time over") may lead to the fact, that it is impossible to match all given slot entry times. Therefore, CASA accepts slots being allocated in time windows [14, 15]. Because each entry time has different possible slots, there are manifold permutations of slot allocations for flights in multiple regulations.

From the perspective of equity and fairness, which is an important subtopic in the context of ATFM optimization [14], it is challenging to deal with reactionary delays. As defined by Broome [16], fairness is relational and based on claims. Claims are to be satisfied proportionally, but *prima facie* they ought to be satisfied. According to Curtis, fairness requires efficiency [17]. It follows that claims require some satisfaction, suggesting that flights threatened by reactionary delays may deserve consideration. Taking reactionary delays into account benefits preceding flights, which leads to the question of whether claims of flights in the same rotation should be aggregated. However, whether claims ought to be *aggregated* is an open question in the fairness debate [18, 19]. Is it fair to delay a flight that will not cause reactionary delay instead of delaying a flight that will? Should reactionary delay be reduced to provide some satisfaction to the claims of the affected flights and to improve overall efficiency? If reactionary delays are to be penalized, how much should they be penalized relative to primary delays? These moral deliberations can only be resolved through cooperation, and the Broomean' framework is a powerful tool for structuring ethical debates [20]. We support this debate by providing insight into how the system responds to different types of cost coefficients.

Due to the different solving approaches, CRO and ISA–CASA result in different delay patterns and remaining capacity overloads. Therefore, the evaluation considers both delays and remaining capacity overloads that may result from slot tolerances. Simulation results for various continental scenarios, including ATM, reactionary, and non-ATM-delays, are compared between ISA–CASA and CRO in terms of solver performance, primary and reactionary delays, and overloads of traffic volumes.

2 Research network strategic tool (R-NEST)

The Research Network Strategic Tool (R-NEST) is a modeling and analysis tool for the European ATM Network (EATMN), which was developed by EUROCONTROL to test innovative processes in the context of network operations in a highly realistic network simulation environment. The tool runs dynamic ATFCM simulations in rolling time steps throughout the day and supports several modules that emulate different ATFCM actors. In this study, only the network management (NM) actor is enabled and its functionality is compared between the built-in ISA–CASA and the user-built CRO. In the simulation, they are called every 5 min to balance demand and capacity (see Fig. 2). Real data from aeronautical information regulation and control (AIRAC) are used and consolidated. Most relevant to this study are the capacity constraints (regulations) on traffic volumes, capacity demand by trajectories, and reactionary and non-ATM delays.

2.1 Regulations

ATFCM regulations limit the capacity of network entities, such as airports or airspace volumes, for specific periods of time to satisfy operational safety on a high level. Sectors are segmented into traffic volumes based on prevailing demand patterns. Traffic volume regulations are implemented for a variety of reasons, including air traffic control (ATC) capacity and staffing. Regulations are segmented into time bands, called entry slots, on which slot allocation lists are generated in which flight movements are assigned to the respective entry slot. For example, if a regulation has a flow rate of 18 flights per hour, there is one single slot available for one flight at every 3 min to distribute the traffic as homogeneously as possible over the regulation time duration. Typically, flow rates are monitored in 20-min integration windows with sliding steps of 10 min, so that the integration windows overlap (see Fig. 3). If all flights are assigned to allocated slots at their scheduled entry time, the flow rate is maintained. However, the regulations include window-width tolerances to allow elasticity for CASA calculations [21].

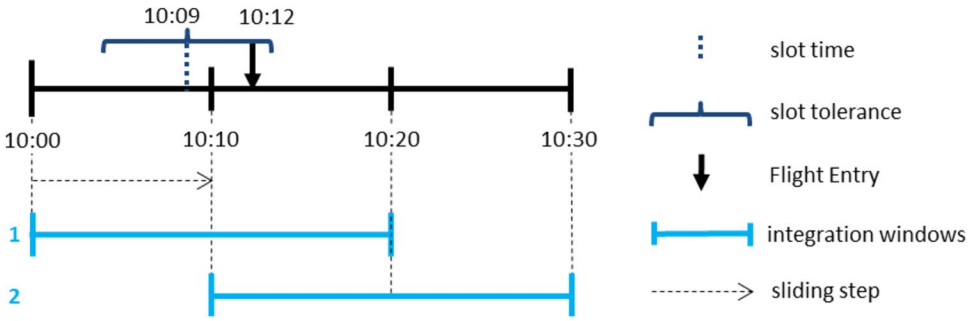

Fig. 3 Temporal visualization of a sample regulation showing the mapping of flights to the integration windows with a slot tolerance. The slot time is 10:09z, while the entry time is 10:12z, within the slot tolerance. However, the slot time is in integration window 1, while the entry time is in integration windows 1 and 2. This can lead to overloads in integration window 2

Each slot has a capacity of one flight, and there are at most as many slots as there is available capacity. If flights arrive in one window but have a slot in another window, as slot times cannot always match entry times, demand may exceed capacity, resulting in overloads. Overloads are calculated as the difference between the number of flights entering a given traffic volume and the declared flow rate for each integration window. For example, if three flights enter a sector in the 10:00–10:20 integration window and the declared capacity of arriving flights in that interval is two, there is an overload of one. If no slot is feasible, flights may be allocated to the end of the regulation, without capacity restrictions.

2.2 Flight plans

R-NEST takes the latest estimated flight plan data sets from the Demand Data Repository (DDR2) [22], which contain 4D flight profiles as being filed by aircraft operators. For the DCB modules, only entry times for the regulated traffic volumes are relevant. Exempted flights, such as government flights, cannot be delayed but still consume capacity.

2.3 Delay types

Three types of delay are applied within R-NEST: primary ATFCM delay, reactionary delay and non-ATM delay. Delay is added to departure time and subsequent entry times, while flight times and 4D-profiles remain fixed.

Primary ATFCM delay is allocated by the DCB module to resolve airport or en-route imbalances. Throughout the day, the ATM-delay can be amended until the flight is issued. While delayed flights may pass through multiple regulations, the delay is attributed to the most penalizing regulation (MPR), which is defined as the regulation that causes the most delay for a given flight [21].

Reactionary delay is the knock-on delay induced on subsequent flights using the same aircraft. If the delay is greater than the rotation margin, reactionary delay occurs for the subsequent flight leg. Rotation margins are defined as scheduled departure time—arrival time—turnaround time, where the turnaround time is 53 min based on averaged Central Office for Delay Analysis (CODA) data [23]. In R-NEST, flight rotations are available for approximately 80% of the flights for which an aircraft registration number is available. After calculating reactionary delays are calculated in one time step of the simulation, they are not applied until the next time step. CRO then allocates slots and delays according to the new estimated off-block time (EOBT), as shown in Fig. 4.

Non-ATM delay emulates stochastic departure disturbances based on historical data [23] that affect 25% of flights. The applied delay distributions with a minimum of 5 min and a maximum of 30 min, are attributed from 2 h to 5 min before EOBT. When a scenario is simulated several times, the random delays evolve equally in each simulation, which means that ISA–CASA and CRO are affected by the same random delays.

2.4 Simulation setup

R-NEST simulations cover a full day of operations within the European ATM network and process flights as well as capacity regulations, while DCB imbalances are resolved iteratively by the DCB-modules called CRO and ISA–CASA (see Fig. 2). It is assumed that all flights and regulations are known at the beginning of a simulation in terms of type, amount and time. The system state is determined by the flight states, that is delays, MPRs and slot (pre-)allocations. With a new iteration, the simulation time as well as the flight and slot allocation states are updated. Departure time slots shall be issued 2 h before the EOBT at the operational slot issue time 1 (SIT1) [21]. Until a flight is released, slots are pre-allocated, i.e., they can still be re-allocated to another flight. Non-ATM delays of up to 30 min are attributed randomly in a time interval prior to the EOBT, while reactionary delays are added when the preceding flight is issued.

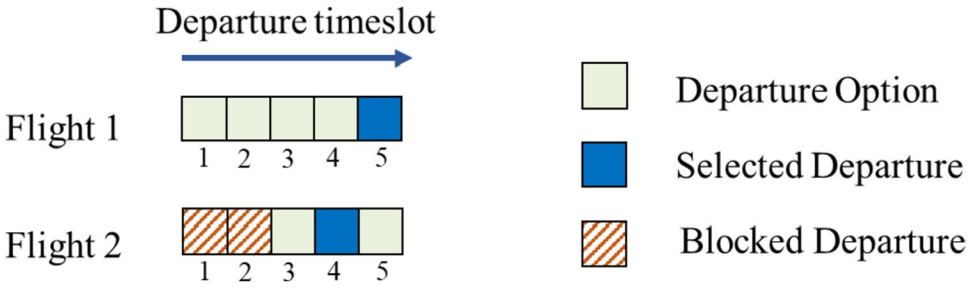

Fig. 4 Departure slot structure of two subsequent flight legs and blocked departures due to reactionary delay. Flight 1 is allocated to its fifth departure timeslot. For Flight 2, its first and second timeslots are blocked, due to reactionary delay from flight 1. Flight 2 is allocated to its fourth departure timeslot

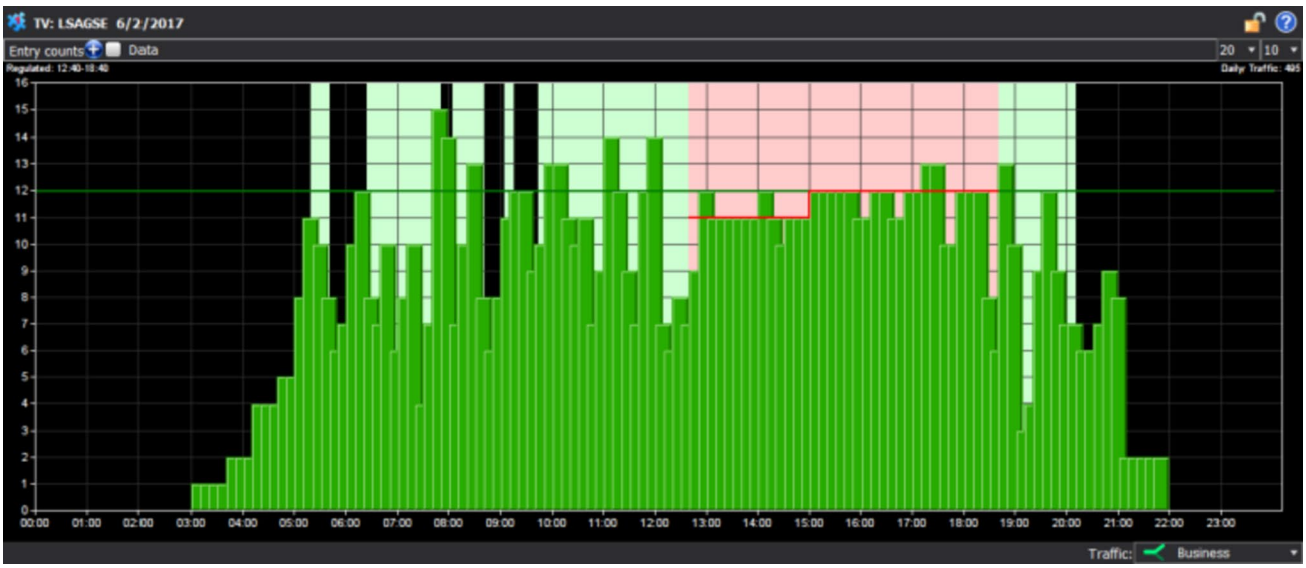

Fig. 5 Exemplary entry counts of regulated traffic volume LSAGSE in R-NEST after ISA–CASA resolution. Green bars are entry counts in overlap intervals of 20-min integration windows with 10 min sliding step. The green line represents nominal capacity, whereas the red line represents the regulated flow rate

Reactionary and non-ATM delays cause off-block and entry time updates, potentially invalidating pre-allocated slots. Time is iterated in 5-min intervals, called true revision intervals. The DCB module is called after each simulation time step, obtaining a potentially unbalanced system state, that should be reconciled. The DCB module computes ATFM delays and slots to resolve DCB imbalances. Allocation consistency is checked by R-NEST.

2.5 ISA–CASA

ISA–CASA is a heuristic DCB module in R-NEST that emulates the operational CASA slot allocation functionality for tactical FPFS solutions to network imbalances (Fig. 5 shows a resolved regulation). Initially, CASA creates a slot allocation list (SAL) in each regulation, allocating one flight after the other. Flights are ordered by the first-planned–first-served principle, prioritized by their planned time over (ETO) into the regulation. The delay of a flight in a regulation is the difference between the entry time of the allocated slot and the planned entry time (delay = CTO − ETO). In general, when one flight is in several regulations, they require different delays that are incompatible. CASA iterates through the regulations and slots to resolve these incompatibilities with the most penalizing regulation [21]. The regulation that forces the highest delay value on a flight is designated as MPR and this delay is selected for the flight. The flight is *forced* into the other regulations with this delay and it will receive a slot. If there is another flight in the candidate slot, it will be pushed out of the slot, requiring subsequent reconsolidation. Entry slots in other (non-MPR) regulations of this flight should be within the window width tolerance, which is defined in each regulation to a value between 200 and

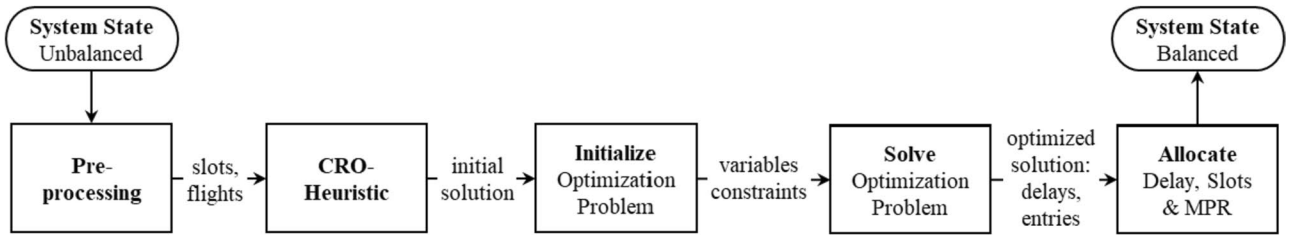

Fig. 6 Constraint reconciliation optimization (CRO) flowchart

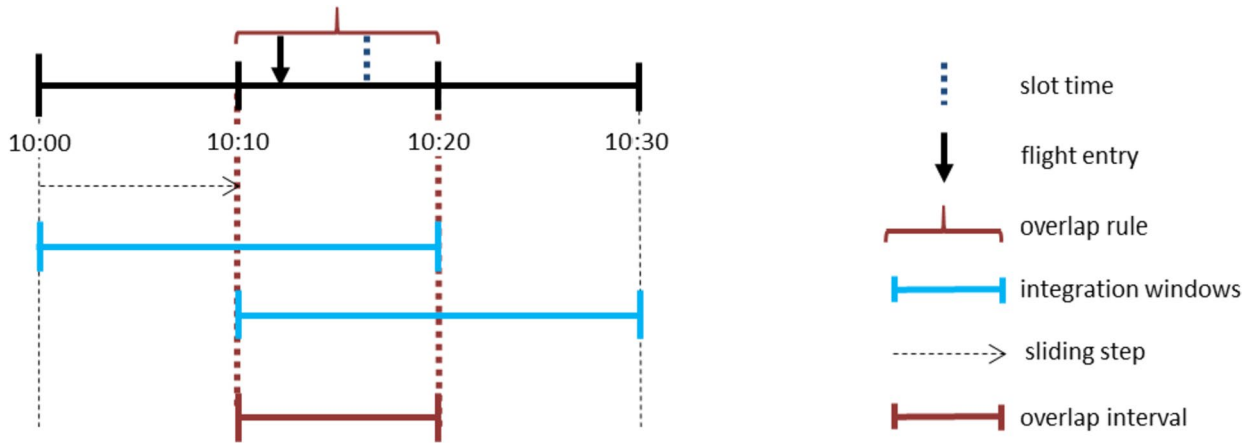

Fig. 7 Overlap interval rule. When all flights have a slot which is in the same overlap interval as the entry time, all integration windows capacities are met. With a 20-min integration window and 10-min sliding steps, the overlap interval is 10 min. In this case, the integration window is composed of two overlap intervals, and the capacity of the overlap intervals is half of the capacity of the integration window

600 s. If no slot is found, an overload slot may be created in the next window [8], which may lead to overloads of the integration windows.

3 Constraint reconciliation optimization (CRO)

CRO is a DCB module in R-NEST that minimizes primary and reactionary delays to reconcile network imbalances by generating and solving a large-scale ATFM optimization problem using the branch-cut-and-price approach. Initially, CRO receives an unbalanced network state, consisting of flights, regulation entries, and slots. In pre-processing, issued and exempted flights are allocated to the closest succeeding slots. With the remaining flights and available slots, an initial solution is generated heuristically. The resolved flights and slots are used to initialize an optimization problem with a feasible solution. The problem is solved incrementally using the column generation method. From the final solution, delays, MPR, and slots are allocated and CRO terminates with a balanced network state (see Fig. 6).

3.1 Overlap interval rule

Slots are allocated within slot tolerances and in the case of CASA, the tolerances depend on the resolution process. MPR depends on all currently allocated flights; when one flight is amended, other flights may be affected. To include tolerances in the optimization, they must be applied independently of the resolution process. To balance delays and overloads, CRO applies the *overlap interval* rule. The basic idea is that capacity integration windows overlap with sliding steps and that capacity compliance of each integration window is guaranteed by the capacity compliance of all overlap intervals contained in it (see Fig. 7).

According to the overlap interval rule, a flight can be allocated to all slots that are in the same overlap interval as the

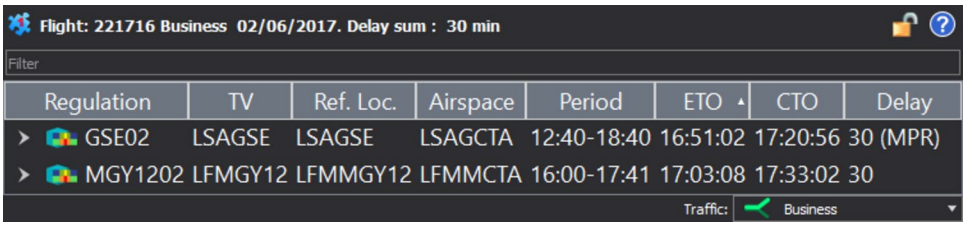

Fig. 8 Slot-entry-list: Flight 221716 enters two regulated traffic volumes, is allocates to a slot in each, and receives a 30 min delay from ISA–CASA, with the most penalizing regulation being GSE02

entry time. Accordingly, no flight will receive a slot outside the overlap interval of its entry time. This provides that the allocated slot and the entry time are in the same integration windows (see Fig. 7), as opposed to different integration windows (see Fig. 3). When all flights arriving in an overlap interval are allocated to a slot, capacity compliance is achieved in all integration windows.

3.2 Flight preprocessing

CRO flight data pre-processing distinguishes between fixed flights and flights to be shifted according to ATFM delay minimization. Exempted and issued flights are fixed; therefore, they do not receive an ATFM delay. If a flight's slot violates the overlap interval rule, i.e., the slot is in a different overlap interval than the entry time, an alternative slot must be allocated. Fixed flights are allocated to the nearest available slot in the same overlap interval. If no slot is available, the fixed flight contributes to overload. The remaining flights to be optimized, are released to free up the remaining slots for optimization.

3.3 Slot-entry-lists

If a flight's entry time is within a regulation, the flight requires a slot. When a flight enters more than one regulation (as in Fig. 8), it requires a slot in each of these regulations. A list of slots, one for each regulation entry of a flight, is referred to in this document as a slot-entry-list, containing a matching delay value. All available slots within the entry time overlap interval can be allocated. In general, with multiple slots in an overlap interval, one of multiple slots can be allocated for a given entry time. If a flight passes multiple regulations, multiple slots are available for each of these regulations. This means that there are many possible permutations for the allocation of slots for a flight. Consequently, there may be several possible slot-entry-lists for each possible delay value. Selecting a feasible combination of slot-entry-lists for all flights is a combinatorial challenge.

3.4 Most penalizing regulation (MPR)

The MPR is defined as the regulation that causes the greatest delay to a flight [21]. As described above, CASA iterates through the flights and regulations and designates the MPR in the solution *process*. In contrast, the optimization problem reconciles all flights and regulations *at once*, so it is not clear which one is the most penalizing one. However, to evaluate who caused the delay, there must be a regulation designated as MPR that is *responsible* for the delay. CRO defines the MPR as the regulation with the maximum time to slot (difference between slot time and ETO) of all entries. The other regulation slots could be reached with less delay being allocated, so they are less penalizing. Therefore, in CRO, the MPR depends on the final slot-entry-list of the flight.

3.5 Initial heuristic

We introduce an initial heuristic that finds feasible solutions to initialize the optimization. In contrast to the optimization approach, the initial heuristic processes all flights sequentially to allocate the earliest slots. If there are free slots for all entries within the slot tolerance, they are allocated without delay allocation. When a flight violates a slot tolerance, the lowest delay that allows a slot for all entries is selected. If no valid slot entry list is found, the flight is allocated to the end of the regulation. Each flight is processed once, so there is no reconciliation of linked flights as in ISA–CASA. This method guarantees that a feasible solution is found for each flight and that the final system state represents a feasible solution to the optimization problem.

3.6 Binary integer problem (BIP)

The BIP represents the challenge of finding slots for all flights with minimum delay.

Sets:

- F: The set of all flights.
- S: The set of all slots.
- L_f: The set of slot-entry-lists available to *f*.

Indices:

- *f*: Flight.
- *s*: Slot.

- l: Slot-entry-list of a flight f containing its delay d and slots s.

Parameters:

- d_f: Delay of a flight f.
- c: Reactionary delay cost coefficient.
- r_f^d: Dependent variable of the reactionary delay of flight f with delay d.
- ${}^s a_f^l$: Is equal to 1 if flight f with list l has slot s, 0 otherwise.
- w_f^l: Delay cost of flight f in slot-entry-list l.

Decision variables:

x_f^l: Is equal to 1 if flight f is allocated to list l, 0 otherwise

$$w_f^l = d_f + cr_f^d \tag{1}$$

$$\min_x \sum_{f \in \mathrm{F}} \sum_{l \in \mathrm{L}_f} w_f^l x_f^l \tag{2}$$

$$\text{s.t.} \sum_{l \in \mathrm{L}_f} x_f^l = 1 \forall f \in \mathrm{F} \tag{3}$$

$$\sum_{f \in \mathrm{F}} \sum_{l \in \mathrm{L}_f} {}^s a_f^l x_f^l \leq 1 \forall s \in \mathrm{S} \tag{4}$$

$$x_f^l \in \{0, 1\} \forall f \in \mathrm{F}, \forall l \in \mathrm{L}_f \tag{5}$$

For a flight f, its delay d and assigned slots s are represented by a slot-entry-list l. The decision to schedule the flight in this manner is represented by a binary decision variable x_f^l (5). Each variable has a delay cost term w_f^l calculated from the primary delay d of its slot-entry-list and the corresponding reactionary delay r_f^d multiplied by the reactionary delay cost coefficient c (1). A start constraint for each flight ensures that exactly one slot-entry-list x_f^l is selected, i.e., that each flight departs only once (3). All slots are capacity constrained to one entry and the parameter ${}^s a_f^l$ specifies which variables require the slot (4).

The cost function (2) is minimized while satisfying all constraints (3–5). Each possible solution contains exactly one slot-entry-list for each flight, with a delay value in minutes. Each feasible solution satisfies all constraints, and each flight entering a regulation time window has a slot in the matching overlap interval. The cost of a solution is the sum of all primary and reactionary delay costs. An optimal solution has the lowest total cost possible, given that all constraints are satisfied. This means that there is no better solution without violating at least one constraint. It's important to note that in the context of this work, "cost" refers to optimization factors, not monetary expenses.

3.7 Solving the optimization problem

In CRO, each flight has many possible combinations of delays and slot-entries. Since each combination is represented by a single variable, there are many variables in the problem formulation and an exponentially large number of feasible solutions. The linear relaxation, i.e., the relaxation of the integer condition, is solved efficiently by the simplex algorithm. The optimal solution of the relaxed problem is a lower bound for the optimal solution of the integer problem, since the more constrained integer problem cannot be better than the less constrained continuous problem. The difference between the lower bound and the upper bound is the optimality gap, which indicates how much better the optimal solution could theoretically be. We use the solving constraint

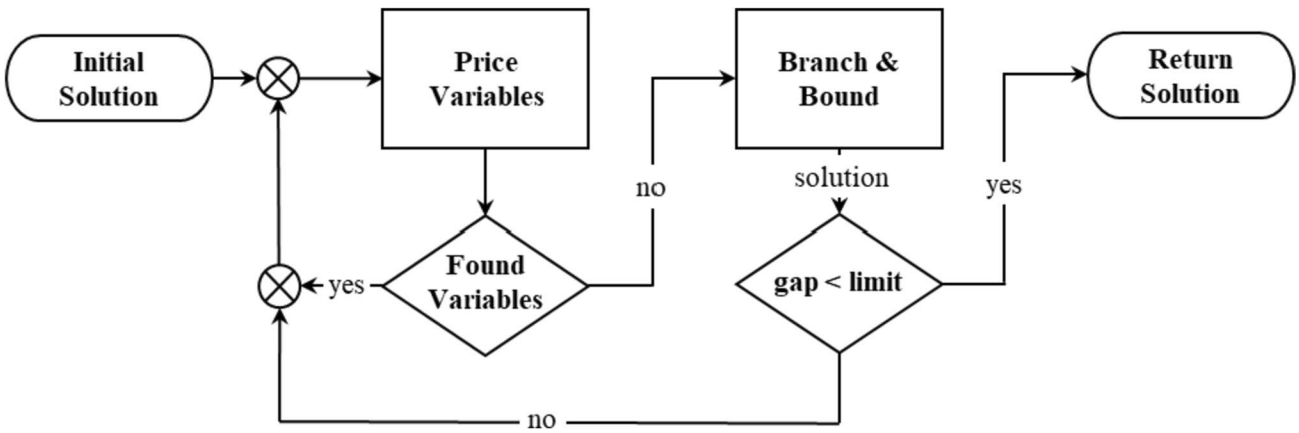

Fig. 9 Column generation workflow

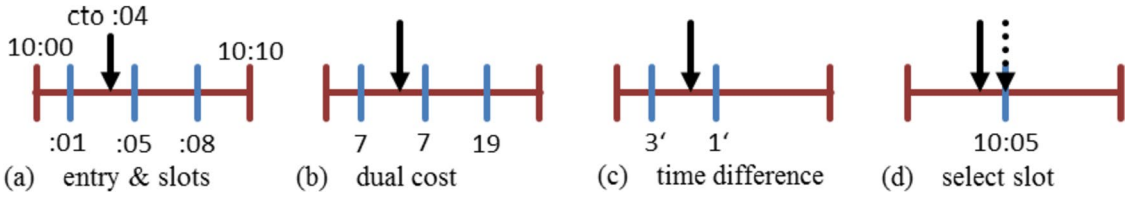

Fig. 10 Selecting a slot with the lowest dual cost and lowest time difference in an overlap interval in the pricing process

integer programs (SCIP) framework [24, 25] to find the best possible solution.

Column generation reduces the computational time for large linear optimization problems by considering only a subset of all possible variables. The basic idea is that the optimal solution of a subset of variables is a global optimum—if all other variables do not have reduced costs. The reduced costs come from the duality theory of linear optimization and can be computed using dual variables [26]. Since the reduced cost can be computed for each departure option and the respective slots, the theory can be applied here.

CRO starts column generation with only a subset of all possible variables and iteratively adds those variables to the problem that could reduce the cost of the solution (see Fig. 9). An initial set of variables is generated from the current network state of R-NEST. CRO then iteratively creates new variables that have the potential to reduce the overall delay cost of the solution. If the reduced costs of a candidate variable are negative, a solution that includes that variable could have a lower cost than the current best solution. Reduced costs are computed from delay costs and dual variables of the variable's start- and slot-constraints [6]. In the optimization problem, each departure time can be mapped to different slots for each entry, so the slot entry list with the most reduced cost is of interest. If more than one candidate slots in the same overlap interval (Fig. 10a) have the same minimum dual cost (Fig. 10b), the one with the lowest time deviation (Fig. 10c) is selected (Fig. 10d).

Variables are generated for one flight after another. Initially, variables are created with zero delay. Then, the delay is gradually increased and variables are generated until the maximum delay is reached. The maximum delay occurs when a flight entry is at the end of the regulation, which corresponds to the highest possible delay. The variable with the most reduced cost is added to the problem. When no more variables with reduced costs are found, SCIP starts the branch-and-bound process.

3.8 Setting the flights (postprocessing)

CRO passes the solution to the ATM problem to R-NEST. For each optimized flight, the primary ATM-delay is set, slot entries are allocated and MPR is assigned. When CRO has finished updating the system state, the simulation continues. After every True Revision Interval, CRO performs the next reconciliation, until the day's simulation is complete.

Table 1 Three scenarios and the activated delay types

Scenario name	Activated delay type		
	ATFM	Reactionary	Non-ATM
Static	x		
Dynamic I	x	x	
Dynamic II	x	x	x

4 Scenario setup

Three realistic ATS scenarios are simulated[1] with historical European traffic data for June 02, 2017 (6th AIRAC cycle), a regular Friday with a summer flight schedule. They are defined by the types of delays being considered, with more delay types associated with an increasing order of complexity (see Table 1). Scenarios of lower complexity facilitate the interpretation of the slot allocation mechanism. The first scenario contains only ATFM delay. It is static, because the initial solution is not disturbed in subsequent iterations, i.e., it does not need to be recomputed. The second scenario contains ATFM delay and reactionary delay, whereas the third scenario also contains non-ATM delay. These two scenarios are dynamic, because reactionary (and non-ATM) delays tactically disturb the system state, creating new slot conflicts that must be reconciled. The random number generator always produces the same non-ATM delay with the same 174,160 min in all simulations.

All scenarios contain the same traffic demand and regulation scheme, i.e., the regulations, rates, and durations are all the same. Only ATC capacity and staffing regulations are activated. Basic data are tabulated in Table 2. With the exception of delay types, all other scenario data, R-NEST

[1] We use a PC with Intel Core i7-6700 CPU @ 3.40 GHz with 32 GB RAM.

Table 2 Scenario data and settings

	Value
Scenario data	
Day	2nd June 2017
Number of flights	34,007
Number of flights impacted by regulations	6815
Number of regulations: ATFCM staffing and capacity reason	81
Number of slots in regulations	5663
Number of entries (flights in regulations)	9475
R-NEST ATFCM settings	
True revision interval	5 min
Slot issue time (SIT1)	2 h
Entry count integration windows	20 min
Sliding step	10 min
CRO optimization settings	
Overlap interval	10 min
Optimality gap limit	1%
Minimum reduced cost of variable	10 s
Minimum delay increment of a priced variable	10 s

settings, and CRO settings are consistent. To reduce computation time, CRO parameters allow deviations from the theoretically optimal solution. Candidate variables are searched with delay increments of 10 s and added to the problem only if the cost is reduced by 10 s. The optimality gap limit is set to 1%.

Some flights are affected by only one regulation, whereas many flights face two or more regulations (see Table 3), which means that there are many interrelated slot-entry-lists.

4.1 Limitations

The European ATS is very complex. To carry out realistic and valuable research, some simplifications have to be made to allow interpretability. In addition, some phenomena are simply too complex for the selected modelling approach, as human actions are often involved in the operations. Therefore, this study has some limitations, mainly due to the dynamic nature of the EATMN.

1. All flights are known at the beginning of the simulation. There are no new flight plans, resubmitted flight plans, or diversions. In addition, except for non-ATM delays, flights strictly adhere to their schedules. There are no other uncertainties in the flights.

Table 4 Static: final network stats

DCB-module	Primary delay [min]	Overloads
ISA–CASA	24,732	66
CRO	10,871	61

2. All regulations and their declared flow rates are known at the beginning of the simulation. Sector configurations and regulations are constant, and there are no FMP actions to adapt to predicted demand/capacity imbalances.
3. Reactionary delay relationships are based on available tail numbers. Buffer times are the same for all relationships. No additional mechanisms are modeled to compensate for reactionary delay.

5 Results

The three scenarios described above are simulated in R-NEST over the course of the day according to the process in Fig. 2, with DCB modules either ISA–CASA or CRO being called every 5 min of simulation time. For each scenario that includes reactionary delay, we perform the trade-off for different reactionary delay cost coefficients. The larger the coefficient, the more reactionary delays are penalized relative to primary delays. Simulations are run for the entire day with a fixed cost coefficient. Reactionary delay cost coefficients range from zero (no penalty for reactionary delays), one (primary and reactionary delays are penalized equally), and infinity (emphasis on reactionary delay reduction). Each setting is simulated once, and delays, computation times, and remaining overloads are evaluated.

5.1 Static scenario

The Static scenario does not include any reactionary or non-ATM delay, and no reactionary delay cost trade-off is performed. CRO finds a near-optimal solution and allocates slots and delays accordingly. Since there are no slot inconsistencies and no external changes to the system state occur when the simulation is iterated to the next time step, CRO does not need to re-optimize and change slots. The final network stats are shown in Table 4. Compared to ISA–CASA, CRO, reduces the remaining overloads by 8% and the primary delay by 56%. The maximum runtime of CRO is 7.85 s. The optimality gap is 0.34%, which means that the optimal solution could be better by a maximum of

Table 3 Number of flights with the number of regulations

# Regulations	1	2	3	4	5	6	7
# Flights	4822	1458	434	81	11	7	2

Table 5 Dynamic I: final network stats for solution modules ISA–CASA and CRO with reactionary delay penalization between 0.0 and infinity (minimum value is indicated in bold)

DCB-Module	Primary delay [min]	Reactionary delay [min]	Primary + reactionary [min]	Overloads
ISA–CASA	24,604	14,498	39,102	73
CRO-0.0	**10,698**	6498	17,196	**63**
CRO-0.1	10,908	2906	13,814	66
CRO-0.5	11,292	1446	12,738	**63**
CRO-1	11,505	1220	**12,725**	64
CRO-2	11,828	1045	12,873	65
CRO-10	12,557	815	13,372	67
CRO-*inf*	12,833	**728**	13,561	65

this percentage, or 37 min. This shows that CRO is able to find good solutions that significantly reduce delays while maintaining current levels of overload, quickly, and close to optimal.

5.2 Dynamic I: ATFCM and reactionary delay

In the Dynamic I scenario, reactionary delay is enabled and reactionary delay is penalized in the cost function. We perform cost trade-offs with reactionary cost coefficients ranging from zero to infinity (CRO-0.0 to CRO-inf). After each CRO run, time is advanced and reactionary delays are induced on subsequent flights. These subsequent delays result in slot inconsistencies and create demand–capacity imbalances. The final network stats are shown in Table 5.

The remaining overloads of CRO resolutions with different reactionary cost coefficients are irregularly distributed from 63 to 67, below the 73 ISA–CASA overloads, which is a reduction from 8% to 14%. The primary and reactionary delays are shown in Fig. 11.

Compared to ISA–CASA, CRO reduces primary and reactionary delays for all reactionary delay cost coefficients.

As the coefficients increase, primary delays in CRO steadily increase from 10,698 to 12,833 min, and reactionary delays, which are all lower, steadily decrease from 6498 to 728 min. The minimum total delay is achieved with an equal delay penalty coefficient of 1.0.

Different reactionary delay cost coefficients in CRO are compared. A small reactionary delay cost coefficient of 0.1 reduces reactionary delay by 3592 min compared to zero reactionary delay cost at the detriment of increasing primary delay by 210 min. A slightly larger coefficient of 0.5 reduces reactionary delay by 5052 min at the detriment of increasing primary delay by 594 min, so the additional savings of 1460 min of reactionary delay comes at the detriment of additional 384 min of primary delay compared to CRO-0.1. A coefficient of 1.0 reduces reactionary delay by 5278 min at a detriment of 807 min of additional primary delay. Compared to CRO-0.5, the additional savings in reactionary delay is 226 min at a detriment of 213 additional minutes of

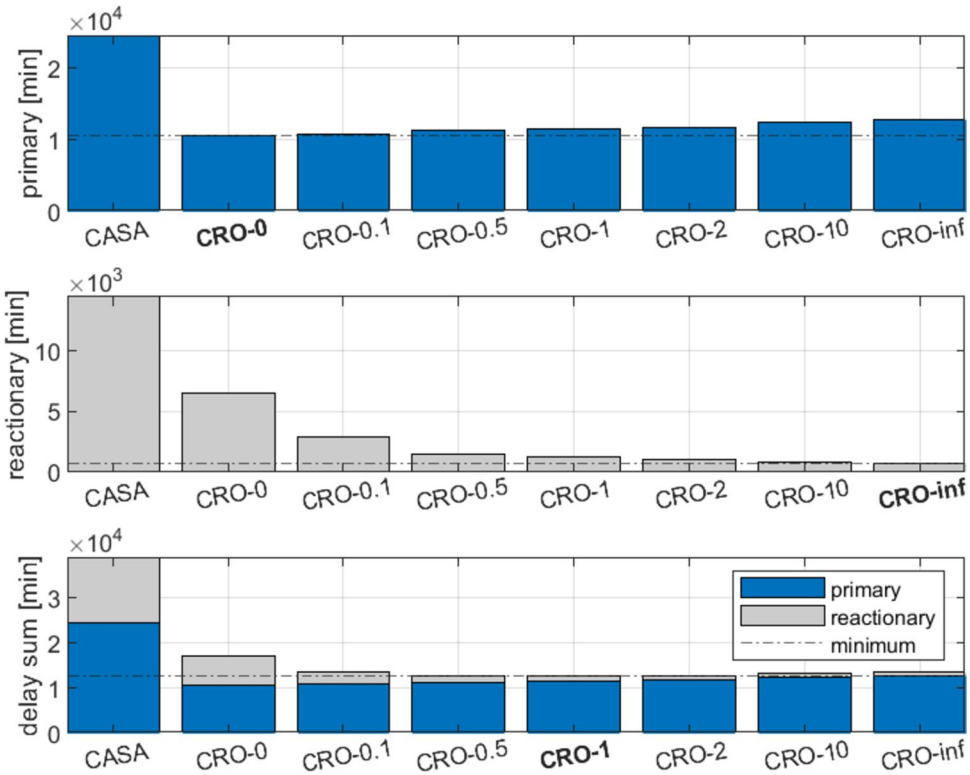

Fig. 11 Dynamic I: primary and reactionary delay for ISA–CASA and CRO (DCB–module of minimum value is indicated in bold)

Table 6 Dynamic I: solving stats, maximum runtime and maximum gap per CRO call, mean runtime and mean gap for the whole day (minimum value bold, maximum value italic)

CRO coefficient	Max runtime (s)	Mean runtime (s)	Max gap (%)	Mean gap (%)
0	9.7	2.9	*0.62*	*0.02*
0.1	8.6	2.8	0.30	**0.00**
0.5	**7.2**	**2.7**	0.54	0.01
1	9.6	3.2	0.25	**0.00**
2	10.6	3.2	0.47	**0.00**
10	11.7	3.2	0.26	**0.00**
inf	*13.8*	*3.6*	**0.04**	**0.00**

Table 7 Dynamic II: final network stats for solution modules ISA–CASA and CRO with reactionary delay penalization between 0.0 and infinity (minimum bold)

DCB-Module	Primary delay (min)	Reactionary delay (min)	Primary + reactionary (min)	Overloads
ISA–CASA	36,676	196,659	233,335	57
CRO-0	21,811	185,422	207,233	39
CRO-0.1	**21,434**	178,441	199,875	40
CRO-0.5	21,498	175,779	197,277	41
CRO-1	22,265	174,761	**197,026**	39
CRO-2	22,215	174,812	197,027	**36**
CRO-10	23,346	174,487	197,833	41
CRO-inf	22,668	**174,466**	197,134	37

primary delay. Larger reactionary cost coefficients increase primary delays more than they reduce reactionary delays.

Comparing the best CRO solution with a penalty of 1.0 with ISA–CASA, CRO reduces primary delay by 13,099 min, reactionary delay by 13,278 min and the total delay by 26,377 min.

The maximum runtime for a single CRO execution ranges from 7.2 to 13.8 s. The mean runtimes for all CRO runs of a simulated day range from 2.7 to 3.6 s. The maximum optimality gap is less than one percent. Solving stats are given in Table 6.

5.3 Dynamic II: ATFCM, reactionary and non-ATM delay

In the Dynamic II scenario, the non-ATM delay is activated as a random system disturbance in addition to the reactionary delay. After each CRO run, time advances and flights receive non-ATM and reactionary delays. The random off-block time deviations create slot inconsistencies and new imbalances. The given non-ATM delay is much larger than the primary delay and is, therefore, the driving factor for the reactionary delay. The simulation results and final network stats are summarized in Table 7.

Overloads for CRO computations with different reactionary cost coefficients are scattered between 36 and 41, reducing ISA–CASA overloads by about 28–34%. Primary and reactionary delay values are shown in Fig. 12:

Compared to ISA–CASA, CRO also reduces primary and reactionary delay in this scenario for all reactionary delay cost coefficients.

In CRO, for reactionary delay cost coefficients between zero and inf, the primary delay ranges from 21,811 to 23,346 min, while the reactionary delay is about seven times greater, ranging from 197,026 to 207,233 min.

A comparison of the different reactionary delay cost coefficients in CRO shows that as the coefficient increases, primary delays tend to increase and reactionary delays tend to decrease. However, counterintuitively, CRO-0.1 and CRO-0.5 have lower primary delays than CRO-0. Compared to zero reactionary delay penalty, a coefficient of 0.1 reduces primary delay by 377 min and reactionary delay by 6981 min. A coefficient of 0.5 reduces primary delay by 313 min and reactionary delay by 9643 min. A coefficient of 1.0 results in the lowest total delay, increasing primary delay by 454 and decreasing reactionary delay by 10,661 min. For larger reactionary penalties, total delays are greater.

Specifically, when comparing CRO-1 and ISA–CASA, primary delay is reduced by 14,411 min, while reactionary delay is reduced by 21,898 min and total delay is reduced by 36,309 min.

The maximum runtime for a single CRO run is 13.8 s. The maximum mean run time for all CRO runs to reconcile a full day is 3.8 s. The maximum optimality gap is 0.98% and the maximum mean gap is 0.09%. The solving stats for CRO are summarized in Table 8.

6 Discussion of the results

One static and two dynamic scenarios, each consisting of a complete daily sample of European air traffic, are resolved iteratively. Two solution strategies are applied: ISA–CASA, a first-planned–first-served-based heuristic, emulating the operational CASA, and CRO, an optimization approach based on column generation, including penalization for reactionary delays depending on a coefficient.

The remaining overloads of the scenarios resolved by CRO are between 8% and 34% lower than those resolved by ISA–CASA, indicating the efficiency of the overlap interval rule. Different reactionary delay cost coefficients showed no systematic influence on the remaining overloads.

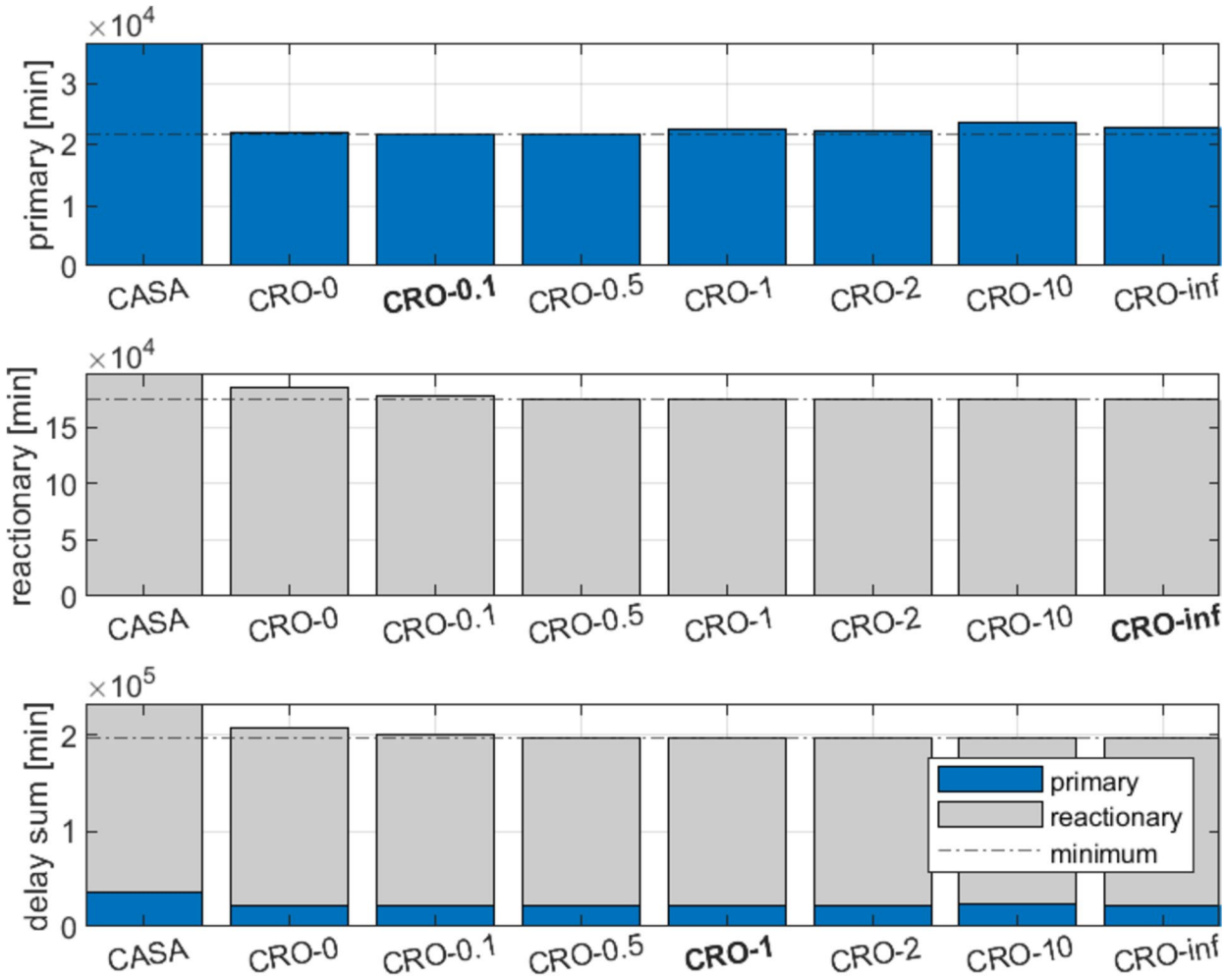

Fig. 12 Dynamic II: primary and reactionary delay for ISA–CASA and CRO (DCB–module of minimum value is indicated bold)

Table 8 Dynamic II: solving stats, maximum runtime and maximum gap per CRO call, mean runtime and mean gap for full day (maximum value italic, minimum value bold)

CRO coefficient	Max runtime (s)	Mean runtime (s)	Max gap (%)	Mean gap (%)
0	11.3	3.2	*0.98*	*0.09*
0.1	10.8	**3.2**	0.97	0.08
0.5	**10.3**	3.3	0.86	0.04
1	10.5	3.5	0.92	0.06
2	10.5	3.5	0.92	0.06
10	12.4	*3.8*	*0.98*	0.04
inf	*13.8*	*3.8*	**0.55**	**0.01**

Comparing the static and the dynamic I scenarios, the primary delays are slightly smaller in the latter, with a difference of 0.5% for ISA–CASA and 1.6% for CRO-0. This shows that the reactionary delay alone does not disturb the flights as much as to increase the primary delays. For both resolution methods, there are significant amounts of reactionary delays, i.e., 59% and 61% of the primary delays.

When comparing the static and the dynamic II scenarios, primary delays are much higher in the latter, with an increase of 48% for ISA–CASA and 101% for CRO-0. Presumably, as flights are pushed into the regulations by non-ATM delays, there is less capacity available, and flights require more primary delays to obtain slots.

In addition, compared to the dynamic I scenario, in the dynamic II scenario, the primary delays are by 49% higher for ISA–CASA and 104% higher for CRO-0, while the reactionary delays are more than ten times higher (as expected from the non-ATM delay).

CRO reduces the primary delay by 56% compared to ISA–CASA in the static scenario, indicating that optimization can improve reconciliation. In the dynamic I scenario, CRO-0 reduces the primary delay by 56.5% and the reactionary delay by 55.2% compared to ISA–CASA, for a total delay reduction of 56%, showing that optimization can reduce reactionary delay even without reactionary delay penalties. In the dynamic II scenario, CRO-0 reduces primary delay by 40.5%, reactionary delay by 5.7%, and total delay by 11.2%.

Cost trade-offs with different reactionary delay cost coefficients are performed in the Dynamic I and II scenarios. In most cases, increasing the reactionary delay cost coefficient results in a smaller amount of reactionary delay at the cost of increasing the primary delay (see Figs. 11, 12). Equally penalizing both types of delay results in an overall delay minimization.

In the dynamic I scenario, comparing the delay-minimizing CRO-1 (equal penalty) setting with ISA–CASA, the primary delay is reduced by 53.2% and the reactionary delay is reduced by 91.6%. The total delay (primary and reactionary) delay is reduced by 67.5%. Reactionary delay is reduced by as much as 95% when the optimization focus is entirely on reactionary delay. In the dynamic II scenario, comparing CRO-1 to ISA–CASA, primary delay is reduced by 39.3% and reactionary delay is reduced by 11.1%. The sum of primary and reactionary delay is reduced by 15.6%. With infinite emphasis on reactionary delay, the sum is reduced by 11.3%.

Primary delay evolves differently compared between the static and dynamic I scenarios, decreasing slightly by 128 min for ISA–CASA while increasing by 653 min for CRO-1, showing that CRO allows an increase in primary delay to reduce reactionary (and total) delay. The difference in primary delay between the Dynamic I and Dynamic II scenarios is greater, with an increase of 12,072 min for ISA–CASA and 10,760 min for CRO-1, indicating that both DCB modules are affected by the non-ATM delays.

An increasing reactionary delay cost coefficient results in decreased reactionary delay and increased primary delay in the dynamic I scenario. On the other hand, in the dynamic II scenario, the effect of the reactionary delay cost coefficient is not as clear, with coefficients between 1 and inf converging to reactionary delay between 174,466 and 174,812 min. One explanation is that in the dynamic II scenario, most of the reactionary delay is caused by non-ATM delay, which is not affected by DCB.

Accordingly, the relative reductions by CRO are smaller in the dynamic II scenario than in the Static and dynamic I scenarios. CRO tends to assign delays just before buffer times, so that when a flight encounters non-ATM delay, a reactionary delay occurs. When all buffers are exhausted, it is impossible to avoid reactionary delay at the network level. These relative reductions must be put into perspective with the absolute delay reductions, which are even greater than in the dynamic I scenario.

An optimality gap of 1% relative to the current problem is achieved for all scenarios. Computation times peak at 14 s, well below the true revision interval of 5 min.

7 Conclusion and outlook

In this work, we introduce a new module for Slot Allocation in an existing simulation framework for European Air Traffic Management (R-NEST). This module CRO resolves demand–capacity imbalances for all flights and regulations by optimization, with the capability to penalize reactionary delays. The simulation contains flight plans, rotation margins, sector configurations, regulations, and deterministic or stochastic delays.

Three scenarios with increasing levels of complexity and realism are resolved. The first scenario (static) contains flight plans, sector configurations, and regulations. The second scenario (dynamic I) includes flight rotations, so that if a flight is delayed, a later flight in the rotation will also be delayed. The last scenario (dynamic II) includes stochastic departure time deviations (non-Atm delay).

CRO reduces primary and reactionary delay for the scenarios while maintaining low levels of overload. Cost tradeoffs with different reactionary delay cost coefficients show that in most cases, increasing the coefficient reduces reactionary delay while increasing primary delay. Equivalent penalty minimizes total delay.

Comparing the relative delay reductions from ISA–CASA to CRO-1 between the dynamic I and dynamic II scenarios, the relative reductions are lower in the latter. The reason for this is that the large amount of non-ATM delay causes high reaction delays. This indicates the limitations of optimization when a large portion of the delay is non-ATM delay. On the other hand, the absolute reductions are greater than in Dynamic I, with a total delay reduction of approximately 36,000 min, indicating significant efficiency gains from optimization.

The optimization results should be interpreted with the caveat that CRO does not take into account the first-planned–first-served paradigm. In addition, it does not reserve capacity for flight plans that are initiated at a later point in time. In particular, it is important to recognize the significant role that the operational environment plays in shaping these results. To get a broader picture of the CRO performance, more traffic samples could be resolved, representing different nominal and non-nominal network cases. Potential computational bottlenecks or solution degradation

for more constrained scenarios need to be investigated. The flexibility of the column generation process can be used to trade off computation time against solution quality.

Primary and reactionary delays are distributed differently when different resolution methods and penalties are used. For the price of a small increase in primary delays, reactionary delays can be substantially reduced compared to CRO-0. The cost coefficient provides some control in the optimization process, i.e., the model can be tuned to minimize the primary delay, the secondary delay, or by any intermediate ratio. However, it is not possible to know which parameters are optimal in terms of airline cost, network stability, and/or fairness, so a broader discussion is warranted.

Modifying an automated decision-making system for resource allocation such as DCB requires careful consideration. A common requirement is that the resolution method be acceptable to the relevant stakeholders, such as passengers, airlines, air traffic control, airports and network management. A normative and an empirical approach could be combined to ensure acceptability, culminating in workshops and surveys. To enable the stakeholders to assess fairness in the technically difficult topic of DCB, it must be prepared in such a way that slots, reactionary delays, heuristics, optimization, and fairness, can be debated in a reasonable amount of time—without priming any ethical judgments.

To structure the debate, it is useful to define who the claimants are, what the reasons for the claims are, what the strength of the claims are, how they relate to each other, and how they are weighted-up. At the heart of the matter is the resolution of competitions involving multiple claims, whether multiple flights require slots in the same regulations, or flights threatened with reactionary delay demand some satisfaction of their claims. In addition, technical details such as slot tolerances affect the relationship between competing flights to some extent. Their influence on sequencing should be evaluated and their relevance to fairness should be determined. However, the contribution of deterministic factors is blurred as sequences are shaken throughout the day by, for example, non-ATM delays. In summary, determining a level of fairness for a system state or a resolution method requires ethical judgments involving a variety of technical details. Accordingly, the applicability of optimization to ATFM relies on a rigorous assessment of fairness.

Beyond addressing fairness concerns, the ability to interoperate with more realistic scenarios, such as hotspot declarations, late filers, or en-route uncertainties could be explored. Additional cost functionalities for equity, cancellation thresholds, or user-driven priorities could be incorporated. How the novel imbalance reconciliation philosophy would interact with dynamic airport (APT), flow management position (FMP), and airspace user (AU) decision making could be explored through functionally extended simulations.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s13272-023-00708-4.

Acknowledgements The authors thankfully acknowledge the support of EUROCONTROL with respect to the provision of large-scale traffic and ATM environmental data, and also thank Hamid Kadour and Nicolas Boulin for their aid concerning interface implementation for R-NEST. Many thanks also to the reviewers for their fruitful and thorough comments, which greatly improved the paper.

Funding Open Access funding enabled and organized by Projekt DEAL. This project has received funding from the SESAR Joint Undertaking under Grant Agreement No 731730 under the European Union's Horizon 2020.

Data availability The R-NEST tool and the underlying data are proprietary to EUROCONTROL. The CRO software code is proprietary to the German Aerospace Center (DLR). Therefore, the code cannot be made available to the public or the readers without any restrictions. A table containing aggregated results is provided. The codes used to analyze the data and to plot the analyzed data are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflicts of interests that are directly or indirectly related to this publication.

References

1. Odoni A.R.: The flow management problem in air traffic control. In: Odoni, A.R., Bianco, L., Szego, G. (eds.) Flow Control of Congested Networks, pp. 269–288. Springer, Berlin (1987). https://doi.org/10.1007/978-3-642-86726-2_17
2. Lulli, G., Odoni, A.R.: The European air traffic flow management problem. Transp. Sci. **41**(4), 431–443 (2007). https://doi.org/10.1287/trsc.1070.0214
3. Kaufhold R., Marx S., Mueller-Berthel, C., Nachtigall, K.: A pretactical generalised air traffic flow management problem. In: 7th USA/Europe ATM Seminar, Barcelona (2007)
4. Bertsimas, D., Stock, S.: The air traffic management problem with enroute capacities. Oper. Res. **46**, 406–422 (1998). https://doi.org/10.1287/opre.46.3.406
5. Balakrishnan, H., Chandran, B.G.: Optimal large-scale air traffic flow management (2014). http://www.mit.edu/~hamsa/pubs/BalakrishnanChandran_ATFM.pdf

6. Lau, A., Berling, J., Linke F., Gollnick, V., Nachtigall, K.: Large-scale network slot allocation with dynamic time horizons. In: 11th USA/Europe ATM Seminar, Lisbon (2015). https://www.semanticscholar.org/paper/Large-Scale-Network-Slot-Allocation-with-Dynamic-Lau-Berling/69b6b19b187e4086d5349a6dcdcba86cc283b744
7. Barnhart, C., Bertsimas, D., Caramanis, C., Fearing, D.: Equitable and efficient coordination in traffic flow management. Transp. Sci. **46**(2), 262–280 (2012). https://doi.org/10.1287/trsc.1110.0393
8. Ruiz, S., Kadour, H., Choroba, P.: A novel air traffic flow management model to optimise network delay. In: 13th USA/Europe ATM Seminar, Vienna (2019). https://www.semanticscholar.org/paper/A-novel-air-traffic-flow-management-model-to-delay-Ruiz-Kadour/2661477805a404644d88e638dd012c06f49495e7
9. Ivanov, N., Netjasov, F., Jovanović, R., Starita, S., Strauss, A.: Air Traffic Flow Management slot allocation to minimize propagated delay and improve airport slot adherence. Transp. Res. A **95**, 183–197 (2017). https://doi.org/10.1016/j.tra.2016.11.010
10. Montlaur, A., Delgado, L.: Flight and passenger delay assignment optimization strategies. Transp. Res. C **81**, 99–117 (2017). https://doi.org/10.1016/j.trc.2017.05.011
11. Campanelli, B., et al.: Comparing the modeling of delay propagation in the US and European air traffic networks. J. Air Transp. Manag. A **56**, 12–18 (2016). https://doi.org/10.1016/j.jairtraman.2016.03.017
12. R-NEST 1.3.1.: https://www.eurocontrol.int/solution/rnest. Accessed 26 June 2023
13. Dalichampt, M., Petit, E., Junker, U., Lebreton, J.: Innovative Slot Allocation (ISA) - Executive summary of EEC Report No. 322, Eurocontrol (1997). https://www.eurocontrol.int/sites/default/files/library/040_Innovative_Slot_Allocation.pdf
14. Eurocontrol Experimental Centre.: Innovative slot allocation : an overview. EEC note no. 10/01. Technical report, Eurocontrol (2001). https://www.eurocontrol.int/sites/default/files/library/016_Innovative_Slot_Allocation.pdf
15. Dalichampt, M., Fortunet, G., Hoffman, J., Mahlich, S., Tibichte, A.: ATFM simulation of CFMU phase 1, executive summary of EEC Report No 291. EUROCONTROL (2014). https://www.eurocontrol.int/publication/atfm-simulation-cfmu-phase-1
16. Broome J.: Fairness. Proc. Aristot. Soc. **91**, 87–101 (1990). http://www.jstor.org/stable/4545128 (ISSN 00667374)
17. Curtis, B.L.: To be fair. Analysis **74**(1), 47–57 (2014). https://doi.org/10.1093/analys/ant093
18. Taurek, J.M.: Should the numbers count? Philos. Public Affairs **6**(4), 293–316 (1977). http://www.jstor.org/stable/ (ISSN 00483915)
19. Parfit, D.: Innumerate ethics. Philos. Public Affairs, pp. 285–301 (1978). https://www.jstor.org/stable/2264959
20. Piller, C.: Treating Broome fairly. Utilitas **29**(2), 214–238 (2016). https://doi.org/10.1017/s0953820816000303
21. Eurocontrol: ATFCM Operations Manual, Ed 23.0 (2019). https://www.eurocontrol.int/publication/atfcm-operations-manual
22. EUROCONTROL: Demand Data Repository. https://www.eurocontrol.int/ddr
23. EUROCONTROL: Central Office for Delay Analysis (CODA). www.eurocontrol.int/articles/central-office-delay-analysis-coda
24. Achterberg, T.: SCIP: solving constraint integer programs. Math. Progr. Comput. **1**, 1–41 (2009). https://doi.org/10.1007/s12532-008-0001-1
25. Gleixner, A., et al.: The SCIP Optimization Suite 6.0. ZIB-Report 18-26, Zuse Institute Berlin (2018). ISSN 1438-0064. https://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/6936
26. Desrosiers, J., Lübbecke, M.E.: A primer in column generation. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.) Column Generation. Springer, Boston (2005). https://doi.org/10.1007/0-387-25486-2_1