IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 6, 2020, accepted August 29, 2020, date of publication September 2, 2020, date of current version September 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021192

Simulation-Based Evolutionary Optimization of Air Traffic Management

ALESSANDRO PELLEGRINI[1,2], (Member, IEEE), PIERANGELO DI SANZO[1,2], BEATRICE BEVILACQUA[2], GABRIELLA DUCA[2], DOMENICO PASCARELLA[3], ROBERTO PALUMBO[3], JUAN JOSÈ RAMOS[4], (Member, IEEE), MIQUEL ÀNGEL PIERA[4], (Member, IEEE) AND GABRIELLA GIGANTE[3]

[1]Lockless S.r.l., 00137 Rome, Italy
[2]Instititute for Sustainable Society and Innovation (ISSNOVA), 80136 Naples, Italy
[3]Centro Italiano Ricerche Aerospaziali (CIRA), 81043 Capua, Italy
[4]Department of Telecommunication and Systems Engineering, Universitat Autònoma de Barcelona (UAB), 08193 Barcelona, Spain

Corresponding author: Alessandro Pellegrini (pellegrini@lockless.it)

This work was supported by the Single European Sky ATM Research (SESAR) Joint Undertaking (Evolutionary Air Traffic Management (EvoATM) Project) through European Union's Horizon 2020 Research and Innovation Program under Grant 783189.

ABSTRACT In the context of aerospace engineering, the optimization of processes may often require to solve multi-objective optimization problems, including mixed variables, multi-modal and non-differentiable quantities, possibly involving highly-expensive objective function evaluations. In Air Traffic Management (ATM), the optimization of procedures and protocols becomes even more complicated, due to the involvement of human controllers, which act as final decision points in the control chain. In this article, we propose the use of computational intelligence techniques, such as Agent-Based Modelling and Simulation (ABMS) and Evolutionary Computing (EC), to design a simulation-based distributed architecture to optimize control plans and procedures in the context of ATM. We rely on Agent-Based fast-time simulations to carry out offline what-if analysis of multiple scenarios, also taking into account human-related decisions, during the strategic or pre-tactical phases. The scenarios are constructed using real-world traffic data traces, while multiple optimization variables governed by an EC algorithm allow to explore the search space to identify the best solutions. Our optimization approach relies on ad-hoc multi-objective performance metrics which allow to assess the goodness of the control of aircraft and air traffic regulations. We present experimental results which prove the viability of our approach, comparing them with real-world data traces, and proving their meaningfulness from an Air Traffic Control perspective.

INDEX TERMS Air traffic control, distributed optimization, evolutionary algorithms, modeling and simulation, multi-objective optimization, support to strategic design.

I. INTRODUCTION

Air Traffic Management (ATM) is a complex socio-technical system composed of multiple entities and physical/human actors. Its ultimate goal is to safely and efficiently manage the flow of aircraft in all flight phases (departure, en-route, landing). The complexity behind ATM lies in the fact that the number of systems, devices, and the amount of people involved in the process is large, as well as the number of aircraft which have to be simultaneously managed. Moreover, this number is subject to bursts in rush hours, thus making the overall process even more complicated. Finally, an additional level of complexity is related to the interoperability and the harmonization of different ATM systems, since various countries have developed their own ATM system individually, each one concentrating on its specific requirements.

Today, enhancing ATM is a relevant international objective, and it is fundamental to be ready to meet future traffic demand and environmental requirements. To overcome the complexity related to interoperability, there is an ongoing effort aimed at designing new services [1]. However, the high complexity of ATM makes enhancements extremely difficult, because performing in-vitro testing of new procedures or equipment is almost unfeasible. At the same time, the criticality of ATM makes it highly improbable that possible enhancements can be safely tested in real-world scenarios,

The associate editor coordinating the review of this manuscript and approving it for publication was Haluk Eren.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

also considering the safety-wise and performance-wise critical requirements of ATM services.

In this article, we propose an approach based on computational intelligence techniques which, combining Agent-Based Modelling and Simulation (ABMS) [2] and Evolutionary Computing (EC) [3], allows to carry out offline what-if analyses of changes to ATM policies or procedures, and supports the design of new solutions aimed at ATM system optimization. The purpose of these analyses is to explore new configurations for the ATM system in advance, namely during the strategic (i.e., several days before the actual operations) or pre-tactical phases (i.e., up to several hours before the actual operations). Such optimization is intended with respect to the estimated performance of the simulated ATM system and implies the minimization or maximization of some reference ATM performance metrics in the identified simulation scenario.

We designed a distributed simulation-based evolutionary optimization architecture which enables what-if analyses via an agent-based simulation model, and allows to evaluate objective measures (such as the timeliness of the flights in a set of sectors), as well as subjective measures related to human behavior—in particular the behavior of Air Traffic Controllers (ATCOs). This simulation model is placed at the center of an optimization loop, which allows the modification of input parameters to the simulation model so as to identify properly-tuned configurations which increase the global performance of the ATM system. Overall, our approach involves ABMS to discover the evolution of the ATM system under certain constraints, and EC to explore a complex search space to better understand how architectural and design choices influence the ATM system at micro-scale, i.e. at the level of individual agent behaviors. The versatility of our optimization architecture lies in the full decoupling between the exploration of the search space, the assessment of the goodness of a specific solution by means of ABMS, and the evaluation of performance metrics. The utilization of different simulation paradigms or the inclusion of additional performance metrics can be easily plugged in the proposed architecture.

The optimization process we designed is based on Evolutionary Algorithms (EAs). EAs are a category of metaheuristics-based algorithms. Inspired by concepts from nature, such as evolution and natural selection, they are particularly useful to provide solutions to computationally-intensive problems. They maintain a population of individuals which must compete for survival, and new offspring are created by recombining and mutating individuals selected from the population.

Placed in the context of ATM optimization, our proposal has a twofold goal. On the one hand, our architecture allows to perform an impact assessment of the changes applied to an ATM system. On the other hand, we support the design of new solutions, by identifying the optimal tuning of specific design parameters to achieve some required performance level. Generally speaking, according to our methodology, both approaches involve the following steps: i) modeling the part of ATM system of interest representing some reference scenario, i.e. the baseline which represents the basis for the change design; ii) modeling the scenario under the envisaged changes in the system; iii) identify the metrics of interest to study the impact of the changes; iv) simulating both the reference and the changed scenarios; v) assess the impact of the changes according to the selected metrics. Our architecture allows to effectively carry out all these steps, with minor human intervention, to study the behavior of an ATM system under differentiated characteristics.

We provide a detailed description of the overall simulation-based optimization architecture, along with an assessment comparing the performances achieved in two different scenarios, based on real-world traces. The two scenarios relate to "direct routing" (i.e., when an aircraft flies through the great circle arc [1] between 2 assigned waypoints) and "free routing" (i.e., when the pilot has the ability to plan/re-plan a route according to some defined preferences) operations of the ATM system. The selected case studies are highly representative of the ATM-domain complexity, and have been identified to stress-test our solution due to the inclusion of all elements of socio-technical systems. In detail, the transition from direct routing to free routing has already been tackled in Italy from 2013 to 2016, therefore it represents an ideal "oracle" with its known change history. Indeed, these scenarios have been used as a test oracle for our validation activities.

The remainder of this article is organized as follows. Section III presents the agent-based simulation model we use in our architecture. In Section IV, we discuss the evolutionary optimization algorithm which allows to explore the search space, seeking for a well-tuned configuration of the input parameters. Section V presents the distributed optimization architecture which we envisaged to cope with the computational demand of our optimization strategy. An experimental evaluation of our proposal, both from a performance and a correctness point of view, is presented in Section VI. Related work is discussed in Section II.

II. RELATED WORK

Agent-based Evolutionary Search (AES) is an emerging research paradigm lying at the intersection between EC and ABM. It is gaining attention [4] as a viable and innovative way to investigate complex adaptive systems thanks to a conceptual framework that allows to capture both the "simplexity" of ABM and the robustness and adaptability of EC methods.

In the context of ATM, some recent proposals in the literature have explored the possibility to rely on Evolutionary Computing to study or optimize some characteristics of ATM (see, e.g., [5]–[9]). This is an indication of the fact that this

[1] The shortest path between two points on the surface of a sphere is given by the arc of the great circle passing through the two points. A great circle is defined to be the intersection with a sphere of a plane containing the centre of the sphere.

kind of optimization strategy is hot. Nevertheless, no proposal has tried to consider all the aspects addressed by our work, nor ABM simulation-based optimization has ever been used for the purpose.

Modelling socio-technical systems has been largely studied in the last decade. Different and interconnected trends are notably to mention [10]: Actor Network Theory (ANT), Agent- Based Modelling technique (ABM), Bayesian Belief Network (BBN), Configuration Modelling (CM). Fuzzy Logic (FL), Morphological Analysis (MA), Social Network Analysis (SNA) and finally System Dynamics (SD). Such approaches have been applied to very different fields from project management to software engineering, from energy to aviation.

The choice of the most suitable formalism may depend on different aspects, the level of abstraction that it is expected to target, the capability to model and distinguish between human and non human elements, the capability to manage the interdependencies, the feedback and the uncertainties. In the last ten years, ABM has been used for modelling complex systems in different domains [11], including ATM.

There are several commercial tools to assess ATM concepts. Among them, we mention Simmod,[2] AirTOp,[3] CAST,[4] RAMSrams plus,[5] Total Airspace and Airport Modeler[6] (TAAM) or the AgentFly ATM simulation suite [12]. It is notable the effort put by NASA and FAA to develop FACET [13]. Most of these tools provide detailed models of airports and airspace for fast-time gate-to-gate simulation; very few use multi-agent architectures for different actors of the scene, e.g. for airport controllers. None embeds the human behavior modelling. In recent years, some attempts have been made within the SESAR Program. Many exploratory research projects (e.g., ELSA [14], ACCESS [15], TREE [16], CASSIOPEIA [17], MAREA [18], SPAD [19], EMERGIA [20]) have adopted the agent-based modeling paradigm to address a wide number of different ATM problems. In these projects, the agent-based simulation is used (usually in combination with other techniques) either as an analysis tool to understand emergent behaviors of the ATM system or to study resilience and disturbance propagation, or even as a tool to determine airport slot auctioning and allocation. Most of these studies address technical aspects in modeling, without considering the social aspects that influence the overall performance of the ATM system, and focus on specific cases instead of defining a methodological approach for modeling and simulation to assess the change impact.

A notable work [21] relies on ABMS as a method for capturing emergent behavior of the socio-technical air transportation system, so as to evaluate novel system designs. The author focuses on the identification of emergent safety risk of an active runway crossing operation, on the evaluation of the role of coordination in Airline Operations Control (AOC) resilience. Differently from our proposal, no utilization of EAs is proposed to rely on simulations so as to optimize ATM scenarios.

As concerns some technical choices which have driven the design of our evolutionary optimization architecture, Scott and De Jong [22] performed both theoretical and practical analysis of a simple asynchronous master-slave EA. They found that the amount of increased throughput that can be expected from an asynchronous EA depends on the number of slave processors, the size of the population and the variance of the evaluation time distribution. Furthermore, they investigated whether there is a bias towards fast-evaluating solutions in asynchronous EAs. In fact, when evaluation time is a heritable trait, many fast-evaluating individuals may be born, evaluated and compete for a place in the population in the time it takes for a single slower individual to be evaluated.

Similarly, Yagoubi *et al.* [23] observed evidence that asynchronous EAs had a harder time at finding good solutions when they were located in a region of the search space that was artificially configured to have slower evaluation times, and they similarly found that there is no independent selection pressure that favors fast-evaluating individuals. Indeed, they found that an asynchronous EA is able to find a global optimum on the Holder table function much more frequently than the generational EA does and that the asynchronous EA does not exhibit any particularly adverse performance. To the contrary, it does a good job of delivering the promised speedup in time-to-convergence and it has a much smaller risk of premature convergence. This does not, however, rule out an evaluation-time bias that can emerge from a combination of reproduction, variation and selection, so the results of [22] do not contradict the bias observed in practice by [23]. Overall, the studies in [22], [23], confirm several of the design choices which have driven the construction of our optimization architecture.

III. AGENT-BASED ATM SIMULATION

In this section, we present our agent-based simulation approach. We discuss the main challenges to face when designing ATM simulation systems and the related modeling problems, and we discuss the design choices that have been put in place, along with the related motivations. The overall organization of the Agent-based Model (ABM) based simulation system is depicted in Figure 1.

We implemented our ABM simulation system as a stochastic time-stepped Discrete Event Simulation (DES) model. This is a traditional paradigm typically used in the context of ABMS, but given the complex ATM scenarios which we target, great care has been put in the management, dispatching, and scheduling of the events, as well as into the ontologies formalizing the agent interactions.

The ABMS system is composed of different components. Agents implement the most important part of the logic in the simulation, and in a given simulation scenario there can be any required number of agents. At the same time, the system

[2] http://www.atac.com/simmod-pro.html, last accessed June 10, 2020.

[3] http://www.airtopsoft.com, last accessed June 10, 2020.

[4] http://www.airport-consultants.com, last accessed June 10, 2020.

[5] http://www.ramsplus.com, last accessed June 10, 2020.

[6] http://www.jeppesen.com/taam, last accessed June 10, 2020.

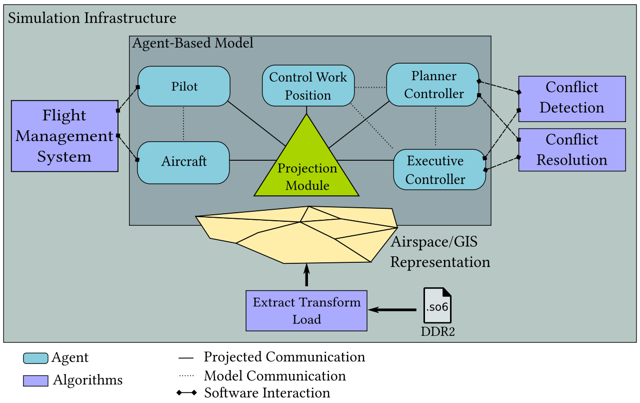


FIGURE 1. Architecture of the ABM Simulation System.

ships with some fundamental ATM-related algorithms, which allow to configure the overall simulation run, or which allow the different agents to carry out tasks which are related to the current dynamics of the simulation scenario. The agents interact with these algorithms by means of explicit function calls in the agents' implementation. The set of algorithms which have been implemented are:

- *Air space configuration*: this algorithm creates a representation of the structured air space (Functional Airspace Blocks, sectors, waypoints, etc) as to be used in the simulated scenario.
- *Extract Transform Load*: it allows to transform real traces associated with flown flights into the internal representation which is used to carry out the airspace simulation.
- *Conflict Detection*: this set of algorithms implement medium- and short-term conflict detection, which can be used by ATCO agents to trigger corrective actions with respect to the current state of the observed flights.
- *Conflict Resolution*: this set of algorithms allow ATCO agents to make decisions on the current state of the flights.
- *Flight Management System*: this is a set of algorithms used to provide the relevant flight information, including the flight intent (expressed as a 4D profile until destination), and allow to modify trajectories according to simulated ATCOs' directives.

The model is fed with an *air space configuration*, in the form of several sectors to which multiple agents are then mapped. These agents are in charge of monitoring and taking decisions with respect to the simulated air traffic—e.g., an ATCO planner and an ATCO executive. Input parameters to the model allow to fine tune the behavior of the agents, in the face of the different simulated conditions that can be encountered.

Traffic information is extracted from so6 files [24], a standard file format for ATM which stores a description of 4D-trajectories of flights in the European airspace. Our dataset is provided by Eurocontrol via their Demand Data Repository 2 (DDR2).[7] Similarly, a dedicated Extract-Transform-Load layer transforms the data contained in so6 files into an internal representation which maps the traffic information immediately to airspace sectors. Moreover, the agents involved in the simulation can, at any time, alter the trajectory of the flights, depending on the trajectory of the stochastic simulation.

There are agents of different nature available in the ABM system. As mentioned, any number of agents can be spawn in a simulation run, depending on the input configuration (in terms of flights and sectors). In particular, at the time of this writing, we have implemented the following agents:

- *Pilot*: this agent is responsible for flying an aircraft, and interacting with the ATCOs in order to obtain clearance to carry out several tasks.
- *Planner Controller*: this agent is the ATCO who is mainly responsible for the coordination of the traffic entering or exiting within the sector.
- *Executive Controller*: this agent is the ATCO who is responsible for the safe and expeditious flow of all flights operating within its sector. This agent monitors and separates flights that operate within its area of responsibility and, if necessary, it issues instructions to pilots for conflict resolution.
- *Aircraft*: this agent implements an aircraft. The reason why we have decided to implement the aircraft as agents is that different companies typically implement different flying strategies. Therefore, this level of abstraction allows us to better capture the intrinsic variability in the behavior of different flights, and enhances the capability of the system to observe emergent behaviors.
- *Controller Working Position*: this agent implements the controller workstation where traffic can be monitored allowing for situational awareness. We have decided to implement a technical part of the system as an agent to reach the goal of allowing to perform what-if analyses of changes in the overall organization of the ATM system. In this sense, mapping a technical system to an agent allows to experiment more easily with changes at the level of ATCOs operations.

As mentioned, one goal of our simulation-based architecture is to allow us to study the behavior of the ATM system including the behavior of the humans which take part of it [25]. Therefore, we have identified a set of human-behavior variables which are related to the ATCOs'tasks, also taking into account the organizational context and traffic situation. The ATCOs' behavior has been modeled [26] by characterizing the tasks assigned to each human actor according to both the human and context variables. This characterization of tasks includes a variety of factors, such as time sensitiveness, task complexity (number of alternative and optional actions in a task), behavioural level (skill/rule/knowledge based), resources allocation (human/computer/human with computer assistance).

The variability of pilots' behavior is represented with a higher-level of abstraction thanks to a characterization of airlines' standard interactions with ATCOs. One of the main

[7] https://www-test.eurocontrol.int/ddr.

TABLE 1. Human-factor characterization: Personal Factors.

Variable	Definition	Values or Resources involved	Dependencies
Age	Biological age.	Expressed in numbers or in range	—
Seniority	Length of continuous service in the same professional field/licences (it includes different roles). Timespan of the experience the actor has developed on the airspace (free-routing).	Expressed in Nr. of years or in a scale	—
Expertise	Level in which the person is knowledgeable. It refers to the range of sector and licences and different professional experience the actor has developed (e.g. pilot, military airspace, ...)	Basic/multiple	Positively correlated with Seniority
English	Level of knowledge and use of the English language	Mother tongue/non-master	It may be positively affected by Expertise
Role	Current professional role. It refers to goals, tasks and responsibilities.	• Planning ATCO • Executive ATCO • Other sector Controllers	May be related to Seniority. It can affect the Mental workload.

TABLE 2. Human-factor characterization: Cognitive Factors.

Variable	Definition	Values or Resources involved	Dependencies
Situation Awareness	Perception of elements in the environment, their interpretation and the projection of the future status [18]. It focuses on how operators develop and maintain a sufficient understanding of "what is going on" in order to achieve success in task performance. It results in problem detection.	• Visual perception • Selective attention • Information processing [18] • Projection	Both external conditions (traffic complexity and automated tools) and Cognitive ones (mental fatigue) may negatively affect the persistence and efficacy, in terms of boredom [28]. Moreover, the Expertise might influence the SA, particularly in terms of projection
Mental Workload	Subjective experience of the demand imposed by the ATC task [29].	• Visual perception • Selective attention • Information processing • Memory • Multitasking	It highly depends on the task nature. External conditions (traffic complexity, time worked and quantity of automated tools) may decrease the mental tenure over time. Personal factors might have positive role (Expertise). It affects Situational awareness.
Problem Solving	Process oriented to reach a solution to cope with a problem that has been identified. It refers to the strategy the actor selects and the impact in terms of changes or requests to other actors that the strategy requires. It results in terms of efficiency (number of changes or actions required to other actors or operations)	• Information processing • Working memory • Episodic memory • Reasoning • Action planning	External conditions (traffic complexity and quantity of supporting tools) and personal factors (Expertise) may affect it.
Automated Tools Mastery	It refers to the confidence or proficiency in use of the digital supports and HCI, as described in Parasuraman et al. [30]	• Selective attention • Information processing • Procedural memory • Action planning	It highly depends on specific characteristics of the tool such as usability, predictability, algorithms reliability. The Mental workload might bring the person to err more. It also correlates with personal factors (age, expertise).
Technology Trust	Attitude that an agent will help achieve an individual their goals in a situation of uncertainty and vulnerability [31]	• Procedural memory • Episodic memory • Information processing	Trust can vary a lot: transitions between levels depends on Automated tools Mastery and Expertise.

modeling targets is to try to emulate the time variability of the ATM tasks by properly representing the ATCOs' mental resource availability and the tasks load considering the contextual conditions. Thus, when an ATCO is performing monitoring functions and a conflict is detected—namely, the possibility that two aircrafts infringe separation minima—the available mental resources will be different if at the same time the ATCO is evaluating a conflict resolution of another conflict in its sector.

Overall, the agents (with particular focus on ATCOs and pilots) are described in terms of multiple sets of attributes:

- *static attributes*: these are attributes which are determined at simulation startup time, and typically persist throughout the whole simulation run;
- *dynamic attributes*: these are human factor variables which can change at runtime, and typically allow to capture the current state of an agent, depending on the dynamic context evolution during the simulation;
- *cultural attributes*: these are attributes that are drawn at simulation startup depending on some probability distributions, which are configured according to the input configuration of the simulation.

We report in Tables 1–3 a characterization of all the variables which are represented in an agent, and which influence the evolution of the simulated ATM system. The metrics built on top of these aspects (as discussed in Section IV-E) will allow to establish a characterization of the ATM socio-technical system [27].

Some agents in the system can interact directly with each other by means of event exchange. Nevertheless, this solution does not scale, given the complexity of the scenarios which we tackle by means of ABM simulation.

Indeed, the performance of an ATM system is strongly affected by the decisions made by ATCOs involved in the process of managing the flights. Different ATCOs, depending on their expertise, might rely on different strategies to minimize the delay of flights and improve the overall safety. Moreover, different companies may have different business models that imply different objectives. In general, the human behavior which shall be captured by the ABM simulation

TABLE 3. Human-factor characterization: Socio-Cultural Factors.

Variable	Definition	Values or Resources involved	Dependencies
Team Work	Cooperation and operational support among staff people and with other organizations' professionals in achieving standard procedures simpler. It can largely affect the output of tasks. It results in a variable level of Shared Situation Awareness.	• Organization of work • Leadership style • Members turnover • Mutual trust	It may correlate with Rule Accomplishment.
Informative Communication	Information quality and quantity the actor is able to transfer to the receiver both in direct and mediated communication, in terms of priority and clarity, including also the informal or tacit communication level with attributes.	• Organization of work • Leadership style • Mutual trust	It may be positively affected by Expertise. In case of mediated communication, it may depend on the tool characteristics.
Negotiation	Inspired to Power Distance [33], the negotiation refers to the inclination of the actor to accept to receive request and accomplish them. It may result in different problem-solving style.	• Organization of work • Leadership style	It may vary according personal factors such as the Seniority and the Expertise. Differences might depend on the cultural distance.
Rules Accomplishment	Organization's tolerance towards deviation from procedures and rules. It refers to the uncertainty Avoidance [33] or tolerance towards written procedures and formal rules. It is a organizational style that affects the flexibility of individuals' behaviors in task accomplishment and error management.	• Organization of work • Leadership style	—
Safety Culture	Result of values, attitudes, perceptions, skills and behaviour patterns of individuals and a group, which determines the commitment, the approach and the skills in the management of safety [34].	• Mutual trust • Confidence in the efficacy of prevention • Training	—
Cultural Distance	Culturally related characteristics able to influence mental models and behaviors, at organizational and individual level, such as error management or timely risk recognition [33]	—	—

system is strongly related to the type of activities performed by the agents. As an example, some companies might have as their primary goal the minimization of fuel consumption, while other companies might primarily target the comfort of their customers. Accordingly, the first ones typically ask the ATCOs in a stubborn way the permission to follow routes which minimize fuel consumption, independently of the current traffic condition, while the second ones ask for routes which, e.g., avoid severe weather conditions—again, independently of the current traffic. ATCOs have to manage all these aspects at once, still guaranteeing the safety of the flights. Ultimately, these different goals of different companies, especially in heavy-loaded traffic scenarios, increase the stress of ATCOs, who are in the end legally responsible for the safety of air traffic.

Generally speaking, the relations and information flow that characterize the ATM system could be too complicated to be handled in a closed form with respect to the large amount of interactions and interconnections which characterize the high number of agents which are involved in the simulation. Therefore, the ABM simulation system which we have devised relies on a *projection module*, which allows to support complex relationship dynamics described by multiple loops and chains, nested loops, mutual cross-feed relationships connecting the agents, inhibitory connections, and preferential reactions in the face of different events. The projection module acts as a broker of events and actions, which allows to efficiently interconnect the large amount of agents in a single simulation, in a data- and event-driven way. The projection module is one of fundamental components which allows to formalize the agent interactions and to analyze the aggregated behaviour from ABM simulations. This module allows for a fast extension of the simulation model, if new agents or different algorithms/behaviors shall be included to extend the modeling capabilities of the architecture.

The ABM model represents the agent behavior in terms of tasks which, in turn, are decomposed into elementary or atomic actions, whose execution consumes time and resources (both human and/or technical). A hierarchical task analysis has been performed to identify the relevant human procedures and protocols, as well as the systems and supporting tools relevant to the simulation scenarios. This hierarchical structure allows both to model tasks performed by a single agent (e.g. the continuous monitoring of the sector activity) as well as to allocate each of the actions of a distributed procedure involving several agents (e.g. the flight handover among two sectors). In the second case, an ontology formalizes the required interaction and communication among the involved agents.

On top of time events in traditional DES systems (e.g. sector monitoring is continuously executed), the simulation is driven by the so-called traffic events, emulating human and technical-component reactions in the face of the operational context evolution—e.g. an aircraft is approaching, entering or leaving a sector, a potential conflict has been detected by the ATCO or by the supporting tools, a communication request has been received from another agent, etc. Time and Traffic events trigger the task execution which consists in performing the corresponding actions. As the result of a performed action, a new action in the same task or a new task can be triggered. In this sense, the projection module plays a fundamental role, because it becomes a co-orchestrator of task-scheduling activities, effectively bridging the gap between two different traditional programming models, namely DES and task-based ones, such as OpenMP [32].

Once an action is triggered, the agent will execute it, provided that the agent is ready and all the required resources (human and/or technical) are also available. This means that at a given time step, an agent can have a certain amount of concurrent actions to perform, which are associated with the different tasks that the agents carry out. Parallelism at action level has been taken into account considering the following criteria, which are reasonable for the ATM scenario from the human factor perspective: i) no more than two actions can be executed in parallel, except when they involve shared physical resources (voice, hearing, sight, hand touch, foot touch); ii) both potential concurrent actions belong to a skill or rule-based task; iii) none of the potential concurrent actions is a safety-related priority action—prioritization criteria depend on the contextual traffic scenario. If an action cannot be executed, then it remains in the pending state, until the system determines that it can be executed or it is cancelled because of a specific deadline (in some cases, performing an action makes sense just during a given period after being triggered).

Figure 2 shows the Finite State Machine which describes the lifecycle of an action in the system. Once an action is triggered, an object that represents it is created and handed over the projection module, along with the associated set of preconditions P, the set of resources R required for execution, and the temporal execution deadline T. A given action, thanks to the projection module, is constantly monitored. Once the conditions in P, and R are met—this could happen after that a previous action is completed, thus meeting some precondition in P, or because some resource in R becomes available—the action is started. During the execution, the action will involve all the required agents in the model. Once the action is completed successfully, the projection module and the scheduler will determine whether some additional actions are ready to be started. On the other hand, if the time required to complete the action T expires, the action is considered as failed and the task associated with the action will be either restarted or cancelled.

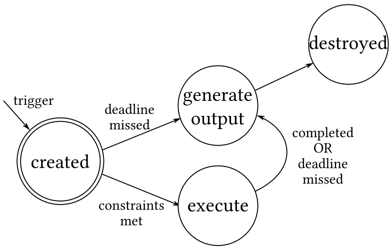


FIGURE 2. **Finite State Machine describing the Lifecycle of an Action.**

The human performance model consists of specific rules that combine traffic complexity, past traffic events and airline characteristics with the human variables identified as relevant for the specific task. This approach enables the simulation model to assign a nominal or degraded behavior to the human agents. The degraded behavior has the effect to extend the duration of the assigned action of a percentage of the nominal duration. Such a human-performance degradation model represents the effect of human behavior variability on the overall ATM system and its single parts.

The output of the simulation model is composed of two different pieces of information. On the one hand, a new so6 file is generated, which describes the simulated trajectories which show differences with respect to the planned trajectories supplied as model input. At the same time, a log file, which reports all the actions taken by the agents, is generated. These outputs are used to compute the objective measures, as we shall describe in Section IV-E.

IV. EVOLUTIONARY OPTIMIZATION FOR ATM SYSTEMS

In this section, we focus on the optimization process aimed at identifying properly-tuned configurations which increase the overall performance of the ATM system. The optimization problem that we tackle here relates to identifying the best non-dominated solution, i.e. the solution which minimizes or maximizes metrics related to the ATM scenario among all feasible solutions, i.e. solutions which do not violate critical thresholds (such as safety separation). We introduce our optimization approach based on evolutionary computing, and then we discuss the formulation of our multi-objective optimization problem and the definition of the objective functions.

In an ATM system, there are generally multiple metrics of interest to be optimized. We designed a simulation-based optimization architecture which relies on an Evolutionary Algorithm (EA), which manages a population of configurations for the ABM simulation system discussed before. The exploration of the overall search space is driven by multiple objective functions, which also target variables associated with the human behavior, such as the stress experienced by ATCOs. These functions are included in a multi-objective minimization problem, whose solutions identify the best-suited configurations of the input parameters. The multiple objectives are jointly optimized at the same time, so that the *fitness function* used to evaluate the goodness of the individuals is directly derived from these objectives.

The idea behind EAs is simple: if a population wants to thrive, it must improve constantly; the fittest individuals should inspire the new offspring, but the other individuals should not be forgotten in order to maintain diversity within the population, needed to produce outstanding individuals. By borrowing terms from biology, the individual or structure is called chromosome, and represents an encoded solution to some problem, in particular decoding into a set of parameters that are the input to the optimization function of the problem under consideration.

Overall, our optimization approach is based on a parallel/distributed variant of the NSGA-II algorithm [35]. In the following, we provide the basics to discuss our design choices of the optimization engine. This discussion is motivated by the "*no free lunch theorem*" [36], which states that no algorithm can be better than all other algorithms on all problems. We therefore describe what principles behind EAs have driven the selection of the final organization of our optimization engine, with respect to the ATM system.

The first issue that must be faced is the representation of individuals, namely the choice of the data structure defining an individual. This representation affects the operators that are applied to the individuals, and is a fundamental building block—although not the only requirement—towards the creation of a solver which could be easily adapted to multiple optimization scenarios. In EAs, the data structure storing an individual has a fixed size, and therefore it shoud be general enough to have the potential to contain an optimal solution. Conversely, in Genetic Programming (GP) approaches the representativeness problem is solved by having individuals stored in variable-size data structures, mostly in parse trees [37]. In the context of ATM optimization, we can observe that configuration variables are mostly numerical. Therefore, it is sufficient to represent an individual as a (fixed-size) set of uniform parameters, without the need to resort to more complicated GP approaches. Generally speaking, we assume that there are multiple input variables to optimize: it is possible to suppose that this number can even change in different ATM-related optimization scenarios. Anyhow, it is legit to assume that a single population is homogeneous, i.e. that all the individuals in a single population (namely, an ATM optimization problem) will share the same set of explanatory variables. Therefore, the optimization framework and algorithm will have to deal with a number of input variables to optimize which is not necessarily known a-priori, but which is stable in a single optimization process. A more complicated individual representation, like the ones used for GP, is not necessary in our scenario.

Any EA must first create an initial population of individuals. Larger populations have higher initial diversity, but there is a trade-off between the need for sufficient diversity to ensure the identification of a proper optimal solution, and the need to avoid a so-large population that slows down the time-to-solution. It is also important that the initial population encompasses a wide range of different solutions. In our ATM context, the domains upon which the single components of the individuals are known beforehand. This knowledge can be exploited in the initialization of the population. In particular, we setup an initial population with a number of individuals proportional to the space covered by the domains of the variables, and each individual is randomly initialized.

A. MULTI-OBJECTIVE OPTIMIZATION

As discussed, ATM optimization requires to solve a problem characterized by multiple objectives. These objectives are formally captured by the metrics that we will discuss in Section IV-E. These metrics can not be handled in isolation, since some of them might be in contrast to each other. As an example, the timeliness of a flight might be in contrast with safety separation. The straightest route could increase timeliness, but it might produce safety-separation violations. Thus, the problem which we address must consider all the objectives at once. It therefore comes handy the family of multi-objective optimization problems (MOPs). A MOP minimizes $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ subject to the constraints $g_i(\mathbf{x}) \leq 0 \quad i = \{1, \ldots, m\}$ and $h_j(\mathbf{x}) = 0 \quad j = \{1, \ldots, p\}$, where k is the number of objectives and $\mathbf{x} = (x_1, \ldots, x_n)$ is an N-dimensional vector.

Having several objective functions f_i, the notion of optimum becomes the *Pareto optimality* (or Pareto efficiency), which is a state of the variables under optimization from which it is impossible to move so as to make any other configuration better off, without making at least one variable worse off. Considering the example of timeliness vs safety separation, this concept becomes clearer: timeliness might not be the best configuration on its own, but increasing timeliness might make safety separation worse. Therefore, with this approach, the search space is explored with the goal of finding the best-suited trade-offs among all the variables to be optimized. More formally, given $\mathbf{f} : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^k$, $\Omega \neq \emptyset$, $k \geq 2$ the *Pareto Front set*

$$PF^* = \left\{\mathbf{f}(\mathbf{x}_i^*) | \mathbf{x}_i^* \in \Omega \wedge \nexists \mathbf{x} \in \Omega \text{ s.t. } \mathbf{f}(\mathbf{x}) \preccurlyeq \mathbf{f}(\mathbf{x}_i^*)\right\} \quad (1)$$

is called the *global minimum*, where the set of all $\mathbf{x}_i^*$ is called the global minimum solution set, $\mathbf{f}$ is the multi-objective function, Ω is the feasible region, and $\preccurlyeq$ is used to indicate the *Pareto dominance*.

The concept of Pareto Front is illustrated in Fig. 3, in a sample search space composed of only the two objective functions introduced before. The boxed points represent feasible solutions to the problem, and smaller values are preferred to larger ones—a minimization problem. Point C is not on the Pareto front because it is dominated by both point A and point B. Points A and B are not strictly dominated by any other, and hence do lie on the front. Deciding to pick either A or B as the final solution highly depends on the problem being tackled, and/or the search algorithm being adopted.

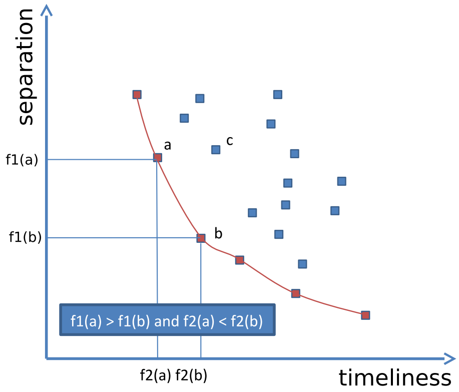


FIGURE 3. Example of a two-dimensional Pareto front.

Multi-objective EAs (MOEAs) algorithms which allow to solve a MOP are generally based on three different techniques: i) *a-priori techniques*, ii) *progressive techniques*, and iii) *a-posteriori techniques*. A-priori techniques require to define the relative importance of the MOP objectives prior to the search. The optimum solution is then obtained by optimizing each objective in sequence, from the most to the least important. This strategy is not suitable for ATM systems, because it is not practical to assign a priority to different objectives. In this scenario, Foruman has proposed [38] to define an order by selecting at each evaluation a random objective. This could lead to wrong results in our context—as an example, safety-separation could be deemed to be not important, which is unacceptable. Similarly, progressive techniques have to define preferences to bias the search. The main issue here lies in the definition of the methods to incorporate the preferences into the MOEA.

We have therefore decided to exploit an a-posteriori technique, which tries to generate the elements of the Pareto optimal set (performing a broad search) followed by a decision-making process. There are several subclasses of these algorithms. We have explicitly discarded approaches which compute the fitness of the individuals according to some linear combination of the objective functions, or which consider only a subset of the objectives, as these approaches have been proven to suffer from an increased time-to-solution or they could miss an optimal solution at all [39]. This has been the only choice, given the large search space characterizing ATM systems, and the criticality of the identification of the optimal solution for the redesign of such systems.

Conversely, we have adopted a Pareto-based approach, which assigns a fitness value to each individual according to Pareto dominance, and therefore directly considers for reproduction all non-dominated individuals in the current population. Since those non-dominated solutions once found are not guaranteed to survive in the next generation, an additional population containing all non-dominated solutions found so far is included [40]. This secondary population is updated to include the newly-found non-dominated solutions and is periodically culled, since Pareto dominance is dependent upon the set on which it is evaluated. The drawback of this approach is that the update rate drastically increases the cost of the algorithm, and as the population size grows, the comparison time may become dominant. This is a relevant aspect. Considering that the evaluation of the population is based on an ABM stochastic model, a large number of simulations is required to effectively explore the search space. In our implementation, we paid close attention to reducing the time required to carry out a single simulation. Based on this, we have shown experimentally (see section VI-A) that, in our context, this approach is computationally viable.

We borrow the Pareto-based approach from NSGA-II [35]. The fitness of an individual is an integer number (the *rank*), representing a non-dominated front. An individual having rank equal to zero is a non-dominated solution in the current set, and individuals with rank i, with $i \leq M$ and M the population size, are dominated only by individuals with smaller ranks. In order to sort the individuals into non-dominated ranks, we rely on the fast non-dominated sorting approach requiring $O(kM^2)$ time and $O(M^2)$ space.

B. ELITISM

Another aspect to deal with in EC is *elitism*. This concept, borrowed from sociology, describes the belief that individuals who form an elite are more likely to be constructive to society as a whole, and therefore deserve influence or authority greater than the others. In the EA jargon, elitism refers to the high likelihood that a few individuals in the population are the best ones, and therefore should be preserved in the next iterations as much as possible.

Given the large search space of ATM optimization problems, elitism can be exploited as a way to reduce the time-to-solution. We preserve elitism by the strategy used to make the population evolve. In particular, after the computation of the rank of the individuals in the current generation of the population, we select the best-suited ones for reproduction, in order to generate an offspring. Different strategies exist to properly deal with the offspring. Our strategy is based on a variation of the Double Tournament Selection, called *Binary Tournament Selection* [41], where a group of two creatures is taken from the population and the most fit individual is picked as a parent, repeating the process to get a second parent, choosing whether the first parent is excluded in the second selection. In particular, we randomly choose two individuals and if they differ in the non-domination rank, we select the solution with the lower (better) rank. Otherwise, if both solutions belong to the same front, then we select the solution that is located in a ‘‘lesser crowded’’ region, to preserve diversity. The selection is based on a combined population, containing individuals of the offspring and individuals of the parent population. Since both the previous and current individuals are included, elitism is ensured.

C. CROSSOVER AND MUTATIONS

In EAs, children can be generated by applying either a single-point or a double-point crossover operator to the pair of parents, depending on the number of variables which are used to represent an individual. The crossover operator is used to combine disparate creatures facilitating novelty, and therefore leading to a broad search. In particular, it is not viable for our ATM scenario to consider an individual as a mere bit string, as in generic EAs. Indeed, as we have already discussed, every input variable is associated with a feasibility domain. This domain already limits the search space, because it excludes configurations which are a-priori known to be unacceptable with respect to the optimization measures. As an example, setting the minimum horizontal separation between two aircraft to a too low value would increase the likelihood of safety-separation violations. Since, as mentioned, our input parameters are all numerical (in most cases, they are real numbers), applying a crossover operator in the middle of the binary representation of a parameter will likely move far away the parameter from the feasible domain—this is related mainly to the binary representation of floating-point values. Our crossover operator, therefore, determines the crossover point at the granularity of a single (atomic) parameter. If the number of parameters is high (over a predetermined threshold), we identify two crossover points, and the exchange of portions of the parameters is carried out in between these two points. In case of a large number of parameters, a two-point crossover operator is likely to explore the search space more conservatively, preventing individuals in a new generation to drift too much from points associated with the parents in the search space.

With some random ϵ-probability, the newly-generated individual is *mutated*. The mutation operator facilitates the local search (especially when a tiny portion of the individual representation is mutated). Again, mutations must not violate the domains of the parameters. To this end, a mutation is applied in terms of an increment/decrement of the current value of the parameter, which anyhow is tested to keep the mutated value in the domain. The magnitude of this variation is such that the variable does not oscillate from the previous value of an amount greater than 10% of the span covered by its associated domain. Empirically, we have observed that this criterion fastens the convergence time to an optimum in ATM scenarios.

D. DIVERSITY PRESERVATION

In our ATM scenario, we do not want the overall population to grow to a large extent, particularly to keep bounded the amount of time which is required to run the simulations needed to evaluate the fitness of the individuals. Therefore, once the children have been generated, we borrow the replacement technique of NSGA-II, to ensure that the average fitness increases over time, to guarantee a sufficient diversity in the population, and to enhance elitism. This is an important point, because (as discussed) losing genetic variance can lead to sub-optimal solutions.

In more details, we adopt a non-dominated sorting approach, where for each solution we calculate the number of solutions dominating it (the *domination count*), and the set of solutions it dominates—this requires $O(kM^2)$ comparisons. Each non-dominated solution has a domination count equal to zero. Then, for each solution with domination count zero, we reduce the rank of the members of the set of solutions it dominates, and if the domination count of any member becomes zero, we put it in a separate list representing the second non-dominated front. The process is repeated to identify all fronts. Since each solution can have dominance rank at most $M - 1$, it will be visited at most $M - 1$ times before identifying its front, thus requiring $O(M^2)$ time. The overall complexity is therefore $O(kM^2)$.

Diversity preservation is faced up by using the *crowding distance estimator*. This estimator assigns to each solution a real number representing the density of solutions surrounding it. The crowding-distance computation requires sorting the population according to each objective function value in ascending order of magnitude. For each objective, the solutions with the highest and the smallest values are assigned an infinite crowding distance value. Each other solution is assigned a value equal to the difference between the values of its adjacent solutions in the current sorting, divided by the difference between the highest and smallest values. The computation is continued with other objective functions and the overall crowding-distance value is calculated as the sum of individual distance values for each objective. In other words, the crowding distance is an estimate of the density of solutions surrounding a particular solution. Fig. 4 shows a graphical representation of this metric in a two-dimensional world.

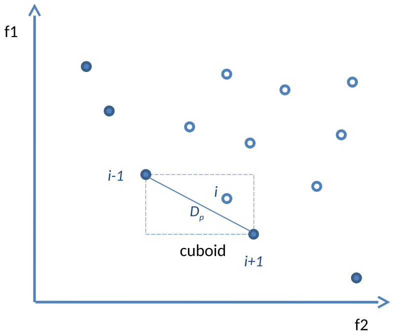


FIGURE 4. Crowding-distance calculation.

When a non-dominated set does not fit entirely in the new population, because the number of its members is higher than the remaining space, we sort the solutions in it using the crowded-comparison operator in descending order and choose only the best solutions to fill the remaining new population slots.

E. ATM OBJECTIVE FUNCTIONS

The objective functions which we use to drive the exploration of the search space are performance metrics which are related to both human and physical aspects of the ATM systems. We introduce the following parameters that we use to define the objective functions:

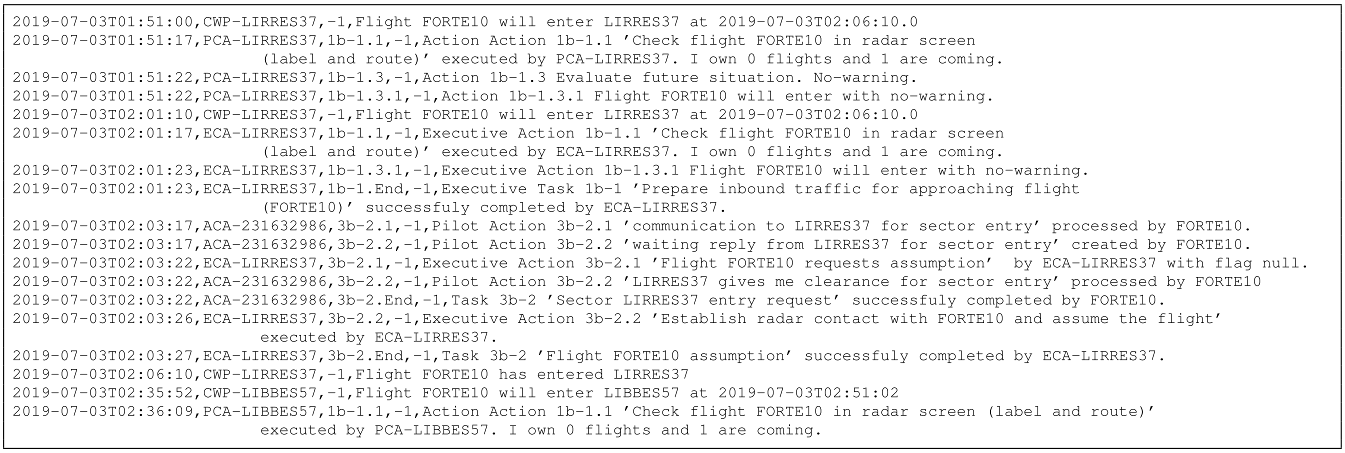
- M is the total number of involved flights,
- N_m is the total number of sectors traversed by one flight,
- t_{n-1} is the time at the entrance in a sector,
- t_n is the time at the departure from the sector,
- T_n is the expected time to cover that sector,
- f_n is the specific fuel consumption per unit thrust and unit time,
- S_n is the throttle setting of the thrust (in Newton),
- G is the total number of groups of ATCOs, and
- N_m is the total number of actions of group m.

The performance metrics we consider are the following:

- *Timeliness*. It is evaluated as the time in seconds in a sector, namely:

$$\Delta t = \sum_{m=1}^{M} \frac{1}{N_m} \sum_{n=1}^{N_m} \frac{t_n - t_{n-1}}{T_n} - 1$$

- *Occupancy*. It is defined as the number of flights in a given sector, in a specified timeframe (in our case, 30 minutes).



```
2019-07-03T01:51:00,CWP-LIRRES37,-1,Flight FORTE10 will enter LIRRES37 at 2019-07-03T02:06:10.0
2019-07-03T01:51:17,PCA-LIRRES37,1b-1.1,-1,Action Action 1b-1.1 ’Check flight FORTE10 in radar screen
                          (label and route)’ executed by PCA-LIRRES37. I own 0 flights and 1 are coming.
2019-07-03T01:51:22,PCA-LIRRES37,1b-1.3,-1,Action 1b-1.3 Evaluate future situation. No-warning.
2019-07-03T01:51:22,PCA-LIRRES37,1b-1.3.1,-1,Action 1b-1.3.1 Flight FORTE10 will enter with no-warning.
2019-07-03T02:01:10,CWP-LIRRES37,-1,Flight FORTE10 will enter LIRRES37 at 2019-07-03T02:06:10.0
2019-07-03T02:01:17,ECA-LIRRES37,1b-1.1,-1,Executive Action 1b-1.1 ’Check flight FORTE10 in radar screen
                          (label and route)’ executed by ECA-LIRRES37. I own 0 flights and 1 are coming.
2019-07-03T02:01:23,ECA-LIRRES37,1b-1.3.1,-1,Executive Action 1b-1.3.1 Flight FORTE10 will enter with no-warning.
2019-07-03T02:01:23,ECA-LIRRES37,1b-1.End,-1,Executive Task 1b-1 ’Prepare inbound traffic for approaching flight
                          (FORTE10)’ successfuly completed by ECA-LIRRES37.
2019-07-03T02:03:17,ACA-231632986,3b-2.1,-1,Pilot Action 3b-2.1 ’communication to LIRRES37 for sector entry’ processed by FORTE10.
2019-07-03T02:03:17,ACA-231632986,3b-2.2,-1,Pilot Action 3b-2.2 ’waiting reply from LIRRES37 for sector entry’ created by FORTE10.
2019-07-03T02:03:22,ECA-LIRRES37,3b-2.1,-1,Executive Action 3b-2.1 ’Flight FORTE10 requests assumption’  by ECA-LIRRES37 with flag null.
2019-07-03T02:03:22,ACA-231632986,3b-2.2,-1,Pilot Action 3b-2.2 ’LIRRES37 gives me clearance for sector entry’ processed by FORTE10
2019-07-03T02:03:22,ACA-231632986,3b-2.End,-1,Task 3b-2 ’Sector LIRRES37 entry request’ successfuly completed by FORTE10.
2019-07-03T02:03:26,ECA-LIRRES37,3b-2.2,-1,Executive Action 3b-2.2 ’Establish radar contact with FORTE10 and assume the flight’
                          executed by ECA-LIRRES37.
2019-07-03T02:03:27,ECA-LIRRES37,3b-2.End,-1,Task 3b-2 ’Flight FORTE10 assumption’ successfuly completed by ECA-LIRRES37.
2019-07-03T02:06:10,CWP-LIRRES37,-1,Flight FORTE10 has entered LIRRES37
2019-07-03T02:35:52,CWP-LIBBES57,-1,Flight FORTE10 will enter LIBBES57 at 2019-07-03T02:51:02
2019-07-03T02:36:09,PCA-LIBBES57,1b-1.1,-1,Action Action 1b-1.1 ’Check flight FORTE10 in radar screen (label and route)’
                          executed by PCA-LIBBES57. I own 0 flights and 1 are coming.
```

FIGURE 5. Excerpt of a log generated by the simulation model.

- *Fuel consumption*. It is the estimated amount of fuel used by the aircraft, computed as:

$$F = \sum_{m=1}^{N} \sum_{n=1}^{N_m} f_n \cdot S_n \cdot (t_n - t_{n-1})$$

- *Safety*. It is evaluated through minimum vertical or horizontal separation, whose minimal thresholds depend on the type of route (e.g., national vs transoceanic)
- *ATCO workload*. It is a human-related metric which describes the cognitive resources required to accomplish the various tasks. Namely, it is numerically expressed as:

$$W = \sum_{g=1}^{G} \omega_g N_g$$

In the last metric, the parameter ω_g measures the comparable workload. This parameter depends on a characterization of the ATCO. In particular, we consider different aspects which affect the value of ω_g, namely the agents’ attributes previously presented in Table 1.

Both physical and human-related metrics are computed starting from the output of a simulation run. As mentioned, the simulator generates an so6 file and a log of timestamped events. From the former, we evaluate the trajectory-related metrics associated with occupancy, timeliness, safety, and fuel consumption. Timeliness, occupancy, and fuel consumption are computed relying on the traffic library [42] and OpenAP [43]. To this end, we compare the trajectories recorded in the so6 file generated by the simulator, and the ones in the original input so6 file—which is preprocessed to determine the reference values for the performance metrics. For each mutated trajectory, we are able to determine the impact on the original metrics, and generate the new measures.

From the generated log file, we compute the final human-related metrics. In particular, this can be done by parsing the log—an excerpt of the log of one simulator is reported in Fig. 5—so as to identify what ATCOs are involved in which actions. The number of actions per timeframe which are found in the log file determine an increase in the workload, accounting for all the characteristic variables which are discussed in Table 1.

It is interesting to note that our approach decouples the metrics of interest form the simulation-based optimization loop. In this sense, it is possible to apply the same simulation-based optimization scheme in order to study different ATM scenarios, only by specifying additional metrics to be maximized or minimized. Conversely, as stated in Section III, if an extension or a change to the simulation model is provided, as well as the inclusion of additional decision variables, the same metrics (and therefore the same ATM scenario) can be studied under different perspectives, possibly in more detail.

V. THE DISTRIBUTED OPTIMIZATION ARCHITECTURE

MOEAs have been proven to be one of the most effective ways to solve MOPs, since they are able to produce several non-dominated solutions in a single run. However, to discover good solution sets, they require a large number of solutions to be evaluated during each iteration. This can be problematic for our ATM target. As explained, our Evolutionary Optimization approach is based on a stochastic ABM model, which therefore requires multiple runs, configured with different random seeds, to determine how well (from a performance-metrics point of view) a certain configuration of the model’s explanatory variables (an individual) behaves.

With respect to the computational demand of the discussed EA approaches, in [44] a comparison between generational and steady-state asynchronous EAs has been carried out on two popular algorithms (NSGA-II [35] and SPEA2 [45]). They concluded that the level of variance in time to evaluate a solution and the parallelization ratio are the key factors that influence the relative performance between steady-state asynchronous and generational EAs. The authors showed that a substantial speedup can be achieved with steady-state asynchronous algorithms.

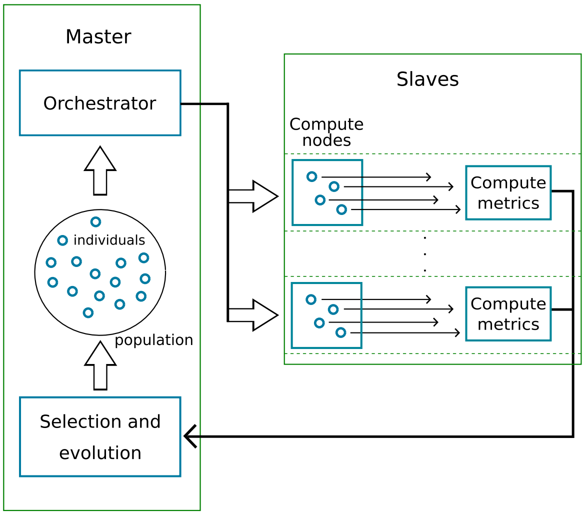


FIGURE 6. **Conceptual organization of the optimization architecture.**

The differences between steady-state and generational algorithms are more apparent when dealing with simulation-based optimization, where the simulation times can differ drastically from one solution to another. The Master-Slave parallelization paradigm [46] is well-suited for this case.

The overall organization of the optimization architecture is therefore depicted in Fig. 6. The logical building blocks of the Master-Slave architecture are the following:

1) *The Orchestrator*: it is responsible for splitting the current population into different subsets of individuals, distributing the sets to the different available computing nodes, and controlling them along the computation.
2) *Compute nodes*: they evaluate (through stochastic simulations) the goodness of an individual of the population. These compute nodes represent performance-critical elements. Indeed, for each individual of the population, the orchestrator schedules for execution multiple runs (the number is parameterizable) of the simulation model (with different random seeds). We refer to the multiple runs of the same configuration of the simulation model as a *batch of simulation*.
3) *Compute metrics*: they are activated after that a batch of simulation is completed, and compute the average values for the metrics of interest. In this way, given that each execution in a batch is configured using a different random seed, the different behavior of the stochastic runs can be safely captured when evaluating an individual. We emphasize that this simulation-based approach allows us to avoid scenarios in which, due to stochasticity, the EA-based optimization process might oscillate across two different points in the search space. The final metrics are here computed from the logs of the ABM simulation and are associated with the individual.
4) *Selection and evolution*: this logical block applies the evolutionary strategy to evolve the current population towards the optimal solution.

The master is therefore responsible for managing the current population, running the Orchestrator and performing the selection and evolution task. The slaves run a loop which executes the following steps: 1) receive in input a subset of the current population, 2) spawn instances of the ABM model on each available processing unit of the computing infrastructure by passing the explanatory variables taken from an individual in input to the simulation model, 3) wait for the completion of a batch of simulation, 4) parse the log files, 5) compute all the average metrics of interest and associate their values to the individual. Once all the subsets of the population have been evaluated by a slave, the metrics are sent back to the orchestrator, which is able to apply the aforementioned evaluation and selection strategy.

VI. EXPERIMENTAL EVALUATION

In this section, we present an experimental assessment of our simulation-based evolutionary optimization architecture, targeting two main goals. As a first goal, we assess the computational performance of the implementation of our architecture, also taking into account the fact that population evaluation is simulation-based. We also evaluate the performance sensitivity to the delay introduced by the execution of the ABM model, which, as discussed, can be non-minimal given the complexity of the problem being tackled. As a second and overall goal, we assess the capabilities of our optimization architecture on real-world scenarios.

We have implemented the software components of our EA-based optimization architecture in the C11 language, relying on the Message Passing Interface (MPI) [47], to support scatter/gather primitives to distribute the population to the slaves and recollect the computed metrics. In the experimental assessment we present, we have used a dedicated compute cluster composed of the following four heterogeneous compute nodes:

- Two HP ProLiant servers equipped with 4 AMD Opteron 6128 multicore processors, each one encompassing 8 cores, for a total of 32 cores per node, and 32 GB of RAM per each node;
- One HP ProLiant server, equipped with 2 AMD Opteron 6174 multicore processors, each one having 12 cores, for a total of 24 CPU cores, and 32 GB of RAM;
- One server equipped with 4 AMD Opteron 6168 multicore processors, each one with 12 cores, resulting in a total of 48 cores, and 128 GB of RAM.

Overall, our computing infrastructure accounts for 136 cores and 524 GB of RAM, making it a medium-size infrastructure. The system software on each node is Linux 4.19.13 (taken from the Debian 9 distribution), the compiler used is gcc 8.2.0. As for the MPI runtime, we have used MPICH 3.3 (Hydra runtime).

A. COMPUTATIONAL PERFORMANCE AND SENSITIVITY TO SIMULATION TIME

To study the computational performance of our architecture, we used a synthetic benchmark. We used a population of 50 individuals described by 10 explanatory variables, which have been randomly generated with no constrained domain. The number 10 has been selected as a worst-case scenario for the goals of our kind of ATM optimization. In particular, at the time of this writing, we have not been able to identify a single real-world optimization scenario which would hit this number of parameters.

For the sake of this performance study, we have replaced all metrics with uniform pseudo-random number generators. This means that the population which is kept in a different iteration of the evolutionary optimization is randomly selected. This has been done because we are not actually interested in the population being selected, rather on internal execution dynamics.

Taking inspiration from the Hold benchmark traditionally used to stress test software data structures [48] and the PHold model for Parallel Discrete Event Simulation [49], we have replaced the inner part of the evolutionary optimization architecture with a busy loop, which mimics the execution delay of arbitrary operations—this is, in our case, the execution latency of a simulation model.

The duration of this loop (in seconds) has been determined according to a Normal distribution, with standard deviation $\sigma = 0.1, 0.2, 0.5$ and with three different means $\mu = 0.5, 1, 2$. Moreover, we have considered the special case in which no duration of the busy loop was considered. The mean value of the busy loop has been set to the order of seconds, because we observed that this is the order of magnitude of an execution of a simulation. Overall, this preliminary experimental assessment (which has been carried out to highlight the functional capabilities of the optimization architecture) is organized to be representative of the operating settings for the optimization problems.

We have carried out experiments running on all available 136 cores. The total population used in this evaluation has been set to 1360 individuals—we consider this to be significant for ATM optimization contexts with this number of explanatory variables in the individuals. The number of total iterations has been set to 10. We report results averaged over 5 different optimization runs.

In Table 4, we report the total execution time (in hours) required to run the synthetic PHold benchmark which we have described above. On the other hand, Table 5 shows the slowdown produced by the busy loop in the overall optimization architecture when executed with no delay at all—in this case, the overall execution time amounts to 1,36 seconds.

TABLE 4. Time required to run the PHold test case in the distributed environment (in hours).

σ μ	0.1	0.2	0.5	stddev
0.5	2.10	2.15	1.33	0.46
1	2.02	1.52	1.18	0.42
2	6.65	7.08	6.88	0.21

TABLE 5. Slowdown with respect to the evolutionary optimization architecture with no delay.

σ μ	0.1	0, 2	0, 5
0.5	5571.10	5681.30	3531.00
1	5339.40	4013.30	3133.60
2	17601.60	18729.90	18204.10

By these results, we can draw two conclusions. On the one hand, the optimization architecture which we have designed is quite efficient and its convergence time does not affect the overall performance of the ATM-optimization framework. On the other hand, as expected, in a simulation-based evolutionary computation system, the part of the loop which affects more the duration of the algorithm to find an optimal solution is the execution of the simulation model. Although the slowdown figures might look large, we note that (Table 4) the total time required to find an optimal solution are perfectly acceptable, given the complexity of the problem which we tackle. Additionally, we can observe (Table 4, last column) that the overall optimization architecture is quite resilient to variations in the duration of the simulation. This result suggests that there is no actual requirement to devise more complicated strategies to schedule the execution of simulations on the distributed environment, which could anyhow increase the deploy complexity.

The fact that for higher standard deviations smaller total execution times are observed is related to the Normal distribution defined on $[-\infty, +\infty]$, while we use this distribution to setup a busy loop. If a negative value is drawn from a distribution, this value is artificially forced to zero. This artifice happens more often with larger values of the standard deviation, and is observed in the reduced amount of total execution time.

B. RESULTS IN REAL-WORLD SCENARIOS

To evaluate the capabilities of our simulation-based optimization architecture, we have relied on a real-world ATM scenario addressed in the execution phase of the SESAR Solution #32 ‘‘Free Route through the use of direct routing’’, and the and SESAR Solution #33 ‘‘Free Route through Free Routing’’ [50]. The direct routing environment highlights the different aspects of the ATM socio-technical complexity, specifically: a) cognitive aspects, dealing with the cognitive behavior of human actors, b) social aspects, dealing with the place and the role of human actors within the system, c) technical aspects, dealing with the engineering of the technical components and of the system architecture, and d) interrelations of the above-mentioned three aspects.

We considered a real-world high-complexity traffic scenario in this part of the experimentation. It is related to a number of elementary sectors within the Brindisi (LIBB) and Rome (LIRR) Area Control Centres (ACCs) of the Italian airspace. Specifically the operational configuration includes four collapsed sectors belonging to Brindisi (LIBBCN37, LIBBCS37, LIBBES37, LIBBND37) and one belonging to Rome (LIRRUS37). The considered flights are those over FL315, which is the threshold for direct routing. Each considered sector has been collapsed. The number of flights in the day are 378.

To perform a preliminary validation to generate trust in the simulation model, we have carried out a comparison between the simulated and the actual timeliness and occupancy in a time window. Fuel consumption has not been used for validation purposes due to the adoption of DDR2 data, which do not allow an accurate determination of the fuel flow in a given trajectory point of a given aircraft. In Fig. 7 we report, for some simulated flights, the comparison between the simulated, planned and flown trajectory. The planned trajectory is the one which is established before the flight is actually departed. The flown trajectory is associated with the actual trace (taken from the so6 file), which is associated with direct routing. Conversely, the simulated trajectory is the output of our simulation model. Similarly, in Fig. 8 we report, for some sectors, the comparison between the simulated and the flown occupancy. The fact that in both Figures the results between the simulated and the flown data are very close is an indication of the capability of our simulation system to well capture the dynamics of individual flights (Figure 7) and also to well capture emergent behavior (Figure 8).

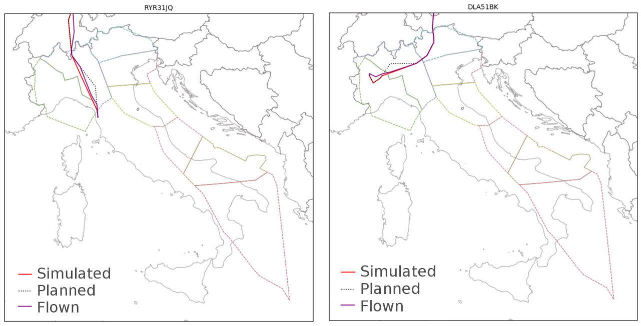


FIGURE 7. Comparison between the simulated, planned and flown trajectory.

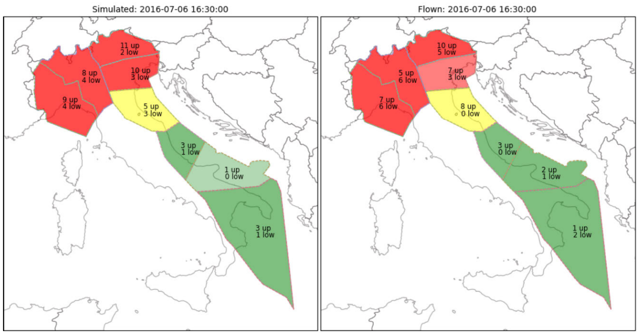


FIGURE 8. Comparison between the simulated (left) and flown (right) occupancy.

Fig. 9 reports an example of the comparison of the occupancy each 30 minutes in the LIBBCN sector. As it is possible to see, the trends are nearly the same. The behavior is more or less the same for the other sectors and days. The deviation of the achieved timeliness in simulated traffic with respect to that evaluated on the actual traffic in percentage does not overcome 10% on all sectors. The deviation of the achieved occupancy in simulated traffic with respect to that evaluated on the actual traffic in percentage does not overcome the 7% on all sectors.

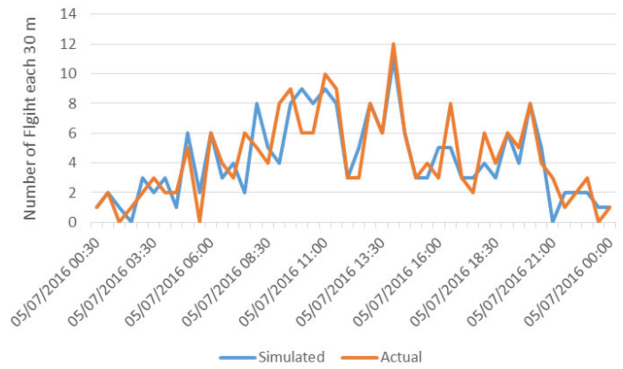


FIGURE 9. Simulated/Actual Occupancy for the LIBBCN Sector on 2016-07-05.

As a second experimentation, we have selected two optimization variables and we have run the evolutionary optimization module to fine tune the values of these variables. The variables are the horizontal separation minima for the ATCO Planner (HP) and the ATCO Executive (HE), expressed in nautical miles. We have run a total of 50 iterations, and we have used the trajectory-related metrics discussed in Section IV-E to drive the exploration of the search space, namely occupancy per 30 minutes (i.e., average number of flights every 30 minutes) and timeliness (in seconds). We have run the evolutionary optimization basing the simulation on the traffic in the aforementioned scenario focusing on the time slot from 12 pm to 2 pm on July 6, 2016 related to the ACC of Brindisi. By the nature of the simulated traffic, we categorize this experiment as a ''high-complexity'' one.

In Table 6, we report the results related to the first 10 iterations of the evolutionary optimization. After 10 iterations, the evolutionary optimization is able to identify the Pareto-optimal configuration—this individual has been generated by means of mutation on the previous iteration, and in the later iterations no better individual has been found, also by mutation. The optimization has been particularly challenging in the proposed case, due to the high number of objective metrics and the low number of decision variables. The obtained results correctly minimize the objective metrics according to Pareto dominance. In addition, it provides more ''conservative'' values for the separation minima, which are anyway able to minimize occupancy and timeliness metrics. This is related to the specific nature of the simulated traffic of the high-complexity scenario, which is mostly a sliding traffic over the ACC of Brindisi.

We have also run a similar optimization, with respect to optimization variables and metrics, but considering a more

TABLE 6. Convergence history for partial traffic on July 6, 2016, Area Control Center of Brindisi.

#	HP	HE	LIMME (occ.)	LIMMWC (occ.)	LIPPCS (occ.)	LIPPN (occ.)	LIPPSD (occ.)	LIMME (tim.)	LIMMWC (tim.)	LIPPCS (tim.)	LIPPN (tim.)	LIPPSD (tim.)
1	7.2	4.8	4.08	4.00	4.08	5.23	3.88	537.63	430.72	803.59	679.80	532.49
2	8.8	7.0	4.10	4.00	4.10	5.23	3.88	533.14	429.86	801.55	680.31	532.41
3	8.8	7.0	4.10	4.00	4.10	5.23	3.88	533.14	429.86	801.55	680.31	532.41
4	8.8	7.0	4.10	4.00	4.10	5.23	3.88	533.14	429.86	801.55	680.31	532.41
5	8.8	7.0	4.10	4.00	4.10	5.23	3.88	533.14	429.86	801.55	680.31	532.41
6	8.8	7.0	4.10	4.00	4.10	5.23	3.88	533.14	429.86	801.55	680.31	532.41
7	8.8	7.0	4.10	4.00	4.10	5.23	3.88	533.14	429.86	801.55	680.31	532.41
8	8.8	7.0	4.10	4.00	4.10	5.23	3.88	533.14	429.86	801.55	680.31	532.41
9	8.8	7.0	4.10	4.00	4.10	5.23	3.88	533.14	429.86	801.55	680.31	527.54
10	8.6	7.0	4.06	3.96	4.02	5.23	3.90	535.22	430.29	808.37	678.69	527.54

TABLE 7. Convergence history for partial traffic on July 6, 2016, Area Control Centers of Milan and Padua.

#	HP	HE	LIBBCN (occ.)	LIBBCS (occ.)	LIBBES (occ.)	LIBBND (occ.)	LIRRUS (occ.)	LIBBCN (tim.)	LIBBCS (tim.)	LIBBES (tim.)	LIBBND (tim.)	LIRRUS (tim.)
1	6.9	5.6	2.77	3.19	3.04	3.77	2.10	552.19	763.08	574.32	549.08	618.03
2	6.9	7.0	2.85	3.21	3.04	3.83	2.08	545.42	773.08	578.98	545.29	623.08
3	7.3	6.9	2.79	3.19	3.04	3.81	2.15	550.84	757.37	568.10	555.08	606.49
4	6.8	6.6	2.77	3.17	3.02	3.79	2.08	549.18	770.27	577.98	552.04	622.77
5	8.8	6.4	2.77	3.19	3.00	3.83	2.10	559.57	766.15	573.86	551.21	618.25
...	...	...	...	...	...	...	...	...	...	...	...	...
13	7.2	6.6	2.77	3.21	3.00	3.79	2.10	551.83	756.40	574.92	554.83	615.26
14	7.2	6.9	2.81	3.23	3.00	3.79	2.13	549.99	769.04	566.87	552.51	614.41
...	...	...	...	...	...	...	...	...	...	...	...	...
16	7.2	4.7	2.79	3.23	3.02	3.83	2.13	552.22	753.99	566.84	551.86	613.96
...	...	...	...	...	...	...	...	...	...	...	...	...
23	6.6	4.7	2.75	3.17	3.00	3.81	2.13	554.45	767.27	578.16	552.09	615.47
...	...	...	...	...	...	...	...	...	...	...	...	...
50	6.6	4.7	2.75	3.17	3.00	3.81	2.13	554.45	767.27	578.16	552.09	615.47

complicated simulation scenario. In particular, we have optimized the aforementioned horizontal separation minima for the traffic on the 6th of July 2016 in the time slot from 12 pm to 2 pm, considering flights within the ACC of Milan and Padua—collapsed sectors LIMME37, LIMMWC37, LIPPCS37, LIPPN37, LIPPSD37, LIMMWC37, LIPPCS37, LIPPN37, LIPPSD37. By the nature of the simulated traffic, we classify this experiment as a ‘‘very-high complexity’’ one.

In Table 7, we report the results related to an excerpt of the first 50 iterations. Also in this case, the optimization is particularly challenging—there is a high number of objective metrics and a low number of decision variables. From the results, we can draw multiple conclusions. On the one hand, the results obtained from the optimization procedure exhibit more ‘‘aggressive’’ values for the separation minima with respect to their nominal values. This is an expected result, also due to the nature of the very-high complexity of the simulated traffic, which is composed of sliding, crossing, climbing and descending parts over the ACC of Milan and Padua. On the other hand, it could be argued that ‘‘multiple’’ optimal solutions are identified. We refer, in particular, to the best-fit individuals identified at iterations 16 and 23—the best-fit individual at iteration 23 was generated by means of mutation. The former on average implies better (i.e., lower) values for the occupancy metrics, whereas the latter on average implies better (i.e., lower) values for the timeliness metrics. While a human might decide that the individual at iteration 16 could be deemed ‘‘better’’ than the one at iteration 23, Pareto-optimality selects the latter as the best-suited one. This is coherent with the contrasting nature of occupancy and timeliness metrics, whose optimization requires somehow a compromise. Moreover, these results highlight the importance of the mutation operator in this kind of ATM optimization.

In a third experiment, we have carried out a different kind of optimization. In particular, we have optimized the operational sectorization configuration of the environment in terms of allocation of elementary sectors in the collapsed sectors of both executive controllers and planning controllers to optimize trajectory-related performances. The decision variables are represented by the combination of elementary sectors in the collapsed sectors, which represent the actual simulated sectors for each iteration of the optimization module. A number of 5 iterations is considered for the convergence threshold of the optimization module. A number of 6 individuals is applied for the population size at each iteration. The human-related metrics encompass the number of communications between the executive controllers (ExeC) and the flight crew (FC). We have used the real-world traffic on July 3, 2019 in the time slot from 12 pm to 2 pm, considering the ACCs of Milan and Padua, which corresponds to our ‘‘very-high complexity’’ scenario. In this scenario, we have also used human-related metrics, with the ultimate goal of minimizing also the workload suffered by ATCOs.

The obtained solution (highlighted in Table 8) on average minimizes the objective metrics. Indeed, even if other solutions exhibit better values for the total metrics (e.g., solution 1, which aggregates as much as possible the elementary sectors) or for the standard deviation metrics (e.g., solution 2, which separates as much as possible the

TABLE 8. Convergence history for partial traffic on July 3, 2019, Area Control Centers of Milan and Padua.

#	Optimal Sector Configuration	Total ExeC Communication	Standard Deviation of Exe Communication	Total ExeC Separations	Standard Deviation of ExeC Separations	Total PC Separations	Standard Deviation of PC Separations
1	LIMME37; LIMMWC37; LIPPCS37; LIPPN37; LIPPSD37	3332	235.65	1371	81.82	3676	429.90
2	LIMME3; LIMME4; LIMME5; LIMME6; LIMME7; LIMMWC3; LIMMWC4; LIMMWC5; LIMMWC6; LIMMWC7; LIPPCS3; LIPPCS4; LIPPCS5; LIPPCS6; LIPPCS7; LIPPN3; LIPPN4; LIPPN5; LIPPN6; LIPPN7; LIPPSD3; LIPPSD4; LIPPSD6; LIPPSD37	5325	66.27	1836	34.59	3865	55.25
3	LIMME37; LIMMWC37; LIPPCS3; LIPPCS4; LIPPCS5; LIPPCS6; LIPPCS7; LIPPN3; LIPPN4; LIPPN5; LIPPN6; LIPPN7; LIPPSD3; LIPPSD4; LIPPSD6; LIPPSD7	4195	186.94	1446	82.21	3155	152.38
4	LIMME34 LIMME57 LIMMWC34 LIMMWC57 LIPPCS3 LIPPCS4 LIPPCS5 LIPPCS6 LIPPCS7 LIPPN3 LIPPN4 LIPPN5 LIPPN6 LIPPN7 LIPPSD3 LIPPSD4 LIPPSD6 LIPPSD7	4514	130.82	1555	61.59	3330	105.18
5	LIMME34; LIMME57; LIMMWC34; LIMMWC57; LIPPCS34; LIPPCS5; LIPPCS6; LIPPCS7; LIPPN34; LIPPN5; LIPPN6; LIPPN7; LIPPSD34; LIPPSD6; LIPPSD7	4144	127.08	536	33.69	3216	107.26

TABLE 9. Automated sensitivity analysis of the ExeC horizontal separation.

#	ExeC Horizontal Separation	Timeliness	Total ExeC/FC Communications
1	5.7	611.60	2169
2	5.4	611.34	2174
3	5.4	611.34	2174
4	4.6	611.10	2182
5	4.6	611.10	2182
6	4.5	611.34	2171
7	4.4	610.73	2178
8	4.4	610.73	2178
9	4.0	611.35	2165
10	4.0	611.35	2165
11	4.2	611.06	2167
12	4.2	611.06	2167

elementary sectors), the highlighted solution represents a good compromise between the ''extreme'' solutions. It qualitatively aggregates the sectors of the ACCs according to their peak of traffic, producing nearly uniform workload distributions for controllers and with a limited total workload.

The results shown so far have highlighted that evolutionary optimization can be effectively used to fine tune optimization parameters, also taking into account human-related metrics. In particular, we have shown that, in real-world scenarios, it is possible to minimize the timeliness and/or the number of ExeC/FC communications. By these preliminary results, we have shown that these aspects are mostly affected by the ExeC horizontal separation, the executive controller outbound lookahead, and the planning controller lookahead. From a numerical point of view, the variation of such parameters may be further explored to provide a detailed analysis for the optimization of the average timeliness and the total number of communication from ExeC to FC. We refer to this exploration as *sensitivity analysis*, i.e. a numerical optimization with the goal of assessing what is the actual decision variable which has the largest impact of the metrics.

Our evolutionary optimization architecture can be also used to carry out this kind of sensitivity analysis, in an automated way. In particular, we have run a set of optimizations in which we have used only one variable for the exploration of the search space. This assessment allows to determine what is the decision variable which has the largest impact on the results, therefore showing what should be the best-suited variables to be used when optimizing an ATM scenario. For this experiment, we have used the ''very-high complexity'' traffic on July 6, 2016 in the time slot from 12 pm to 2 pm, considering the ACCs of Milan and Padua.

Tables 9–12 report the results obtained when optimizing one variable at a time—among ExeC horizontal separation (nautical miles), ExeC Outbound Lookahead (seconds), HP Horizontal Separation (nautical miles), and PC Lookahead (seconds)—while using as performance metrics the overall timeliness (seconds) and the number of ExeC/FC communications. In all optimizations the evolutionary architecture is able to find Pareto-optima, but the most interesting

TABLE 10. Automated sensitivity analysis of the ExeC outbound lookahead.

#	ExeC Outbound Lookahead	Timeliness	Total ExeC/FC Communications
1	378.4	611.51	2192
2	378.4	611.51	2192
3	367.7	614.90	2189
4	320.0	613.80	2187
5	291.2	614.90	2167
6	291.2	614.90	2167
7	261.5	617.94	2141
8	261.5	614.90	2141
9	244.1	614.90	2141
10	220.2	617.54	2133
11	206.7	617.94	2121
12	206.7	617.94	2121

TABLE 11. Automated sensitivity analysis of the PC horizontal separation.

#	PC Horizontal Separation	Timeliness	Total ExeC/FC Communications
1	7.5	616.02	2164
2	6.9	616.02	2164
3	6.9	616.02	2164
4	7.5	616.02	2164
5	6.9	616.02	2164
6	6.9	616.02	2164
7	7.5	616.02	2164
8	6.9	616.02	2164
9	6.9	616.02	2164
10	7.5	616.02	2164
11	6.9	616.02	2164
12	6.9	616.02	2164

TABLE 12. Automated sensitivity analysis of the PC lookahead.

#	PC Lookahead	Timeliness	Total ExeC/FC Communications
1	849.5	613.13	2186
2	849.5	613.13	2186
3	859.3	612.02	2180
4	859.3	612.02	2180
5	909.1	608.84	2181
6	909.1	608.84	2181
7	936.4	610.22	2172
8	937.2	611.10	2170
9	967.0	608.91	2166
10	975.8	607.30	2167
11	975.8	607.30	2167
12	971.6	609.84	2163

result lies in that some of these optimization variables (e.g., PC horizontal separation) do not affect the metrics. Therefore, this kind of analysis effectively allows to identify what are the best-suited parameters to be dealt with for the optimization of an ATM scenario.

The achieved validation results indicate good agreement between simulated and real flights, with limited deviations for the assessment of performance metrics under evaluation. Indeed, assessment errors are always under low thresholds, such as 4 minutes for flight timeliness and 0.6 flights/10 minutes for sector occupancy. Additionally, all the results have also been validated by ATCOs from Air Navigation Service Providers (ANSPs), such as ENAIRE (Spain) and ENAV (Italy), in a series of dedicated meetings. The optimizations which have been identified by our architecture have been considered satisfactory. In particular, the controllers have found acceptable the behavior of emulated human agents, highlighting that often humans behave in different ways though they are facing the same situation and such variability was, in fact, fairly modelled in our approach.

The achieved results are strictly generalizable since the simulation errors are limited and show the same magnitudes for a given performance metric, without a clear dependence on the sectors and on the type of traffic. Furthermore, the simulation errors appear to be mainly related to real and non-deterministic ATM events (e.g., delays, severe weather conditions, etc.). These events produce deviations of the real traffic flow from the planned one and cannot be predicted by the simulation engine without injecting additional input data which are not usually available during the strategic and pre-tactical phase. Nevertheless, our simulation model effectively represents the socio-technical dynamics of a generic ATM scenario. This is directly applicable to an accurate performance evaluation of an ATM solution using both human-related and trajectory-related metrics, which in turn feed our optimization architecture for automated support to the design of the ATM solution itself.

As a consequence, the proposed simulation system and optimization architecture may be a first attempt at a ‘‘standardized’’ toolset to support ATM change management. Indeed, such a toolset would enable a systematic process for the design and the verification of new concepts in ATM by means of fast-time simulation and optimization, including human behavior assessment.

VII. CONCLUSION AND FUTURE WORK

We presented an ABMS-based evolutionary architecture designed for the evaluation and optimization of ATM systems. Our proposal allows to carry out several differentiated optimizations, involving multiple different optimization variables, and relying on several trajectory- and human-based metrics to assess the goodness of a solution. The optimization is based on an ABM model which allows to simulate both aircraft and humans (ATCOs) involved in the ATM procedures. Experimentally, we have shown that our architecture allows to carry out optimizations of real-world scenarios in a reduced amount of time, thus placing itself as a versatile tool for what-if analysis in the contexts of ATM.

Currently ongoing work entails the identification of proper trade-offs between accuracy and timeliness for real-time applications (during the tactical phase), in order to allow the identification of better-performing operations in a bounded execution time, as an assistive technology to ATCOs. This kind of any-time optimization is currently being tackled by tweaking the time-window and the sectors involved in the simulations in the main loop of the evolutionary algorithm, and also in terms of the selection of a proper subset of the simulated flights.

REFERENCES

[1] J. U. Sesar, "The roadmap for delivering high performing aviation for Europe–European ATM master plan–executive view," SESAR Joint Undertaking, Brussels, Belgium, Tech. Rep. MG-04-15-465-EN-N, 2015.

[2] S. Abar, G. K. Theodoropoulos, P. Lemarinier, and G. M. P. O'Hare, "Agent based modelling and simulation tools: A review of the state-of-art software," *Comput. Sci. Rev.*, vol. 24, pp. 13–33, May 2017.

[3] B. Doerr and T. Jansen, "Theory of evolutionary computation," *Algorithmica*, vol. 59, no. 3, pp. 299–300, Mar. 2011.

[4] R. A. Sarker and T. Ray, *Agent-Based Evolutionary Search* (Adaptation, Learning, and Optimization), vol. 5. Berlin, Germany: Springer, 2010.

[5] J.-B. Gotteland and N. Durand, "Genetic algorithms applied to airport ground traffic optimization," in *Proc. Congr. Evol. Comput. (CEC)*, 2003, pp. 544–551.

[6] J. V. Hansen, "Genetic search methods in air traffic control," *Comput. Oper. Res.*, vol. 31, no. 3, pp. 445–459, Mar. 2004.

[7] X.-B. Hu and E. Di Paolo, "An efficient genetic algorithm with uniform crossover for air traffic control," *Comput. Oper. Res.*, vol. 36, no. 1, pp. 245–259, Jan. 2009.

[8] J. Tang, S. Alam, C. Lokan, and H. A. Abbass, "A multi-objective evolutionary method for dynamic airspace re-sectorization using sectors clipping and similarities," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

[9] L. Yliniemi, A. K. Agogino, and K. Tumer, "Evolutionary agent-based simulation of the introduction of new technologies in air traffic management," in *Proc. Conf. Genetic Evol. Comput. (GECCO)*, 2014, pp. 1215–1222.

[10] M. G. Oladokun and I. A. Odesola, "Household energy consumption and carbon emissions for sustainable cities—A critical review of modelling approaches," *Int. J. Sustain. Built Environ.*, vol. 4, no. 2, pp. 231–247, Dec. 2015.

[11] J. Dugdale, "Human behaviour modelling in complex socio-technical systems: An agent based approach," Université de Grenoble Alpes, Grenoble, France, Tech. Rep. tel-01006225, 2013.

[12] D. Šišlák, M. Pěchouček, P. Volf, D. Pavlíček, J. Samek, V. Mařík, and P. Losiewicz, "AGENTFLY: Towards multi-agent technology in free flight air traffic control," in *Defence Industry Applications of Autonomous Agents and Multi-Agent Systems*. Basel, Switzerland: Birkhäuser, 2008, pp. 73–96.

[13] K. D. Bilimoria, B. Sridhar, S. R. Grabbe, G. B. Chatterji, and K. S. Sheth, "FACET: Future ATM concepts evaluation tool," *Air Traffic Control Quart.*, vol. 9, no. 1, pp. 1–20, Jan. 2001.

[14] C. Bongiorno, S. Miccichè, M. Ducci, and G. Gurtner, "ELSA air traffic simulator: An empirically grounded agent based model for the SESAR scenario," in *Proc. SESAR Innov. Days*, 2015.

[15] R. Herranz, M. Ramirez, N. Alsina, and L. Castelli, "Price-setting auctions for airport slot allocation: A multi-airport case study," in *Proc. 1st 3rd 5th SESAR Innov. Days*, Dec. 2015, pp. 1–8.

[16] C. Ciruelos, A. Arranz, I. Etxebarria, S. Peces, B. Campanelli, P. Fleurquin, V. M. Eguiluz, and J. J. Ramasco, "Modelling delay propagation trees for scheduled flights," in *Proc. 11th USA/Eur. Air Traffic Manage. Res. Develop. Seminar*, 2015, pp. 23–26.

[17] M. Molina, J. Martin, and S. Carrasco, "An agent-based approach for the design of the future European air traffic management system," in *Advances in Practical Applications of Heterogeneous Multi-Agent Systems* (Lecture Notes in Computer Science), vol. 8473, Y. Demazeau, F. Zambonelli, J. M. Corchado, and J. Bajo, Eds. Cham, Switzerland: Springer, 2014, pp. 359–362.

[18] S. H. Stroeve, T. Bosse, H. A. Blom, A. Sharpanskykh, and M. H. Everdij, "Agent-based modelling for analysis of resilience in ATM," in *Proc. SESAR Innov. Days*, 2013.

[19] A. Pasquini, M. Ragosta, S. Silvagni, M.-A. Sujan, E. Rigaud, and E. Hollnagel, "Modelling of automation degradation: A case study," in *Proc. SESAR Innov. Days*, 2013.

[20] H. A. Blom and G. J. Bakker, "Agent-based modelling and simulation of trajectory based operations under very high traffic demand," in *Proc. SESAR Innov. Days*, 2016.

[21] S. Bouarfa, "Agent-based modelling and simulation of safety and resilience in air transportation," Ph.D. dissertation, Technische Universiteit Delft, Delft, The Netherlands, 2015.

[22] E. O. Scott and K. A. De Jong, "Understanding simple asynchronous evolutionary algorithms," in *Proc. 13th ACM Conf. Found. Genetic Algorithms (FOGA)*. New York, NY, USA: ACM Press, 2015, pp. 85–98.

[23] M. Yagoubi, L. Thobois, and M. Schoenauer, "Asynchronous evolutionary multi-objective algorithms with heterogeneous evaluation costs," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 21–28.

[24] S. Wandelt, X. Sun, and V. Gollnick, "SO6C: Compressed trajectories in air traffic management," *Air Traffic Control Quart.*, vol. 22, no. 2, pp. 157–178, Apr. 2014.

[25] G. Gigante, D. Pascarella, M. Sánchez Cidoncha, M. A. Piera, J. J. Ramos, G. Duca, M. Ciaburri, L. M. Braga, and J. L. Muñoz-Gamarra, "A socio-technical holistic ABM simulation framework to assess ATC performance variability," in *Proc. 10th EUROSIM Congr.*, E. Jiménez and J. I. Latorre, Eds. ARGESIM Publisher, 2019, p. 13.

[26] G. Gigante, D. Pascarella, M. S. Cidoncha, M. A. Piera, G. Duca, N. Lettieri, L. M. Braga, and J. L. M. Gamarra, "ATM change management: An evolutionary and agent-based approach," in *Proc. MATEC Web Conf.*, vol. 259, Jan. 2019, p. 02005.

[27] A. Sharpanskykh, "Agent-based modeling and analysis of socio-technical systems," *Cybern. Syst.*, vol. 42, no. 5, pp. 308–323, Jun. 2011.

[28] S. Straussberger, "Monotony in air traffic control-contributing factors and mitigation strategies," in *Proc. EEC*, 1997, p. 467.

[29] B. Hilburn and P. G. A. M. Jorna, "Workload and air traffic control," in *Stress, Workload, and Fatigue*. Mahwah, NJ, USA: Lawrence Erlbaum Associates Publishers, 2001, pp. 384–394.

[30] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Trans. Syst., Man, Cybern. A, Syst. Hum.*, vol. 30, no. 3, pp. 286–297, May 2000.

[31] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 46, no. 1, pp. 50–80, 2004.

[32] S. Shah and M. Bull, "OpenMP," in *Proc. ACM/IEEE Conf. Supercomput.*, 2006, p. 13.

[33] R. L. Helmreich, J. A. Wilhelm, J. R. Klinect, and A. C. Merritt, "Culture, error, and crew resource management," in *Improving Teamwork in Organizations: Applications of Resource Management Training*. Mahwah, NJ, USA: Lawrence Erlbaum Associates Publishers, 2001, pp. 305–331.

[34] *Third Report: Organizing for Safety*, ACSNI Human Factors Study Group, HSE Books, London, U.K., 1993.

[35] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[36] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[37] A. V. Aho, R. Sethi, J. D. Ullman, M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.

[38] M. P. Fourman, "Compaction of symbolic layout using genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms*, 1985, pp. 141–153.

[39] J. T. Richardson, M. R. Palmer, G. E. Liepins, and M. Hilliard, "Some guidelines for genetic algorithms with penalty functions," in *Proc. 3rd Int. Conf. Genetic Algorithms*. San Mateo, CA, USA: Morgan Kaufmann, 1989, pp. 191–197.

[40] J. Horn, "F1.9 multicriterion decision making," *J. Math.*, vol. 97, no. 1, 1997.

[41] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.

[42] X. Olive, "Traffic, a toolbox for processing and analysing air traffic data," *J. Open Source Softw.*, vol. 4, no. 39, p. 1518, Jul. 2019.

[43] J. Sun, "Open aircraft performance modeling: based on an analysis of aircraft surveillance data," Ph.D. dissertation, Delft Univ. Technol., Delft, The Netherlands, 2019.

[44] A.-C. Zăvoianu, E. Lughofer, W. Koppelstätter, G. Weidenholzer, W. Amrhein, and E. P. Klement, "Performance comparison of generational and steady-state asynchronous multi-objective evolutionary algorithms for computationally-intensive problems," *Knowl.-Based Syst.*, vol. 87, pp. 47–60, Oct. 2015.

[45] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Swiss Federal Inst. Technol., Zurich, Switzerland, Tech. Rep. 103, 2001.

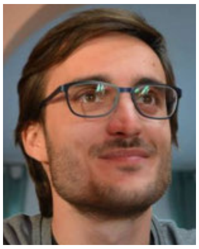
[46] M. Depolli, R. Trobec, and B. Filipič, "Asynchronous master-slave parallelization of differential evolution for multi-objective optimization," *Evol. Comput.*, vol. 21, no. 2, pp. 261–291, May 2013.

[47] W. Gropp, "MPI 3 and beyond: Why MPI is successful and what challenges it faces," in *Recent Advances in the Message Passing Interface* (Lecture Notes in Computer Science), J. Träff, S. Benkner, and J. Dongarra, Eds. Berlin, Germany: Springer, 2012, pp. 1–9.

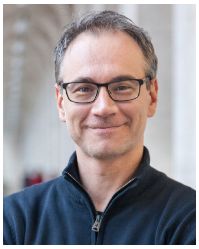
[48] J. G. Vaucher and P. Duval, "A comparison of simulation event list algorithms," *Commun. ACM*, vol. 18, no. 4, pp. 223–230, Apr. 1975.

[49] R. M. Fujimoto, "Performance of time warp under synthetic workloads," in *Proc. Multiconference Distrib. Simulation*, Society for Computer Simulation, 1990, pp. 23–28.

[50] *SESAR Solutions Catalogue*, 2nd ed., SESAR Joint Undertaking, SESAR, Brussels, Belgium, 2017.



ALESSANDRO PELLEGRINI (Member, IEEE) received the Ph.D. degree in computer engineering from the Sapienza University of Rome, in 2014. He worked as a Researcher with many national/international research institutes, such as ISSNOVA, CINI, CINFAI, IRIANC, and BSC. His research interests include simulation on parallel and distributed architectures, a field in which he has more than 70 publications on books, journals, and conference proceedings. His additional competences are on compilers, operating systems, high-performance computing systems, and distributed/concurrent algorithms. He has also successfully contributed to the development of open-source applications currently used in several European research institutes. In 2018, he won the HiPEAC Technology Transfer Award, while, in 2015, he also won the prize for the Best Ph.D. Thesis of the Year from the Sapienza University of Rome.



PIERANGELO DI SANZO received the M.Sc. and Ph.D. degrees in computer engineering from the Sapienza University of Rome. He worked as a Researcher with various research institutes, such as CINI, CINFAI, and ISSNOVA. He is currently a Postdoctoral Researcher with the Department of Computer, Control, and Management Engineering, Sapienza University of Rome. His research interests include the area of performance modeling, evaluation, and optimization of computer systems.

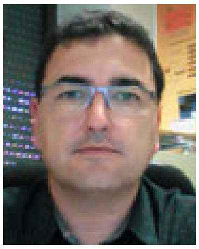
BEATRICE BEVILACQUA received the M.Sc. degree in computer science engineering from the Sapienza University of Rome. During her career, she visited Linkoping University as an Erasmus Exchange Student, and Purdue University as a Visiting Scholar. Her research interest ranges from low-level programming to data-related topics. Her work focuses on developing performance solutions for data-intensive applications, especially in the context of representation learning.

GABRIELLA DUCA received the Ph.D. degree (Hons.) in architecture in 2004.

She is currently a Human Factors Consultant and an Independent Researcher specialized in transdisciplinary application of human factors with a focus on task analysis, process design, human performances, work spaces and interfaces in safety critical contexts, human factors in systems sustainability/efficiency, and human–built environment interaction. She is also the President of the Institute of Sustainable Society and Innovation, where she also leads the Human Factors and Ergonomics Laboratory. She is a member of the HFES Europe Chapter. In 2003, she received the EurErg international qualification as a professional Ergonomist. She was a National Board Member of the SIE Italian Ergonomics Society.

DOMENICO PASCARELLA received the B.S. and M.S. degrees in information engineering from the Seconda Università degli Studi di Napoli University, Italy, in 2007, and the Ph.D. degree in electronics and information engineering from the Department of Industrial and Information Engineering, School of Polytechnic and Basic Sciences, Seconda Università degli Studi di Napoli University, in 2016. Since 2008, he has been a Researcher with the Intelligent Systems, Verification, and Validation Laboratory, Safety and Security Department, CIRA (Italian Aerospace Research Centre), Italy. He is the author of more than 30 articles. His current research interests include modeling of complex and critical systems, distributed intelligent systems, multi-agent systems, and automated planning.

ROBERTO PALUMBO received the M.Sc. and Ph.D. degrees in aerospace engineering from the University of Naples Federico II, in 2004 and 2007, respectively. In 2004, he joined CIRA, Italian Aerospace Research Centre, Capua, Italy, where he is currently a Research Engineer with the Department of On-Board Systems and Air Traffic Management. His research activities are mainly focused on flight systems modeling (for both aeronautics and space), data analysis, and agent-based simulation for air traffic management applications.



JUAN JOSÈ RAMOS (Member, IEEE) received the Ph.D. degree from the Universitat Autònoma de Barcelona (UAB), in 2003. He is currently a Researcher focusing on modeling, simulation, and optimization of dynamic systems, especially in the field of logistics. He is also a Professor with the Department of Telecommunication and System Engineering, UAB, where he is currently the program Director of the European Master in Logistics and Supply Chain Management, UAB. He is a member of LogiSim, a recognized research group on modeling and simulation of complex systems, and a Co-Founder of two spinoff companies. He is a coauthor of four patents/IPR currently in industrial applications. He is also an Expert in air transportation, production technologies/logistics, intelligent transport systems, industrial collaboration, and technology transfer. He was a recipient of the Air Traffic Management Research and Development Seminar Best Paper Award in the Separation Track in 2017.

MIQUEL ÀNGEL PIERA (Member, IEEE) received the degree (Hons.) in computer engineering from the Autonomous University of Barcelona (UAB), Barcelona, Spain, in 1988, the M.Sc. degree in control engineering from the Institute of Science and Technology, The University of Manchester, Manchester, U.K., in 1991, and the Ph.D. degree from UAB, in 1993. He is currently a full-time Professor with the Telecommunications and System Engineering Department, UAB. He is also the former Director of the Technical Innovation Cluster on Aeronautical Management and the former Director of LogiSim (research group on modeling and simulation of complex systems). His main area of expertise is modeling simulation and optimization of manufacturing, logistics, and transport systems, and he has been coordinating several H2020 projects in these areas (AGENT, PARTAKE, and E-PILOTS). He is contributing as the Scientific Advisor to the company ASLOGIC (www.aslogic.es).

GABRIELLA GIGANTE received the Ph.D. degree from the Department of Industrial and Information Engineering, School of Polytechnic and Basic Sciences, Second University of Naples, Italy. She is currently the Head of the Laboratory Intelligent Systems, Verification and Validation, Department of Safety and Security with CIRA. Her current research interests include intelligent systems and software certification aspects, intelligent systems verification and validation, safety aspects in innovative aerial unmanned platforms, and multi-agents systems.