

Towards a Heuristic Optimizer for a Target Time Management System in Air Traffic Flow Management

Sebastian Gruber*, Paul Feichtenschlager[†], Christoph Fabianek[‡], Eduard Gringinger[§], and Christoph G. Schuetz[¶]

*Johannes Kepler University Linz, Linz, Austria

ORCID: 0000-0002-5714-1870

[†] Frequentis AG, Vienna, Austria

ORCID: 0009-0007-3490-5560

[‡] Frequentis AG, Vienna, Austria

ORCID: 0009-0002-4410-8796

[§] Frequentis AG, Vienna, Austria

ORCID: 0000-0003-3897-3003

[¶] Johannes Kepler University Linz, Linz, Austria

ORCID: 0000-0002-0955-8647

Abstract—In Air Traffic Flow Management (ATFM), flights are issued arrival times during unexpected events, such as adverse weather conditions, in order to ensure smooth and safe airport operations. These target times are traditionally allocated on a first-planned, first-served basis, which often leads to inefficiencies due to the different delay costs of individual flights. The HARMONISED network through smart technology and Collaboration (HARMONIC) project, funded by the Single European Sky ATM Research (SESAR) Joint Undertaking within the EU Horizon Europe program, therefore, explores a collaborative optimization platform to enable flight prioritization across airspace users. In this paper, we describe initial results of the development and validation of a Target-Time Management System (TTMS) in the HARMONIC project. In particular, we present different architecture options, one assuming a trusted platform provider, another option for the assumption of an honest-but-curious platform provider, using secure multi-party computation to protect the private inputs. We also present experimental results for a genetic algorithm for the optimization of assignment of target times to flights.

Index Terms—flight prioritization, multi-party computation, evolutionary algorithm, heuristic optimization

I. INTRODUCTION

In Air Traffic Flow Management (ATFM), flights are assigned to specific departure and arrival times at airports. When unexpected events such as disruptive weather conditions reduce airport capacities, these assignments must be adjusted, leading to potential delays and additional costs for airspace users. Traditionally, flights are rescheduled on a first-come, first-served basis, which is not always cost-effective due to varying delay costs between individual flights.

In order to allow for more flexible assignments of arrival times in case of reduced capacity in the air traffic network, the regional model of the HARMONISED network through smart technology and Collaboration (HARMONIC) [1] industrial

research project, funded by the Single European Sky ATM Research (SESAR) Joint Undertaking within the EU Horizon Europe program, with participation from key stakeholders Zurich Airport, Swiss International Airlines, Skyguide, and EUROCONTROL, investigates the potential of a collaborative optimization platform. This project aims to develop a Target Time Management System (TTMS) that enables flight prioritization across airspace users.

The key concerns in developing a TTMS are preserving the privacy of airspace user preferences and the time available for optimization. With regard to preserving the privacy of airspace user preferences, a TTMS is hosted either by a trusted platform provider such as an airport, or by an honest-but-curious platform provider such as a third-party service provider. In the latter case, methods for privacy-preserving computation such as multi-party computation (MPC) [2] are required for optimization while preserving the privacy of airspace user preferences. With regard to the time available for optimization, a TTMS should find a solution in the order of minutes [3]. However, methods for privacy-preserving computation induce considerable additional runtime overhead.

The Heuristic Optimizer is the component of a TTMS that uses an optimization algorithm in order to find arrival flight lists. In the case of a trusted platform provider, the Heuristic Optimizer will use a deterministic algorithm that always provides optimal solutions. In the case of an honest-but-curious platform provider, a deterministic algorithm may not be able to find a solution in the time available for optimization due to privacy-preserving computation. For this purpose, the Heuristic Optimizer may use an evolutionary algorithm in order to guarantee a solution in time. In an iterative process, an evolutionary algorithm searches for solutions, and the solutions found are evaluated using MPC. When the

time available for optimization concludes, the iterative process terminates and the best solutions found so far are returned.

Optimizing arrival flight lists also affects airport operations, potentially leading to changed waiting times for passengers and higher demand for staff, among others. The optimization shall consider both airspace user preferences and airport operations, aiming for a balance that improves overall efficiency. Thus, flight prioritization in times of reduced airport capacity is a multi-objective optimization problem, where the aim is to meet the needs of airspace users and the airport despite at times conflicting goals.

In this paper, we discuss the design of a Heuristic Optimizer for a TTMS that considers both options, a trusted and an honest-but-curious platform provider. We have already conducted preliminary experiments considering an honest-but-curious platform provider using different configurations of the genetic algorithm. Our results are promising but biased, favoring airport operations because the input of the airport does not require privacy measures.

The remainder of this paper is structured as follows. Section II briefly introduces ATFM and privacy-preserving optimization. Section III discusses flight prioritization and the input for the Heuristic Optimizer. Section IV presents the architecture of the TTMS for a trusted platform provider. Section V extends the architecture of the TTMS for an honest-but-curious platform provider. Section VI presents the results of our preliminary experiments. Section VII summarizes the paper.

II. BACKGROUND

This section provides a brief introduction to ATFM and privacy-preserving optimization.

A. Air Traffic Flow Management (ATFM)

In the European air traffic network, the EUROCONTROL Network Manager regulates flights arriving at an airport in times of imbalance between the demand for arrivals and airport capacity. Therefore, the Computer-Assisted Slot Allocation (CASA) algorithm calculates target times of arrival (TTAs) - or slots - on a first-planned, first-served basis. This means that when regulating a flight sequence, the order of the flights is generally retained, but the flights are shifted back in time. For some of those flights, however, the delay is more problematic than for others due to varying delay costs.

The delay costs of a flight can be described by a non-linear function with one or more delay targets [4]. Fig. 1 illustrates a function of the delay costs of a flight. While the costs increase moderately within delay targets, they rise sharply after a delay target is missed. A delay target is an interval of arrival times that, for example, allows passengers to reach a connecting flight despite the delay.

The User Driven Prioritization Process (UDPP) [5] enables flight prioritization of an airspace user's own flights. This process improves the flexibility of airspace users while preserving equity between them. In this regard, "equity" means that one airspace user's prioritization actions only affect its flights, not

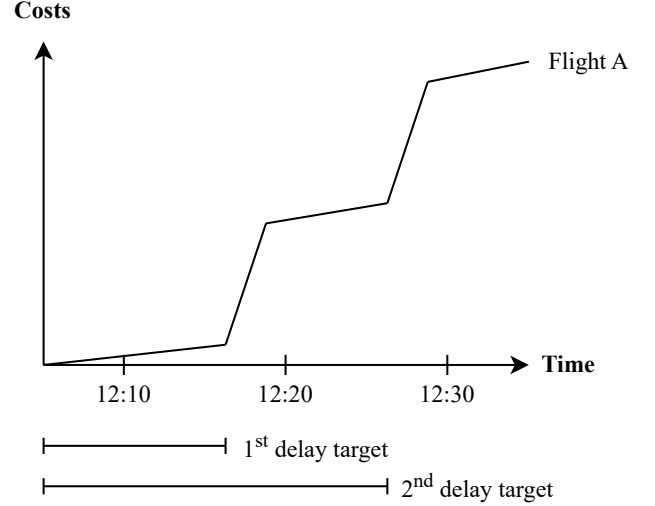


Fig. 1. Non-linear function of the delay costs of a flight based on Schuetz et al. [4]

those of other airspace users [6]. Nevertheless, airspace users aspire to prioritize individual flights by swapping assigned TTAs with other airspace users.

The SlotMachine [7] project proposed a privacy-preserving platform for slot swapping that allows airspace users to collaboratively optimize flight lists in case of ATFM regulation while protecting the confidentiality of the submitted preferences [3]. The preferences for the optimization are confidential information that airspace users want to protect from competitors but potentially also from an honest-but-curious platform provider. While an optimized flight list may result in inequity between airspace users in the short run, an optimization platform must ensure equity in the long run.

In the SlotMachine [7] project, different market mechanisms served to achieve long-term equity. In a market mechanism based on a real currency [8], for example, the airspace users who receive favorable slots would pay money to the airspace users who give up these slots. In another market mechanism [9], delay credits were introduced which airspace users receive for giving up favorable slots and which they can use to prioritize important flights in the future. In practice, market mechanisms may be impractical due to the lack of acceptance by airspace users or because the precise value of slots in terms of real-world currency or delay credits is difficult to determine.

The HARMONIC [1] project is a continuation of the SlotMachine [7] project, which additionally takes into account the airport's preferences for optimization. Instead of a market mechanism, an equity mechanism should guarantee that, over a given time, no airspace user is at an advantage or disadvantage compared to the others. A Heuristic Optimizer is currently implemented and will be evaluated as part of a TTMS at Zurich Airport, where the airport is the trusted platform provider, and in a test environment with privacy-preserving optimization, assuming an honest-but-curious platform provider.

B. Privacy-Preserving Optimization

Collaborative optimization based on sensitive information requires either trust between the participants or methods for privacy-preserving computation that protect the sensitive information from unauthorized access. Various methods for privacy-preserving computation have been proposed, such as secure multi-party computation (MPC) and homomorphic encryption. Cramer et al. [2] provides a comprehensive introduction to privacy-preserving computation.

In time-critical settings, solutions must be found within the time constraints of the problem context. Optimizing the slot assignment during ATFM regulation, for example, must be finished within minutes [3]. In addition, the solution of the optimization should be better than the solution of the CASA algorithm. However, methods for privacy-preserving computation cause considerable additional runtime overhead, which makes it difficult for privacy-preserving optimization to find solutions in the time available for optimization.

Deterministic algorithms always find an optimal solution for an optimization problem. Considering flight prioritization as an assignment problem [10], the *Hungarian method* [11], for example, always finds an optimal solution. Even with multiple objectives, the optimization problem can be converted into a single-objective optimization problem by weighting and combining the objectives. Loruenser et al. [12] investigated the runtime of privacy-preserving implementations of deterministic algorithms based on MPC to solve single-objective assignment problems. However, the runtime may not be sufficient for flight prioritization, i.e., using these implementations may lead to situations where a solution is not found in time.

In contrast to deterministic algorithms, evolutionary algorithms can be terminated at any point in time and still provide solutions. Various works address privacy-preserving optimization using evolutionary algorithms. Zhao et al. [13], for example, proposed a privacy-preserving implementation of the genetic algorithm following the concept of evolutionary computation as a service. Privacy-preserving implementations of evolutionary algorithms always find solutions in the time available. Nevertheless, the solutions may be of insufficient quality because the privacy-preserving computation of the entire algorithm slows down the runtime behavior and thus reduces the number of possible generations.

Schuetz et al. [3] proposed to separate the search for solutions from the evaluation of the solutions for privacy-preserving optimization in time-critical settings. Therefore, methods for privacy-preserving computation are only required to evaluate the solutions, which reduces the impact on runtime performance. This approach was developed and evaluated as part of the SlotMachine [7] project, and indicated promising results [3], [10]. We extend this approach for the multi-objective flight prioritization problem to design a Heuristic Optimizer for a TTMS.

III. FLIGHT PRIORITIZATION

This section discusses the flight prioritization of airspace users and the encoding of preferences for a heuristic optimizer

based on the work of Schuetz et al. [4], and extends flight prioritization to include airport operations.

A. Flight Prioritization of Airspace Users

The optimal assignment for airspace users minimizes the overall delay costs based on the individual cost functions, as shown for example in Fig. 2. The optimization therefore requires these cost functions as input from the airspace users. For airspace users, it is however difficult to determine the individual cost functions due to the operational complexity.

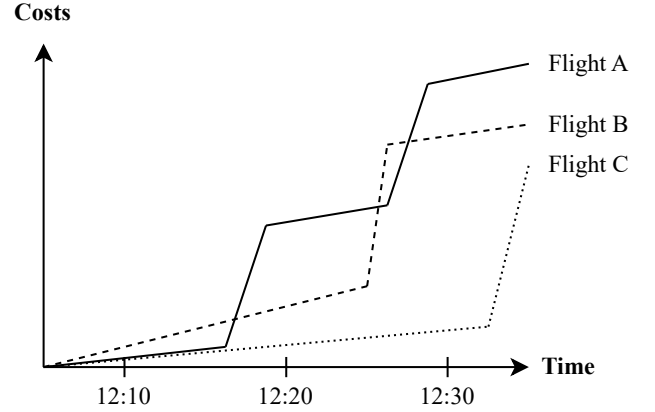


Fig. 2. Example of the cost functions of Flight A, Flight B and Flight C. The flight sequence (Flight C, Flight B, Flight A) would be inefficient in terms of costs. The optimized flight sequence (Flight A, Flight B, Flight C) improves cost-effectiveness.

Time triplets are one option for airspace users to encode their preferences for optimization. The time triplet of a flight consists of (TimeNotBefore, TimeWished, TimeNotAfter). TimeWished refers to the optimal arrival time of a flight, while the flight should not arrive before TimeNotBefore, and should not arrive after TimeNotAfter. The TimeNotBefore and TimeNotAfter are acceptable margins around the TimeWished to provide some flexibility for optimization. In practice, TimeNotBefore and TimeWished may be close to each other. The TimeNotAfter may be used to indicate the last arrival time within a major delay target of a flight.

Time triplets are combined with a priority based on a ranking of flights, which indicates the relative importance of a flight for the airspace user. Therefore, an airspace user ranks its flights, and the ranking is directly translated into the priority. The priority starts with a value of 1 for the most important flight, i.e., rank 1, and decreases as the value increases. Fig. 3 shows an example of time triplets together with a priority for two flights. If two or more flights are equally important for an airspace user, they receive the average of this rank and the corresponding other ranks as priority. For example, Flight X is rank 1, and Flight Y, Flight A, and Flight B are rank 2. In this case, Flight Y, Flight A, and Flight B receive the average of rank 2 and the subsequent ranks 3 and 4, i.e., $(2 + 3 + 4) \div 3 = 3$. The next most important flight, e.g.,

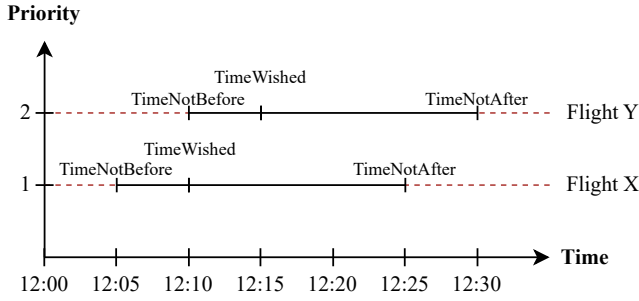


Fig. 3. Example of time triplets for Flight X and Flight Y of an airspace user. The red dotted lines indicate possible but unfavorable assignments for the airspace user. The time triplet of Flight X is (12:05, 12:10, 12:25). The optimal arrival time for Flight A is therefore 12:10 but the flight may also arrive between 12:05 and 12:25. The time triplet of Flight Y is (12:10, 12:15, 12:30). The optimal arrival time for Flight Y is therefore 12:15 but the flight may also arrive between 12:10 and 12:30. For the airspace user, Flight X is more important than Flight Y.

Flight C, then continues with rank 5, because ranks 3 and 4 have already been assigned.

The other option for airspace users to encode their preferences for optimization is weight maps. A weight map is a list of key-value pairs with the available arrival times as the keys and the weights as the values indicating the importance of a particular arrival time for a particular flight. The higher the numerical value, the more important the arrival time for this flight. Table I provides weight maps for two flights based on the time triplets from Fig. 3. Weight maps allow a much more fine-granular representation of preferences than the time triplets but at the same time are more difficult to derive and compare between airspace users.

TABLE I
EXAMPLE OF WEIGHT MAPS FOR FLIGHT X AND FLIGHT Y FROM FIG. 3

	12:00	12:05	12:10	12:15	12:20	12:25	12:30
Flight X	0	90	100	85	80	75	0
Flight Y	0	0	60	100	75	60	45

When allowing airspace users to provide their preferences as weight maps, a TTMS shall ensure that airspace users do not cheat, that the weight maps are comparable to the preferences encoded as time triples and that the weight maps of an airspace user are comparable to the weight maps of other airspace users. For example, if an airspace user specifies very high weights for only a few target times in a weight map, they are more likely to receive one of those target times, giving them an unfair advantage over others who specify more optimization-friendly preferences. To prevent such unfavorable behavior, a TTMS needs to establish rules for constructing weight maps, such as a minimum and maximum value for the weights together with limiting the amount of weights an airspace user can spend on one weight map or all of its weight maps. We are currently investigating different rules for constructing weight maps.

The TTMS uses a weight matrix for optimization, since the evaluation of solutions with MPC for privacy-preserving

optimization works well if only the values of the weight matrix have to be summed up. Therefore, the TTMS must be able to map the preferences provided as time triplets into weight maps that are comparable to the preferences provided as weight maps. Various mapping approaches are possible for this purpose. For example, the target time that corresponds to the TimeWished receives the highest value, while the weights decrease linearly to the target times that correspond to the TimeNotBefore and TimeNotAfter. We are currently investigating different mapping approaches.

B. Multi-Objective Flight Prioritization

The optimal assignment for airport operations improves several key performance indicators (KPIs), such as minimizing the average passenger delay. The optimization therefore requires the KPIs and the information necessary for calculating the KPIs as input from the airport. In contrast to the airspace user preferences, the KPIs and the information required for calculation may not be sensitive. The KPIs may however conflict with the airspace user preferences.

Optimization becomes increasingly difficult as the number of objectives increases. The following approaches to multi-objective flight prioritization are therefore conceivable. The first option is to weigh and combine the preferences of airspace users and the airport's KPIs into a weighted sum. However, this option comes with limitations, such as the difficulty of choosing the right weights, the loss of flexibility to adjust the weights based on obtainable solutions, and the fundamental problem of aggregating preferences [14]. The second option is to consider the preferences of airspace users as one objective and each individual airport KPI as an additional objective. However, evolutionary algorithms have difficulty approaching the Pareto frontier when there are more than three objectives [14].

Our approach to multi-objective flight prioritization is to consider the preferences of the airspace users as one objective, and a combination of the KPIs of the airport as an additional objective. The airport objective can, for example, be computed from a situation-dependent weighted sum of the KPIs, and translated either into time triplets or weight maps. The prioritization of flights is therefore a bi-objective optimization problem of the preferences of airspace users and the airport. This enables the use of efficient evolutionary algorithms that can approximate near-optimal solutions that reconcile both objectives.

Participation in flight prioritization remains voluntary for airspace users. In the SlotMachine [7] project, flights of airspace users who did not want to participate in the collaborative optimization were excluded from optimization. In the HARMONIC [1] project, however, the airport can prioritize flights of airspace users who have not provided preferences for the optimization. Therefore, all regulated flights must participate in the optimization, regardless of whether an airspace user has indicated preferences or not. In this case, a TTMS must set default preferences for flights of airspace users who

have not indicated preferences. We are currently investigating different settings for default preferences.

C. Long-Term Equity

Another objective of the optimization platform is to achieve long-term equity between airspace users, i.e., no airspace user (AU) is considerably disadvantaged or advantaged in the long run. In UDPP, equity meant that “the actions of one AU must not impact another’s flights” [15], which led to an implementation where one AU could prioritize its own flights but no slot swaps between airlines were allowed. Inter-airline optimization promises higher efficiency but may lead to one AU being considerably advantaged or disadvantaged over the long run. The proposed TTMS must, therefore, also consider equity between airlines over the long run, and introduce an *equity monitor* and possibly an *equity mechanism* that intervenes in the optimization in case of perceived inequity between airlines. The equity monitor and mechanism is not the focus of this paper. We refer to Carré [16] for a comprehensive investigation of this topic, the results of which may serve as the basis for the development of an equity monitor and mechanism for the proposed TTMS.

IV. TRUSTED PLATFORM PROVIDER

The TTMS will enable airspace users and the airport to collaboratively optimize flight lists in the case of ATFM regulations. Fig. 4 shows the proposed architecture of a TTMS considering a trusted platform provider and Fig. 5 illustrates the whole optimization process.

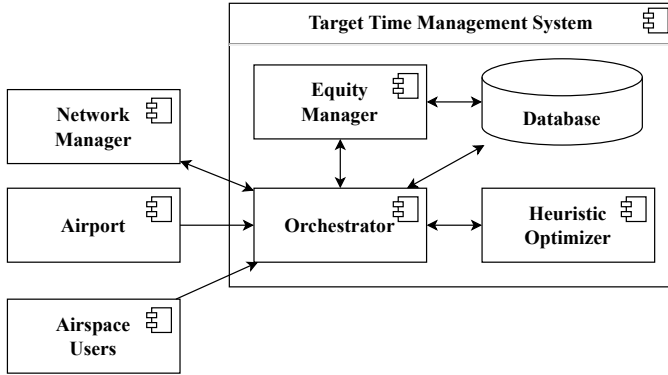


Fig. 4. The architecture of a Target Time Management System (TTMS) for a trusted platform provider.

During continuous operations, the Orchestrator component of a TTMS continuously receives the current preferences of the airport and airspace users, and stores the preferences in the database of the TTMS. The Orchestrator component also continuously requests active regulations from the EUROCONTROL Network Manager and the Network Manager responds with an ATFCM (Air Traffic Flow and Capacity Management) measures list. In case the measures list is relevant for a particular airport, it starts an optimization of slot assignments.

For the optimization, the Orchestrator component requests all necessary information related to the regulation from the

EUROCONTROL Network Manager, such as the list of regulated flights and the flight list proposed by the CASA algorithm. The Orchestrator component selects the preferences of the regulated flights from the database, converts the preferences into weight maps if necessary, and requests equity information from the Equity Manager component. The Equity Manager component computes, for example, inequity weights, based on historical data from previous regulations stored in the database. The Orchestrator component uses the received equity information to modify the weight maps, e.g., by adding the inequity weights to the weights maps. Once all the information for the optimization has been collected, the Orchestrator component calls the Heuristic Optimizer component to perform an optimization. The Heuristic Optimizer component performs an optimization using the weight maps received from the Orchestrator component and returns optimized flight lists. The Orchestrator component stores the optimization result in the database of the TTMS, selects one of the optimized flight lists by, e.g., using an approach for selecting the best compromise solution [17], evaluates the selected solution against internal KPIs, and sends a request with the optimized TTA assignments to the EUROCONTROL Network Manager in case the optimized solution surpasses the predefined KPIs. The Orchestrator component then stores the implemented TTA updates in the database in order to enable the Equity Manager component to take the implemented arrival flight list into account for future optimizations.

The Heuristic Optimizer component uses either a deterministic or an evolutionary algorithm for optimization. In the case of a trusted platform provider, the TTMS receives the preferences of airspace users as plain text, eliminating the need for privacy-preserving computation. Thus, a deterministic algorithm may be efficient enough to provide solutions in time.

If a deterministic algorithm is sufficient for the time constraints, the Heuristic Optimizer component normalizes and combines the weight maps of the airspace users and the airport to obtain a single weight map for each flight. The weight maps together form a weight matrix for which a deterministic algorithm such as the *Hungarian method* [11] finds an optimal solution. The Heuristic Optimizer may use different weightings between the objective of the airspace users and the objective of the airport to obtain different weight maps and thus a set of Pareto-optimal solutions.

In contrast to deterministic algorithms, evolutionary algorithms cannot guarantee optimal solutions, but they can remove the above-mentioned difficulty of choosing the correct weights and provide the flexibility to adjust the weights based on the solutions obtained, c.f. [14]. In an iterative process, an evolutionary algorithm such as a genetic algorithm searches for solutions, and the solutions found are evaluated based on the weight maps, assigning each solution with two fitness values. Well-known genetic algorithms for multi-objective optimization are, for example, SPEA-2 [18] and NSGA-II [19]. The choice of optimization algorithm ultimately depends on various factors, such as the time available for optimization and the relationship between airspace user and airport preferences.

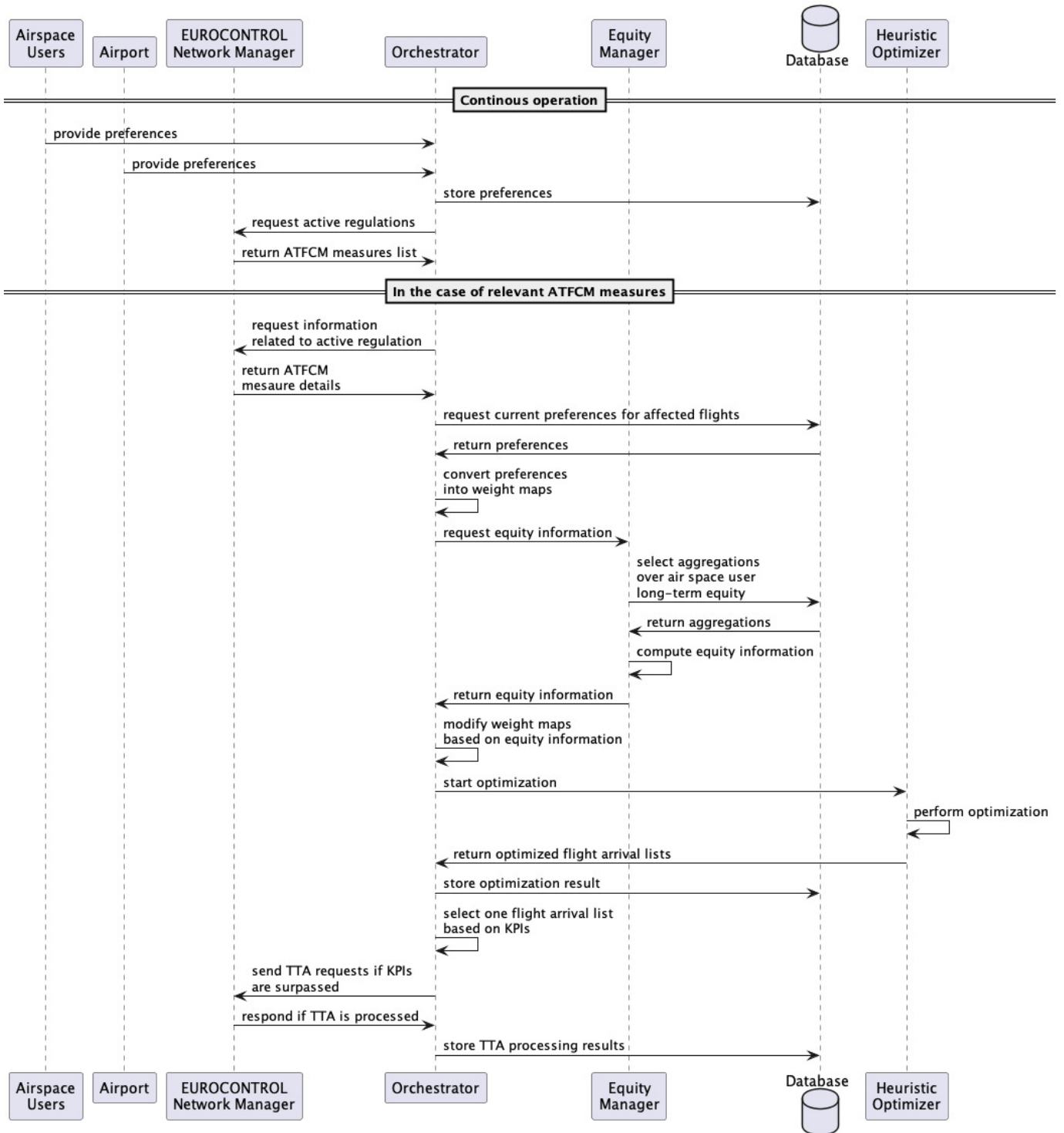


Fig. 5. The UML sequence diagram of a Target Time Management System (TTMS) for a trusted platform provider.

V. HONEST-BUT-CURIOUS PLATFORM PROVIDER

The TTMS will enable airspace users and the airport to collaboratively optimize arrival flight lists even under the assumption of an honest-but-curious platform provider. Fig. 6 shows the architecture of a TTMS for an honest-but-curious platform provider. In contrast to the architecture for a trusted platform provider, this architecture includes a Privacy Engine component for privacy-preserving computations. Fig. 7 illustrates the optimization involving a Privacy Engine component.

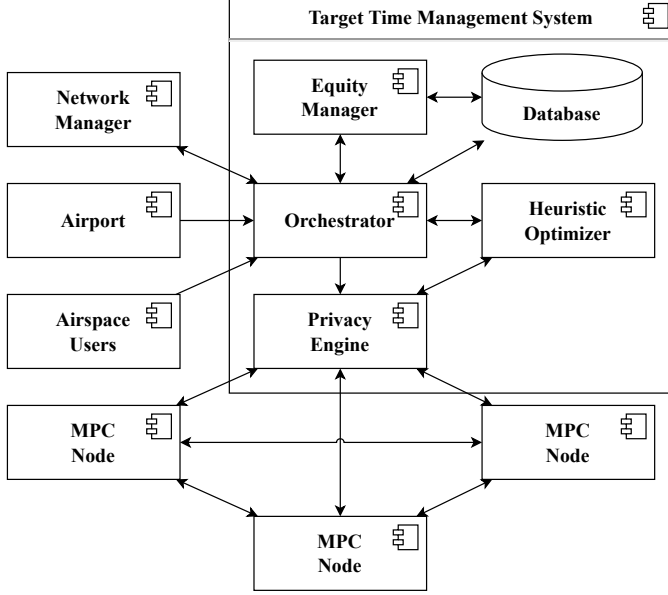


Fig. 6. The architecture of a Target Time Management System (TTMS) for an honest-but-curious platform provider with a Privacy Engine component using multi-party computation (MPC) for privacy-preserving optimization.

In the case of an honest-but-curious platform provider, the preferences of airspace users are only available in secret shares of the weight maps. In contrast to optimization in the TTMS for a trusted platform provider, the Orchestrator component needs to first initialize the Privacy Engine component with secret shares of the weight maps. The Privacy Engine component distributes the secret shares to the MPC nodes hosted by different parties. For optimization, the Heuristic Optimizer component uses an evolutionary algorithm to search for solutions and provides populations of solutions to the Privacy Engine component for the evaluation. The Privacy Engine component uses the MPC nodes to evaluate a population of solutions and obtains an obfuscated evaluation result that is used by the evolutionary algorithm to create the next population of solutions. The optimization terminates when the time available for optimization concludes, and the best solutions found so far are returned to the Orchestrator component.

VI. EXPERIMENTS

This section describes the setup, the metrics, and the results of our preliminary experiments using a Heuristic Optimizer for privacy-preserving optimization.

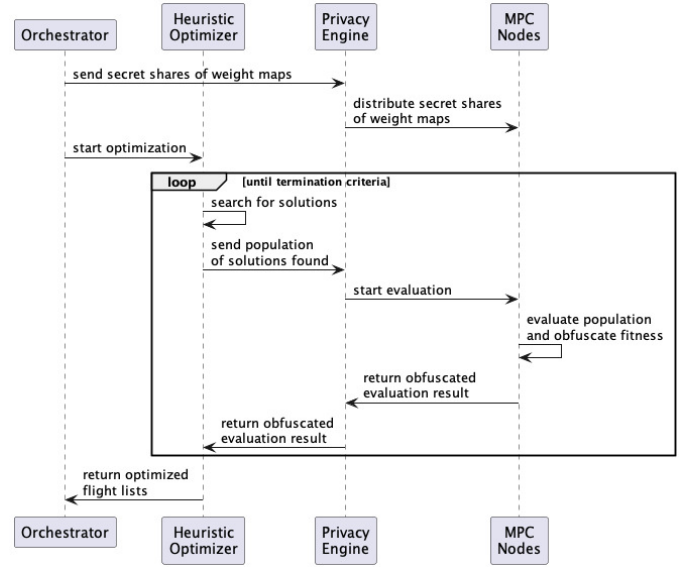


Fig. 7. The UML sequence diagram of optimization in a Target Time Management System (TTMS) for an honest-but-curious platform provider.

A. Setup

We generated 20 datasets, each with 100 flights and target times, and each flight was assigned a weight map for the preferences of the airspace user and a weight map for the preferences of the airport. The preferences of airspace users were generated based on the statistical distribution of an airline's preferences from a previous project [20]. For the preferences of the airport, we assumed that the airport prioritizes 20% of flights with a high priority, 40% of flights with a medium priority and 40% are not prioritized at all.

For conducting experiments, we developed a prototype based on Jenetics [21], which is a Java library for genetic algorithms. For the sake of simplicity, the prototype only simulates the Privacy Engine. More precisely, the evaluation results are obfuscated but the evaluation does not use an MPC protocol. While the results are the same as if a Privacy Engine had been used, it allows us to carry out the experiments faster and without dealing with the technological complexity of MPC for the performance experiments.

We assumed an honest-but-curious platform provider and assumed that only the preferences of airspace users are sensitive information and need to be obfuscated before the evaluation results are revealed to the Privacy Engine. We used the obfuscation methods order, order quantiles, and fitness range quantiles because they performed well in previous work. We refer to Feichtenschlager et al. [20] for a more detailed discussion on the obfuscation process and methods.

B. Metrics

We measure the quality of the optimized solutions by the *performance* and *imbalance*. Both metrics require, however, true Pareto-optimal solutions as reference points. Therefore, we converted the multi-objective optimization problem into a single-objective optimization problem following a *weighted*

sum approach [22], i.e., $f(x) = \omega f_1(x) + (1 - \omega)f_2(x)$ with $\omega \in [0, 1]$. We executed the *Hungarian method* [11] 11 times to solve the single-objective optimization problem, starting with $\omega = 0$ and increasing $\omega = \omega + .1$ after each execution to obtain a subset P_{known} of true Pareto-optimal solutions with a size of $2 \leq |P_{known}| \leq 11$. This approach is able to identify true Pareto-optimal solutions on the vertices of the convex hull of the Pareto frontier [23].

The performance of an optimized solution x_{opt} is measured by comparing its fitness with the fitness of a reference solution x_{ref} for each objective $i \in [1, 2]$:

$$performance_i(x_{opt}) = \frac{f_i(x_{opt})}{f_i(x_{ref})} \quad (1)$$

For the reference solution x_{ref} , the closest solution that dominates x_{opt} is selected based on the Manhattan distance to solutions in P_{known} . However, P_{known} only represents a subset of the true Pareto-optimal solutions. For some x_{opt} , thus, no solution of P_{known} may dominate x_{opt} which would result in $performance_i(x_{opt}) > 100\%$. In such cases, the performance of x_{opt} for both objectives is 100%.

Some optimized solutions may have high performance but are not well balanced between the objectives of the airspace users and the airport. For example, solution x_1 in Fig. 8 has a performance of $3.5 \div 4 = 87.5\%$ for the objective of the airspace users and $8 \div 8.5 = 94.11\%$ for the objective of the airport considering x_{ref1} as the reference point. The solution x_2 , on the other hand, has a performance of $6 \div 7 = 85.71\%$ for each of both objectives considering x_{ref2} as the reference point. Although x_2 has a lower performance than x_1 for both objectives, x_2 may be the better solution as it seems to be more balanced between the two objectives. Therefore, we additionally introduce an imbalance ratio to assess the quality of optimized solutions.

We measure the imbalance ratio of an optimized solution x_{opt} by comparing its fitness with the maximum fitness for an objective based on solutions of P_{known} and comparing the result with the result for the other objective:

$$imbalance(x_{opt}) = \left| \frac{f_1(x_{opt})}{f_1(x_{max_{f_1}})} - \frac{f_2(x_{opt})}{f_2(x_{max_{f_2}})} \right| \quad (2)$$

The imbalance ratio of x_{opt} is between 0 and 1. The closer the imbalance ratio is to zero, the more balanced the solution is between the two objectives; the closer it is to 1, the more unbalanced the solution. In Fig. 8, for example, the maximum fitness for the objective of the airspace users is $f_1(x_{f1}) = 8$, while the maximum fitness for the objective of the airport is $f_2(x_{f2}) = 9$. Therefore, solution x_1 has an imbalance ratio of $|(3.5 \div 8) - (8 \div 9)| = .45$, while solution x_2 has an imbalance ratio of $|(6 \div 8) - (6 \div 9)| = .08$. Thus, x_2 is a more balanced solution than x_1 .

C. Results

We executed a genetic algorithm configured with SPEA-2 [18], NSGA-II [19] and tournament selection adapted for

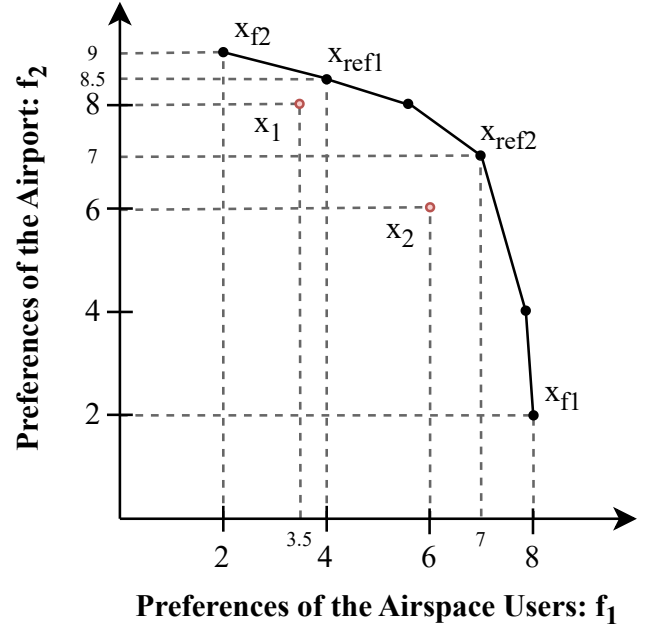


Fig. 8. Example for measuring the performance and imbalance ratio of x_1 and x_2 . The filled points that are connected by a line represent the subset of true Pareto-optimal solutions. The performance of x_1 is 87.5% for the preferences of the airspace users and 94.11% for the preferences of the airport. The performance of x_2 is 85.71% for each objective. The imbalance ratio of x_1 is .45, while the imbalance ratio of x_2 is .08

multi-objective optimization 100 times on each dataset. Tables II–IV show the results of the optimization when only the airspace user preferences are obfuscated, leaving the airport preferences public. For computing the metrics, we simulated a human decision-maker with an approach based on Abido and Bakhshwain [17] to choose the best compromise solution from the set of Pareto-optimal solutions found. The results are quite balanced between the objectives but indicate a small bias towards the airport preferences. Results are promising for future work, indicating generally good performance of the optimization algorithm.

VII. SUMMARY AND FUTURE WORK

In this paper, we extended the single-objective problem of flight prioritization in ATFM to a multi-objective problem that includes airport operations. We addressed the need for long-term equity for a TTMS and presented the architecture of a TTMS for a trusted platform provider and extended this architecture for an honest-but-curious platform provider for privacy-preserving optimization. Results from the Slot-Machine [7] project and preliminary results of the ongoing development of an extension to multi-objective optimization in the HARMONIC [1] project indicate that good results can be achieved, both under the assumption of a trusted platform provider as well as an honest-but-curious platform provider.

Future work will further investigate the performance of the optimization algorithm with real-world data. Furthermore,

TABLE II
TOURNAMENT OPTIMIZATION WITH DIFFERENT OBFUSCATION METHODS

Obfuscation	Airline Preferences		Airport Preferences		Imbalance Ratio	
	Mean	SD	Mean	SD	Mean	SD
—	.9681	.0304	.9715	.0291	.0495	.0369
Fitness Range Quantiles with 5 Quantiles	.7544	.0794	.9813	.0130	.5998	.0366
Fitness Range Quantiles with 7 Quantiles	.8674	.0600	.9798	.0117	.5341	.0376
Fitness Range Quantiles with 10 Quantiles	.9355	.0453	.9797	.0124	.4803	.0440
Order Quantiles with 5 Quantiles	.9735	.0482	.9852	.0185	.3219	.0582
Order Quantiles with 7 Quantiles	.9819	.0283	.9886	.0179	.2175	.0465
Order Quantiles with 10 Quantiles	.9844	.0257	.9870	.0217	.1147	.0474
Order	.9842	.0248	.9842	.0260	.0916	.0472

TABLE III
SPEA-2 OPTIMIZATION WITH DIFFERENT OBFUSCATION METHODS

Obfuscation	Airline Preferences		Airport Preferences		Imbalance Ratio	
	Mean	SD	Mean	SD	Mean	SD
—	.9818	.0257	.9857	.0204	.0726	.0621
Fitness Range Quantiles with 5 Quantiles	.8787	.0715	.9244	.0326	.4445	.0661
Fitness Range Quantiles with 7 Quantiles	.9138	.0588	.9377	.0338	.4097	.0541
Fitness Range Quantiles with 10 Quantiles	.9305	.0506	.9544	.0288	.3346	.0534
Order Quantiles with 5 Quantiles	.9452	.0751	.9744	.0209	.3636	.0653
Order Quantiles with 7 Quantiles	.9676	.0391	.9749	.0265	.3104	.0563
Order Quantiles with 10 Quantiles	.9684	.0368	.9828	.0218	.2721	.0543
Order	.9338	.0386	.9229	.0440	.1285	.0589

TABLE IV
NSGA-II OPTIMIZATION WITH DIFFERENT OBFUSCATION METHODS

Obfuscation	Airline Preferences		Airport Preferences		Imbalance Ratio	
	Mean	SD	Mean	SD	Mean	SD
—	.9471	.0410	.9434	.0423	.1541	.1016
Fitness Range Quantiles with 5 Quantiles	.7397	.0738	.9586	.0187	.5826	.0314
Fitness Range Quantiles with 7 Quantiles	.8436	.0641	.9550	.0189	.5192	.0445
Fitness Range Quantiles with 10 Quantiles	.9243	.0444	.9571	.0208	.4619	.0431
Order Quantiles with 5 Quantiles	.9411	.0887	.9736	.0276	.2815	.1059
Order Quantiles with 7 Quantiles	.9435	.0884	.9662	.0302	.2104	.1054
Order Quantiles with 10 Quantiles	.9599	.0413	.9668	.0306	.1226	.0724
Order	.9585	.0298	.9391	.0450	.1910	.0603

the developed system will be validated in a representative environment as part of the HARMONIC validation exercises. We will also investigate the issue of achieving long-term equity between airspace users.

ACKNOWLEDGMENT

This work was conducted as part of the HARMONIC project. This project has received funding from the SESAR Joint Undertaking under grant agreement No 101114675 under the European Union's Horizon Europe research and innovation program. UK participant NATS in Horizon Europe Project HARMONIC receives funding from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10091990]. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI). The views expressed in this paper are those of the authors.



Co-funded by
the European Union



Project funded by



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs,
Education and Research EAER
State Secretariat for Education,
Research and Innovation SERI

REFERENCES

- [1] European Union, "HARMONISED network through smart technology and Collaboration," <https://doi.org/10.3030/101114675>, 2024.
- [2] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.
- [3] C. G. Schuetz, T. Lorünser, S. Jaburek, K. Schuetz, F. Wohner, R. Karl, and E. Gringinger, "A distributed architecture for privacy-preserving optimization using genetic algorithms and multi-party computation," in *Cooperative Information Systems*, M. Sellami, P. Ceravolo, H. A. Reijers, W. Gaaloul, and H. Panetier, Eds. Cham: Springer International Publishing, 2022, pp. 168–185.

- [4] C. G. Schuetz, E. Gringinger, N. Pilon, and T. Lorünser, "A privacy-preserving marketplace for air traffic flow management slot configuration," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021, pp. 1–9.
- [5] A. Cook, "Improved flexibility and equity for airspace users during demand-capacity imbalance-an introduction to the user-driven prioritisation process," *Sixth SESAR Innovation Days*, 2016.
- [6] S. Ruiz, L. Guichard, N. Pilon, K. Delcourte *et al.*, "A new air traffic flow management user-driven prioritisation process for low volume operator in constraint: Simulations and results," *Journal of Advanced Transportation*, 2019.
- [7] European Union, "A Privacy-Preserving Marketplace for Slot Management," <https://doi.org/10.3030/890456>, 2024.
- [8] E. Gringinger, S. Ruiz, and C. G. Schuetz, "Business and economic concepts for a privacy-preserving marketplace for ATFM slots," in *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, 2022, pp. 1–11.
- [9] C. Schütz, S. Ruiz, E. Gringinger, C. Fabianek, and T. Lorünser, "An auction-based mechanism for a privacy-preserving marketplace for atfm slots," in *33rd Congress of the International Council of the Aeronautical Sciences*, 2022, pp. 1–14, iCAS Conference 2022 ; Conference date: 04-09-2022 Through 09-09-2022.
- [10] K. Schuetz, C. G. Schuetz, and S. Jaburek, "Privacy-preserving implementation of local search algorithms for collaboratively solving assignment problems in time-critical contexts," in *2023 IEEE Congress on Evolutionary Computation (CEC)*, 2023, pp. 1–10.
- [11] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [12] T. Loruenser, F. Wohner, and S. Krenn, "A verifiable multiparty computation solver for the linear assignment problem: And applications to air traffic management," in *Proceedings of the 2022 on Cloud Computing Security Workshop*, ser. CCSW'22. New York, NY, USA: Association for Computing Machinery, 2022, p. 41–51.
- [13] B. Zhao, W.-N. Chen, F.-F. Wei, X. Liu, Q. Pei, and J. Zhang, "Pega: A privacy-preserving genetic algorithm for combinatorial optimization," *IEEE Transactions on Cybernetics*, pp. 1–14, 2024.
- [14] R. Kruse, S. Mostaghim, C. Borgelt, C. Braune, and M. Steinbrecher, *Fundamental Evolutionary Algorithms*. Cham: Springer International Publishing, 2022, pp. 287–341.
- [15] N. Pilon, A. Cook, S. Ruiz, A. Bujor, and L. Castelli, "Improved flexibility and equity for airspace users during demand-capacity imbalance-an introduction to the user-driven prioritisation process," in *Proceedings of the 6th SESAR Innovation Days*, 2016.
- [16] M. Carré, "Multi-airline operations optimization under major disruptions," Ph.D. dissertation, Université Clermont Auvergne, 2024. [Online]. Available: <https://theses.hal.science/tel-04547756>
- [17] M. Abido and J. Bakhshwain, "Optimal var dispatch using a multiobjective evolutionary algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 27, no. 1, pp. 13–20, 2005.
- [18] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK report*, vol. 103, 2001.
- [19] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii," in *Parallel Problem Solving from Nature PPSN VI*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 849–858.
- [20] P. Feichtenschlager, K. Schuetz, S. Jaburek, C. G. Schuetz, and E. Gringinger, "Privacy-preserving implementation of an auction mechanism for atfm slot swapping," in *2023 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, 2023, pp. 1–12.
- [21] F. Wilhelmstötter, "Jenetics," URL: <http://jenetics.io>, 2024.
- [22] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006, special Issue - Genetic Algorithms and Reliability.
- [23] A. Przybylski, X. Gandibleux, and M. Ehrgott, "Two phase algorithms for the bi-objective assignment problem," *European Journal of Operational Research*, vol. 185, no. 2, pp. 509–533, 2008.