Multi-agent Systems for Air Traffic Conflicts Resolution by Local Speed Regulation

Romaric Breil, Daniel Delahaye
Laboratory in Applied Mathematics, Computer Science and Automatics for Air Transport
National Civil Aviation University
Toulouse, France
E-mail: breil@recherche.enac.fr

Laurent Lapasset
Capgemini Technology Services
Toulouse, France

Éric Féron
Daniel Guggenheim School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332 USA

***Abstract*—Air Traffic Flow Management (ATFM) aims at structuring traffic in order to reduce congestion in airspace. Congestion being linked to aircraft located at the same position at the same time, ATFM organizes traffic in the spatial dimension (e.g. route network) and/or in the time dimension (sequencing and merging in TMA, Miles-in-Trail for en-route airspace).**

The objective of this paper is to develop a methodology that allows the traffic to self-organize in the time dimension when demand is high. This structure disappears when the demand diminishes.

In order to reach this goal, a multi-agent system has been developed. This algorithm regulates aircraft speed in order to reduce the number of conflicts, thus decreases overall traffic complexity, which becomes easier to manage by air traffic controllers. This algorithm was applied on realistic examples.

Air traffic management; multi-agent system; conflict resolution; speed regulation

I. Introduction

Air traffic volume has been constantly increasing during the past decades, and ICAO [1] predicts that the annual number of flights will double in 2030 in comparison to 2013. Air traffic controllers are in charge of ensuring traffic safety and fluidity by temporarily diverting flights from their original trajectory when necessary. In doing so, a minimum separation distance is maintained between all aircraft. This task is known as conflict detection and resolution. It is increasingly perceived that the present centralized way of managing traffic cannot scale up anymore. In order to deal with traffic growth, major research programs around the world, such as SESAR (Single European Sky ATM Research) and NextGen (US Next Generation Air Transportation System) consider automating some tasks previously done by controllers, allowing them to manage more flights simultaneously. In a more daring effort, decentralized flow management, whereby traffic flow management is delegated to individual aircraft, is also an option.

A. Air Traffic Management

The current air traffic system is structured [2] on a route network whose vertices are geographical positions called waypoints, through which an aircraft shall go. Each waypoint is identified by a name. Before departure, the pilot or the airline dispatcher is to submit a flight plan to the civil aviation authority, containing information related to the flight (route, departure and arrival airports, etc.). This flight route is defined by a set of waypoints identified by their names. Once airborne, however, modifications to the flight plan may be initiated by the flight crew or air traffic control, depending on local traffic and weather conditions.

The airspace of every country is usually divided into several sectors. Each sector is managed by a team of air traffic controllers in charge of trajectory planning, collision avoidance management and of communicating with aircraft. Controllers can only take charge of a limited number of flights simultaneously. Sectors and route networks are designed in such a way that controllers' workload is reduced; they have only a limited number of flights to manage simultaneously in their sectors, each of them following a predefined route. They have to keep separation distances between each pair of aircraft above given threshold values: in cruise phase, a conflict occurs when two aircraft are separated by less than 5 nautical miles (1 NM = 1.852 km) horizontally and 1,000 feet vertically (1 ft = 30.48 cm). When a conflict is foreseen, the controller requests that one or both pilots execute a maneuver, usually temporarily changing the heading or altitude to increase separation, before returning to initial flight path.

The air traffic is constantly increasing and the current air traffic structure is reaching its maximum capacity. To cope with this situation, parts of the control process could be delegated to automatic algorithms, like conflict detection and resolution or other traffic management tasks.

Aircraft are increasingly capable of communicating full estimated 4D trajectories over the next minutes by means of messages. These data can be transmitted by using Automatic Dependent Surveillance – Contract (ADS-C) [3, 2.2.6]. These messages are exchanged following a request-answer protocol. Usually, a ground station sends a request to a specific aircraft, which sends back a data frame containing aircraft identifier, position, speed and the predicted route composed of a set of 4D positions (3D + time). Information carried by ADS-C is more accurate than radar positioning since an aircraft uses GPS to get its position. Ground stations receiving these messages are able to provide controllers with accurate representation of air traffic. This is well adapted to airspace where radar coverage is not available (e.g. oceanic airspace).

Free flight is an air traffic management concept developed in the U.S. that enables aircraft to choose their path in low traffic zones with more freedom, by ignoring route networks. The implementation is currently studied by the NextGen research program. In Europe a similar concept was developed by the SESAR project; the Free Route Airspace is already deployed in some areas. These zones are managed or not by air traffic controllers. In the second alternative, automatic separation assurance systems would benefit from aircraft information exchange systems.

B. Algorithms Solving ATM Problems

Many studies have already been performed to design automatic conflicts resolution methods. Some are referenced in [4]. Most of those methods try to reproduce the way controllers regulate traffic by changing aircraft heading for a short period. According to air traffic controllers, those algorithms can interfere with their own decisions since the controllers and the algorithms take the same kind of decision in the same controlled areas [5].

In 2004, a new way to solve this problem was proposed as a part of the project ERASMUS (En Route Air traffic Soft Management Ultimate System). According to Villiers [5], instead of trying to reproduce what controllers are doing, those algorithms should help them by removing a part of the conflicts before they appear by slightly changing aircraft speed for instance. This type of automated system organizes traffic in order to create a favorable traffic situation, more easily managed by controllers, and not interfering with their own decisions. This concept was validated with technical experiments [6] and human factors studies [7]. The speed regulation method was implemented as a genetic algorithm.

Since problems encountered in Air Traffic Management (ATM) are highly combinatorial, deterministic optimization methods tend to become inefficient when dealing with real traffic scenarios (from hundreds to thousands of aircraft). To overcome this situation, heuristics have been used in several research works, giving approximate but good results in a time horizon compatible with operational constraints [8].

Multi-agent systems can be used to develop heuristic algorithms, and have already been applied to air traffic management problems. Some studies focus on traffic regulation in free flight (or Free Route) zones. Aircraft flying in free flight areas must automatically be able to find conflict-free trajectories, respecting the required distance separation between aircraft. This problem can be solved by multi-agent systems, such as the one that was developed by Wollkind, Valasek and Ioerger [9], or the one developed by Sislak, Volf and Pechoucek [10]. Those algorithms solve conflicts using maneuvers such as heading and flight level changes. As shown by Villiers, this regulation method can interfere with controllers' decision process.

The algorithm described in this paper aims at implementing the traffic regulation method of the project ERASMUS as a multi-agent system. As detailed in Section I-C, this type of algorithm has several advantages compared to global optimization methods used in ERASMUS.

C. Multi-Agent Systems

Multi-agent systems have been used to solve many problems in operations research, like regulation of urban transportation networks [11], design of mechanical systems [12], or path-finding problems [13]. This paradigm is often regarded as a kind of distributed artificial intelligence. Multi-agent systems are made of autonomous agents interacting among themselves and with their environment [14]. Usually, agents have a limited perception of environment and they partially know the internal state of their neighbors, via message exchanges. Their behavior can either be simple (whereby reactive agents are only influenced by environmental changes) or complex (whereby cognitive agents try to fulfill an objective).

Self-organization is a key aspect of multi-agent systems. If the rules that direct agents are carefully chosen, a complex behavior can emerge at the system level from local interactions and behavior of agents. When multi-agent systems are used to solve operations research problems, a carefully chosen set of agent behaviors can help to find an overall solution to the problem (system level) by only using local rules (agents level).

Multi-agent systems can be implemented either within a computer simulation, or as a physical system that is composed of robots that are able to communicate and to interact with their environment. When agents are implemented within a computer simulation, computations of agents can be done in parallel, exploiting modern hardware architectures (multi-core processors, computations on graphic card). A multi-agent system can also run on a cluster of computers.

Those systems have several advantages compared to centralized decision methods. When correctly designed, they exhibit a good resilience when facing disruptive events [15]. Agents try to fulfill a goal and act in order to become closer to this objective. When they are confronted to local perturbations in their environment, they adapt their actions to take those changes into account, enabling the system to get back to a new stable state. Since decisions are decentralized at agents level, the failure of an agent will not impact the whole system. In centralized decision processes, a central regulation entity failure may prevent the system to work. In the field of information technologies, such a central point would be defined as a Single Point of Failure (SPOF).

Applied to air traffic management, the implementation of new onboard collaborative decision processes can be done progressively, whereby equipped aircraft cooperate among themselves and are given more freedom in their decisions than non-equipped aircraft (for example by constraining the latter to follow rigid corridors).

Even if current technology like ADS-C does not allow aircraft to really exchange data about trajectories directly from aircraft to aircraft, these data can be collected by ground stations. Traffic regulation algorithms like the one described in this article can then compute speed changes, which can then be sent back to real aircraft. The distributed aspect of a multi-agent approach to the management of traffic is somewhat diminished since aircraft do not take decisions themselves. Yet, other advantages are preserved, like resilience and general performances.

This article describes a conflict resolution algorithm based on aircraft speed self-regulation. This implementation of the ERASMUS concept takes benefit from the use of multi-agent systems. Section II details the algorithm. Section III describes various scenarios aiming at validating this algorithm.

II. Aircraft Speed Self-Regulation for Conflict Resolution

A. Hypotheses

The algorithm described in this section regulates aircraft in cruise phase. Their altitude is supposed to be constant.

Each flight has a preferred cruise speed depending on general aircraft performances and airlines preferences. Airlines can give priority to reducing fuel cost by reducing aircraft speed, or to the reduction of crew costs by increasing speed, which also increases fuel consumption. This setting is adjusted by using a value called Cost Index (CI), which is the ratio of the cost of flight time (including crew costs) to the cost of fuel.

Let v be the current speed of an aircraft and v_{opt} its optimal speed. In order to be inserted into a route network, aircraft may have to choose a speed v different from v_{opt} within a given speed interval. A lower bound $v_{min} = v_{opt} - 6\%$ will allow to insert this aircraft into a flow without dramatically increasing fuel consumption [16]. A speed interval of $[v_{opt} - 6\%, v_{opt} + 3\%]$ is a common choice for speed regulation in en route airspace [17].

The Airbus A320 and A380 are representative of the aircraft that are able to fly at 36,000 ft and above, keeping in mind that the A320 belongs to the slowest aircraft of this category, and that the A380, to the fastest (alongside the Boeing 777, for instance). In our simulation, optimal speed of aircraft are randomly chosen in the interval [447 knots, 487 knots] (1 kt = 1 NM/s), which are the optimal speeds of an A320 [18] and an A380 [19]. Moreover, finding a solution is not guaranteed when flights are following each others — for instance in the Miles-in-Trail scenario described in Section III-A — if the maximum speed of the slowest aircraft is lower than the minimum speed of the fastest one.

Aircraft acceleration and deceleration are fixed to $\pm 4{,}000\,\mathrm{NM/h^2}$ ($\pm 0.572\,\mathrm{m/s^2}$) for all aircraft. The standard turn rate [20, PCG S–6] of 3°/s is used so that a complete 360° turn is done in 2 minutes.

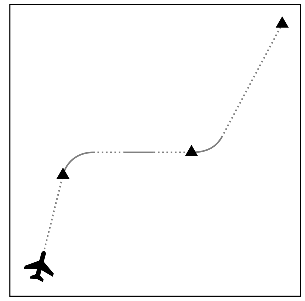

Figure 1. In the algorithm described in Section II, a trajectory is a curve composed of segments (constant speed as dashed line, acceleration as plain line) and of arcs of circles. Aircraft flies from a waypoint to another (triangles).

In a general traffic situation or in free route scenarios, some conflicts cannot be solved only by using speed control, as in head-on encounters, for instance. The aim of this algorithm is to simplify the traffic by doing subliminal speed changes in order to help controllers. The algorithm reduces the number of conflicts and delegates the remaining ones to air traffic controllers.

Thus, these conflicts need additional maneuvers to be solved. In the current implementation of the algorithm, these maneuvers are not implemented: the algorithm only minimizes the number and duration of conflicts and does not try to solve all of them.

B. Algorithm

In this multi-agent system, aircraft are agents exchanging ADS-C messages containing estimated 4D trajectories. A trajectory is stored and exchanged as a sequence of arcs that can be straight segments (aircraft flying at a constant speed, accelerating, or decelerating) and arcs of circles (aircraft turning), as shown in Fig. 1. This curve is differentiable at least once everywhere.

This curve is built from a flight plan defined by a set of waypoints. An aircraft has to fly above each waypoint (Fig. 1). After an aircraft reaches a waypoint, it turns in order to head towards the next one. Speed changes are planed at given times (Section II-B2), and are applied in straight segments.

This multi-agent system (Fig. 2) is timed by a global clock. Each tick corresponds to a second in the simulation. All agents are synchronized: at the end of an iteration, agents drop off messages into the mailbox. When the next iteration begins, those messages are delivered to addressee agents. Therefore, even if agents processes are run asynchronously, agents work logically in parallel. This choice helps the overall system to avoid problems related to sequence order. For performance reasons, agents are run in parallel, using multiple threads.

During the lifecycle of an agent, a sequence of three steps is repeated at every iteration of the multi-agent system until the agent is removed. The perception step allows agents to receive ADS-C messages and refresh internal representation of airspace. During the decision step, aircraft plan speed changes

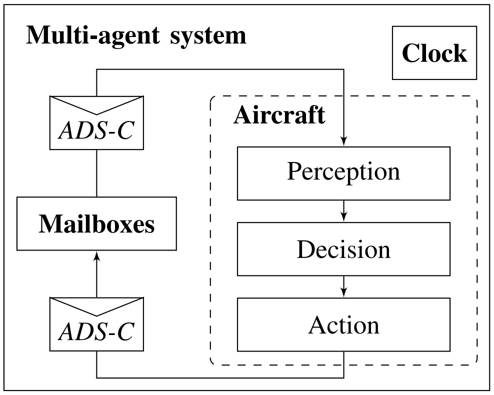

Figure 2. Multi-agent system lifecycle.

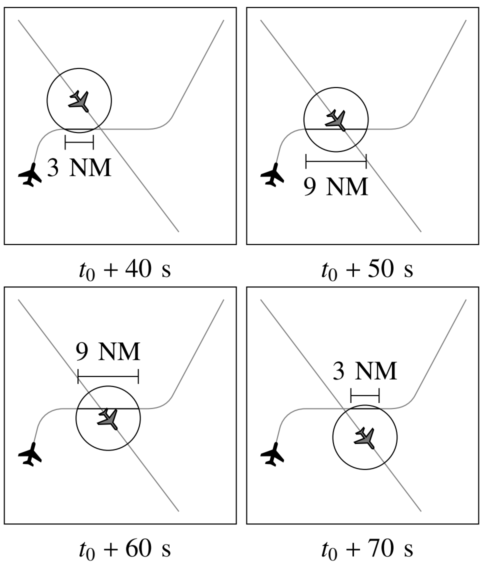

Figure 3. The black aircraft detects potential conflict with the gray aircraft for each time sample in the next minutes. A potential conflict is the intersection between separation distance circle of the gray aircraft and path of the black aircraft.

on the basis of this internal representation. In the action step, agents update their position using updated speed, and broadcast an ADS-C message.

1) Perception: Each aircraft agent first receives messages from its neighbors, from which it extracts 4D trajectories. In order to detect conflicts, the aircraft then samples other aircraft trajectories to get their predicted position every 10 seconds during 20 minutes (Fig. 3).

a) Conflicts Detection: A conflict occurs when the distance between two aircraft becomes smaller than 5 NM on a horizontal level. When an aircraft follows a path, its future positions are defined by its own current position and by its speed changes. Since speed is to be chosen during the decision phase, all potential conflicts need to be detected, regardless of its own speed. Then, for each intruder predicted position, the algorithm searches for possible intersections between its own path and a circle of 5 NM centered on the intruder predicted position (Fig. 3). Thus, whatever speed changes the aircraft

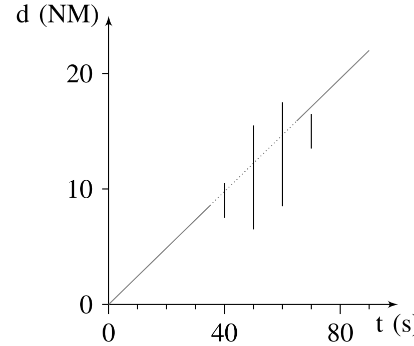

Figure 4. Representation of the 2D trajectory of the black aircraft (distance as a function of time) and intrusions of the gray neighbor in the black aircraft's path at each time sample (vertical segments) of the Fig. 3. A conflict will occur if the black aircraft maintains its initial speed: the curve representing the black aircraft's trajectory intersects with the vertical segments that represents potential conflicts.

will choose, it will always manage to detect conflicts.

b) Internal Representation of Own Trajectory: Since an aircraft strictly follows its path, the decision process only modifies speed. To manipulate a trajectory defined by a 4D curve is not necessary: this 4D trajectory can be simplified by using the traveled distance over the route as a function of time, by integrating the instant speed as a function of time. In other words, the aircraft trajectory can be represented as a 2D curve defined by the arc length as a function of time. This curve is then projected into a 2D space (Fig. 4).

Conflicts detected during the conflicts detection step (Section II-B1a) are also projected in this space. Each potential intrusion extracted from the sampled neighbor's trajectory is projected according to its position along the path (Fig. 3). The portion of the path intersecting with the circle of 5 NM centered on the intruder's position gives an interval of positions forbidden to the agent (Fig. 4).

2) Decision: Using conflicts projected in the 2D representation (displayed in Fig. 4), aircraft can plan speed changes. The goal is to avoid conflicts by way of speed changes. This goal is achieved by using a decision tree where for each time sample, the aircraft can maintain its speed, accelerate or decelerate.

A time step of 5 seconds was chosen. For each time sample, three choices are tested: cruise, acceleration or deceleration at the maximum rate (Fig. 5). Then at each time step and for each possible decision, the time of the first conflict is computed. The problem is solved using a greedy algorithm that locally maximizes the time of the first conflict.

As in the pseudo-code in Appendix A, the set of decisions D is iteratively constructed. For each time step, acceleration (acc), deceleration (dec) and cruise (cr) choices are tested: the time when the first conflict occurs is stored in the variable cflTime. Then, the algorithm looks for the decision leading to the latest conflict date. If speed constraints are respected (checked by the function isValid()), this decision is accepted.

Using a greedy algorithm makes possible to get good results (Section III) after a short period of computation. Yet, since a

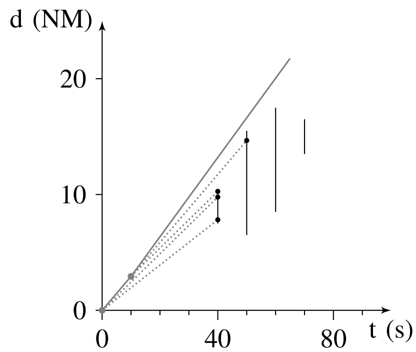

Figure 5. Exploration of the decision tree. For each time step, 3 choices are tested: to accelerate, to cruise, to decelerate. The choice leading to the longest conflict-free trajectory is applied (in this case, it accelerates two times at the beginning, then it cruises).

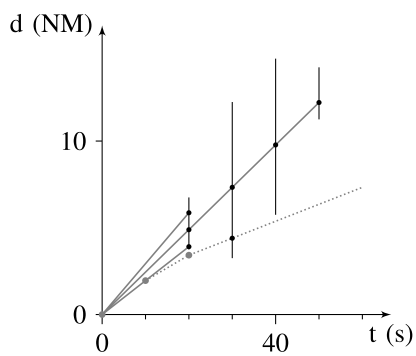

Figure 6. The conflict is unsolvable because of the objective function that tries to maximize the delay before the first conflict; in the first time sample, all the possible choices (to accelerate, to cruise or to decelerate) lead to a conflict at $t = 20$ s. In this case, the aircraft chooses to cruise, which leads to a severe conflict lasting 40 seconds. An alternative objective function that minimizes the duration of a conflict can choose to decelerate three times in order to cause a less severe conflict lasting only 10 s (dashed segments).

greedy algorithm is only a local optimization process, it may find a local optimum and be unable to find an existing conflict-free solution, as in Fig. 6. Therefore, some conflicts cannot be solved. The choice of this method is a compromise between the computation time and the quality of the results.

Results given by the greedy algorithm can nonetheless be refined by the addition of intermediate decisions at each time step (e.g. accelerations and decelerations of $\pm 2{,}000\,\mathrm{NM/h^2}$ and $\pm 4{,}000\,\mathrm{NM/h^2}$), but at the cost of longer computation times.

3) Action: In the action step, the set of decisions D computed by the algorithm of Section II-B2 is used to generate the new 4D trajectory, which is communicated to neighbors using an ADS-C message.

Since all agents apply the same decision process iteratively, the system should converge to a stable state in which conflicts are solved. When aircraft follow the same route, they fly at similar speeds in order to avoid conflicts. When they follow intersecting routes, the separation distance between aircraft of each route becomes similar in order to allow them to cross alternatively the intersection point.

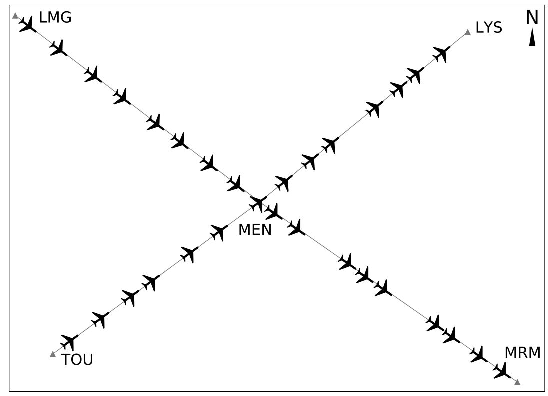

Figure 7. Route network used to simulate the intersection of two flows.

In order to validate the algorithm, various traffic scenarios were tested. A description of the scenarios and some results can be found in Section III.

III. Test Scenarios

A. *Miles-in-Trail*

In order to validate our algorithm, our first experiment is related to the management of an intersection of two Miles-in-Trail (MIT) traffic flows. Miles-in-Trail [21] is a method used by controllers to reduce air traffic complexity. When flight density increases in a given area, traffic is structured into flows of aircraft following the same path. Flights are separated by a given distance (for example 20 NM) and their speed is regulated. This manner of structuring creates queues of aircraft. Those queues are easier to perceive and to manage by controllers. Their job is then to monitor inter-aircraft spacing and to apply speed control whenever it appears to be necessary.

Miles-in-Trail also makes merging or crossing of flows easier. Aircraft must be separated by at least 5 NM, which defines the maximum aircraft density in a flow: when two flows merge, each one must apply a 10 NM Miles-in-Trail so that aircraft alternatively come from the first and the second flow. The same reasoning is applied to crossing flows (Fig. 7) where aircraft must be separated by more than 10 NM, depending on the angle between flows. The exact separation distance can be computed using the method described in [22, Lemma 1], but the algorithm detailed in this article is able to compute approximated values.

This first scenario is an attempt to reproduce the Miles-in-Trail traffic structure at the intersection of two flows. In this simulation, aircraft fly along two crossing routes (Fig. 7). They receive messages from the other agents containing planned 4D trajectories and regulate their speed to avoid conflicts.

The route network used in this simulation contains two intersecting paths defined by 5 waypoints of French airspace: the first path is composed of the waypoints LMG, MEN and MRM, and the second one by TOU, MEN and LYS. The distances between waypoints are: 116 NM for the segment LMG-MEN, 119 NM for MEN-MRM, 97 NM for TOU-MEN

Table I
PERFORMANCES OF THE ALGORITHM FOR THE MILES-IN-TRAIL SCENARIO (AVERAGE VALUES OF 10 RUNS).

Average time interval	**Decision process**	**Number of aircraft**	**Conflicts**	**Solved conflicts (%)**
140 s	Disabled	51.2	16.8	
	Enabled	51.1	2.3	86.3
	Disabled for 10 % of aircraft	51.3	2.7	83.9
110 s	Disabled	65	27	
	Enabled	65	6.9	74.4
	Disabled for 10 % of aircraft	65.8	10.7	60.4

and 105 NM for MEN-LYS. Aircraft are generated randomly at one of the western positions (LMG or MEN). The arrival rate of aircraft along each route follows a Poisson distribution which is considered to be a valid approximation for air traffic flows [23, III–C].

Two sets of three scenarios were tested. Aircraft were generated according to a Poisson point process on each route, and flying during 1 hour. Each set of scenarios corresponds to a different average time interval between the generation of two aircraft: 140 and 110 seconds. The number of aircraft generated on each route per hour is given by the related parameter λ of the Poisson process, being respectively $25.7\,\mathrm{h}^{-1}$ ($1/\lambda$ = 140 s) and $32.7\,\mathrm{h}^{-1}$ ($1/\lambda$ = 110 s). The latter value is the minimum theoretical interval, considering an average optimal speed of 467 kt and a minimal separation distance between aircraft on the same route of 14.14 NM: when aircraft of each route are separated by this distance, the minimum distance between two aircraft flying along perpendicular routes becomes 5 NM [22, Lemma 1]. For each set, three scenarios were tested: with regulation, another without, and a third one where 10 % of aircraft were not regulated (aircraft decisions process were disabled). Aircraft optimal speeds were chosen randomly within [447 kt, 497 kt].

During simulations, distances between all pairs of aircraft were monitored. Each time aircraft were separated by less than 5 NM, a conflict was registered (a single record for the duration of the conflict).

Results of the six scenarios are shown in Table I. Each scenario was run 10 times, values in the table are the average of these 10 runs. When $1/\lambda$ = 140 s, 86 % of conflicts are solved when the decision process is enabled compared to when it is disabled, decreasing from 16.8 conflicts to 2.3 on average. Thus, with regulation, not all conflicts could be solved. The decision process is a greedy algorithm that searches for a local optimum maximizing the time before the first conflict. This algorithm is faster than a global optimization method but is not meant to find a global optimum that would lead to an existing conflict-free solution.

When $1/\lambda$ = 110 s, 74 % of conflicts are solved, decreasing from 27 conflicts to 6.9. This results indicates that the algorithm becomes less efficient as the initial inter-aircraft distance approaches the maximum traffic capacity of 65 aircraft per hour. This loss of efficiency comes from the traffic generation process: the λ value is the average time between two aircraft generations. Thus some aircraft are generated at closer positions than others. Generating two aircraft too close to each other prevents the algorithm to find a conflict-free solution.

So as to validate the resilience of the algorithm to disruptive events, a third scenario was added to each set of scenarios with 10 % of non-cooperative aircraft by disabling their decision process. For the first set, with an average interval of generations of 140 s, results are similar when every aircraft is cooperative and when 10 % of them are non-cooperative, with an average number of unsolved conflicts increasing from 2.3 to 2.7. As long as cooperative aircraft know the estimated trajectories of their non-cooperative neighbors, they will be able to avoid conflicts, because non-cooperative aircraft are taken into account as a constraint in the decision process of cooperative aircraft.

For the second set of scenarios, with average interval of generations of 110 s, the average number of unsolved conflicts increases from 6.9 to 10.7. In other words, the number of solved conflicts decrease from 74 % to 60 %. In this second scenario, aircraft are generated every 110 s, which is the maximum theoretical capacity of the routes. As previously mentioned, the aircraft generation process can produce situations where aircraft are in an unsolvable conflict as soon as they are generated. Non-cooperative aircraft can accentuate this problem. Some conflicts require that all involved aircraft cooperate to be solved, and the solution cannot be found if only one aircraft maneuvers. Moreover, some conflicts that can be avoided occur because none of the involved aircraft takes decisions.

B. Miles-in-Trail Capacity Increase with Crossing Route Topology

In order to increase the capacity at the intersections of a route network in terms of number of aircraft per hour, Yoo and Devasia [24] developed a route topology allowing to put aircraft on parallel tracks to enable them to cross the intersection. In theory, this method allows to reach the maximum capacity of each route which is one aircraft every 5 NM. As shown in Fig. 8, this topology first divides the incoming aircraft flow in three parallel tracks separated by 5 NM laterally. Aircraft then pass the intersection. Finally, the three sub-flows are merged into a single outgoing flow.

A new behavior was added to aircraft agents in Section II, as they can choose one path among three alternatives, as shown in Fig. 8. Aircraft applies the algorithm described in Section II-B2 to each choice, then selects the path leading to the longest conflict-free trajectory.

To validate this method, a new set of three scenarios was tested. Aircraft were generated according to a Poisson point process on each route, and flying during 1 hour. The average time interval was fixed to 110 seconds, as in the first set of scenarios in Section III-A. In the first scenario, the aircraft decision process was disabled. In the second, it was enabled. In

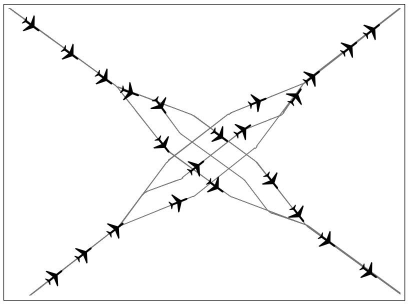

Figure 8. 3-ways route topology allowing to increase the capacity of the route network depicted in Fig. 7. Parallel routes are at a distance of 5 NM.

Table II
PERFORMANCES OF THE ALGORITHM FOR THE MILES-IN-TRAIL WITH CROSSING TOPOLOGY SCENARIO (AVERAGE VALUES OF 10 RUNS).

Average time interval	**Decision process**	**Number of aircraft**	**Conflicts**	**Solved conflicts (%)**
110 s	Disabled	65.2	26.5	
	Enabled	65.1	1.5	94.3
	Disabled for 10 % of aircraft	64.6	2.9	89.1

the third, the decision process was disabled for 10 % of aircraft in order to test the resilience of the algorithm to disruptive events. Results are shown in Table II.

When the decision process is enabled, 94 % of conflicts are solved compared to when the decision process is disabled (1.5 conflicts on average are not solved when enabled, 26.5 when disabled). For the last scenario, while 10 % of aircraft do not cooperate, solved conflicts decrease to 89.1 % (2.9 remaining conflicts).

These results indicate that the capacity of a Miles-in-Trail route network can be increased using an alternative route network at the intersections. In the scenarios described in Section III-A, only 2 unsolved conflicts remain while one aircraft is generated every 140 s on average. In this scenario, this capacity is increased by 27 %, up to one aircraft every 110 s, without increasing the number of unsolved conflicts. It means that the part of the controllers' workload caused by the number of conflicts does not increase while using the 3-ways route topology described in this section.

C. Traffic Based on Actual Flight Plans

Another set of scenarios was tested to validate the algorithm on actual flight plans of aircraft flying over France during 10 hours (from 4 AM to 2 PM), as shown in Fig. 9. Here again, the conflict resolution was disabled in the first scenario, then enabled in the second. Only the flights at 37,000 ft have been analyzed since this altitude contains the most flights, 465 in this case.

Since the route network is more complex and more dense over France than in the Miles-in-Trail network shown in Fig. 7, conflicts can be harder to solve. The small interval of admissible speeds can be insufficient to maintain separation

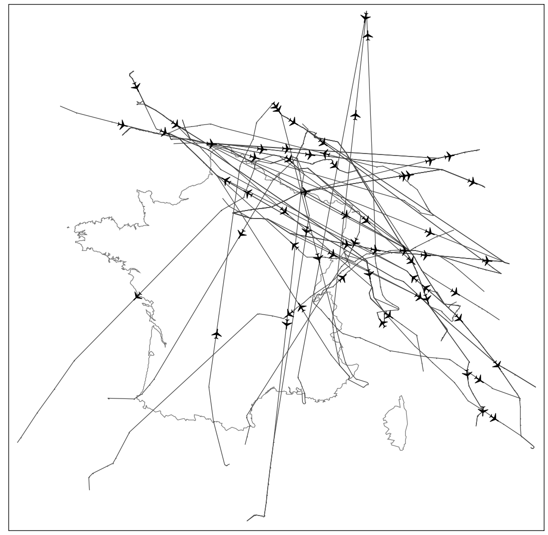

Figure 9. Traffic scenario using actual flight plans, described in Section II. This picture displays flight plans of aircraft flying at 12:00.

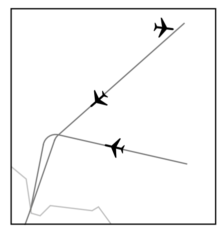

Figure 10. In the scenario based on real flight plans, aircraft are sometimes starting at too close positions and no speed regulation can solve this conflict.

Table III
PERFORMANCES OF THE ALGORITHM FOR THE REAL TRAFFIC SCENARIO (AVERAGE VALUES OF 10 RUNS).

Decision process	**Number of aircraft**	**Conflicts**	**Solved conflicts (%)**
Disabled	465	444	
Enabled	465	256.5	42.2
Disabled for 10 % of aircraft	465	275.9	37.9

in all circumstances. In some situations, aircraft start from positions that lead to an unsolvable conflict (see Fig. 10). Furthermore, speed regulation alone cannot solve face-to-face conflicts. Nevertheless, this algorithm is able to solve 42 % of the 444 conflicts initially present, leaving in the mean 256.5 unsolved conflicts. Results are shown in Table III.

In the third scenario the decision process is disabled for 10 % of aircraft. Then the average number of conflicts slightly increases from 256.5 to 275.9 (−4 % of solved conflicts). This value indicates that this multi-agent system is resilient to disruptions, since the cooperative agents take the non-cooperative ones into account in order to include them as constraints into the decision process.

IV. Conclusion

The algorithm described in this article has several advantages related to the usage of multi-agent systems. It is able to solve up to 86 % of the conflicts, is resilient to perturbations like non-cooperative agents, and could eventually be implemented on board, removing the need to rely on ground equipment.

But this algorithm is not able to solve all conflicts, and is not meant to do so. To avoid all conflicts, aircraft have to execute other types of maneuvers in addition to speed regulation, like heading changes. Nowadays, these decisions are taken by air traffic controllers and are not planned to be delegated to algorithms in human-controlled airspace.

However, some of our recent experiments indicate that allowing aircraft to delay departure time or change flight level helps the algorithm to remove all the remaining conflicts. Our next research work will aim at reducing the number of conflicts by adding new behaviors to the multi-agent system.

We will also implement a global optimization method to solve the problem described in this article. It will allow to measure the difference in terms of computation time and result optimality between results returned by this multi-agent system and by a global optimization method, often used to solve similar problems.

Thanks to the ability of multi-agent systems to integrate non-cooperative agents and to recover from disruptive events, they offer a good framework to mix human-controlled traffic with an automated one. Eventually, these algorithms could then collaborate with humans in air traffic management applications, because they work over different scales, time and space-wise.

Appendix
Algorithm of the Decision Process

choices $\leftarrow \{\text{cr}, \text{acc}, \text{dec}\}$
$D \leftarrow \{\}$
for $i \leftarrow t_0$ **to** t_{end} **step** Δt **do**
    bestChoice $\leftarrow$ cr
    bestCflTime $\leftarrow i$
    for c **in** choices **do**
        cflTime $\leftarrow$ getFirstConflict($D \cup \{c\}$)
        if isValid($D \cup \{c\}$) & cflTime > bestCflTime **then**
            bestChoice $\leftarrow c$
            bestCflTime $\leftarrow$ cflTime
        end if
    end for
    $D \leftarrow D \cup \{\text{bestChoice}\}$
end for

References

[1] International Civil Aviation Organization, *Global Air Transport Outlook to 2030 and Trends to 2040*, ser. ICAO circular. International Civil Aviation Organization, 2013.

[2] D. Isaacson, A. Sadovski, and D. Davis, "Tactical scheduling for precision air traffic operations: Past research and current problems," *Journal of Aerospace Information Systems*, vol. 11, no. 4, pp. 234–257, 2014.

[3] *Global Operational Data Link Document (GOLD)*, http://www.icao.int/APAC/Documents/edocs/GOLD_2Edition.pdf, International Civil Aviation Organization, apr 2013.

[4] J. Kuchar and L. Yang, "A review of conflict detection and resolution modeling methods," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 1, no. 4, pp. 179–189, Dec 2000.

[5] J. Villiers, *Automatisation du contrôle de la circulation aérienne: "ERASMUS", une voie conviviale pour franchir le mur de la capacité*, ser. Etudes & documents - I.T.A. ITA, 2004. [Online]. Available: https://books.google.fr/books?id=FXuFQwAACAAJ

[6] D. Bonini, C. Dupré, and G. Granger, "How erasmus can support an increase in capacity in 2020," in *Proceedings of the 7th International Conference on Computing, Communications and Control Technologies: CCCT*, 2009.

[7] G. Chaloulos, E. Crück, and J. Lygeros, "A simulation based study of subliminal control for air traffic management," *Transportation research part C: Emerging technologies*, vol. 18, no. 6, pp. 963–974, 2010.

[8] D. Delahaye, N. Durand, J.-M. Alliot, and M. Schoenauer, "Genetic algorithms for Air Traffic Control systems," in *IFORS 1996, 14th Triennal Conference of the International Federation of Operational Research Sociaties*, Vancouver, Canada, Jul. 1996. [Online]. Available: https://hal-enac.archives-ouvertes.fr/hal-00938864

[9] S. Wollkind, J. Valasek, and T. R. Ioerger, "Automated conflict resolution for air traffic management using cooperative multiagent negotiation," in *AIAA guidance, navigation, and control conference*, 2004, pp. 1–11.

[10] D. Sislak, P. Volf, and M. Pechoucek, "Agent-based cooperative decentralized airplane-collision avoidance," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 36–46, March 2011.

[11] F. Balbo and S. Pinson, "An agent oriented approach to transportation regulation support systems," in *Proceedings of the 5th Workshop in Agent in Traffic and Transport*, 2008, pp. 225–242.

[12] D. Capera, M.-P. Gleizes, and P. Glize, "Self-organizing agents for mechanical design," in *Engineering Self-Organising Systems*. Springer, 2004, pp. 169–185.

[13] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *Computational Intelligence Magazine, IEEE*, vol. 1, no. 4, pp. 28–39, Nov 2006.

[14] J. Ferber, *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley Reading, 1999, vol. 1.

[15] C. Rieger, K. Moore, and T. Baldwin, "Resilient control systems: A multi-agent dynamic systems perspective," in *Electro/Information Technology (EIT), 2013 IEEE International Conference on*, May 2013, pp. 1–16.

[16] L. Delgado and X. Prats, "Fuel consumption assessment for speed variation concepts during the cruise phase," in *Proceedings of the Conference on Air Traffic Management (ATM) Economics*, 2009.

[17] P. Averty, B. Johansson, J. Wise, and C. Capsie, "Could erasmus speed adjustments be identifiable by air traffic controllers?" in *Proceedings of the VII USA/Europe Air Traffic Management R&D seminar, Barcelona*, jul 2007.

[18] Air Canada, "Our fleet: Airbus a320-200 (320)," http://www.aircanada.com/en/about/fleet/a320-200xm.html, accessed: 2015-08-25.

[19] Emirates, "Emirates a380 specifications," http://www.emirates.com/english/flying/our_fleet/emirates_a380/emirates_a380_specifications.aspx, accessed: 2015-08-25.

[20] *Aeronautical Information Manual, Official Guide to Basic Flight Information and ATC Procedures*, http://www.faa.gov/atpubs, Federal Aviation Administration, apr 2014.

[21] S. M. Green, "En route spacing tool: Efficient conflict-free spacing to flow-restricted airspace," in *ATM2000 3rd USA/Europe Air Traffic Management R&D Seminar, Napolo*, 2000.

[22] J. Yoo and S. Devasia, "Decoupled conflict resolution procedures for non-perpendicular air traffic intersections with different speeds," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec 2013, pp. 275–280.

[23] E. Salaün, M. Gariel, A. Vela, and E. Feron, "Aircraft Proximity Maps Based on Data-Driven Flow Modeling," *ArXiv e-prints*, Jan. 2011.

[24] J. D. Yoo and S. Devasia, "On-demand conflict resolution procedures for air-traffic intersections," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 4, pp. 1538–1549, 2014.