Computational Red Teaming for Correction of Traffic Events in Real Time Human Performance Studies

Rubai Amin*, Jiangjun Tang*, Mohamed Ellejmi†, Stephen Kirby† and Hussein A. Abbass*
*School of Engineering and Information Technology
University of New South Wales at Canberra, Australian Defence Force Academy, Australia
Email: r.amin@adfa.edu.au,j.tang@adfa.edu.au,h.abbass@adfa.edu.au
†EUROCONTROL Directorate SESAR and Research (DSR) at Bretigny-sur-Orge, Paris, France
Email: mohamed.ellejmi@eurocotrol.int,stephen.kirby@eurocontrol.int

***Abstract*—In human performance studies using real-time air traffic simulation, the human performance analyst faces challenges to ensure that certain events and scenario characteristics will occur during the experiments. While some events, such as specific categories of conflicts, can be designed in the scenario, the interaction of the humans can undo these events early in the simulation. This poses a challenge that can compromise the objective of the experiment; making the experimental data less useful.**

Computational Red Teaming (CRT) is a computational environment that attempts to play the role of a devil advocate. A CRT is designed and used in this paper to monitor, resteer and adjust traffic events in these real-time air-traffic simulation environments. The approach was able to correct events successfully, when possible. In situations were the time to correct events is greater than the time remaining for the experiments or when the constraints of the scenario do not allow certain steering requests to be issued or accepted, those events can't be recreated. Therefore, analysts are advised to avoid designing events closer to the end of the session to allow for the CRT to take corrective actions if the session does not evolve as planned.

I. Introduction

Human performance studies are very common in safety critical systems in general, and in air traffic management (ATM) in specific. These studies are critical to evaluate new concepts such as dynamic sectorization and user preferred trajectories [1], by studying mental workload and other cognitive processes for air traffic controllers (ATCs) [1], [2]. The complex processes of air traffic control rely significantly on and are limited by human performance [2].

Human performance experiments in the air traffic field have been conducted by many researchers [3], [4]. Real-time simulation is a common means for human performance studies [3], [5]. A major challenge faced by an analyst when conducting human performance studies using real-time simulation is how to correct for deviations from pre-designed air traffic events in real-time.

For example, a scenario can be designed before the experiment to test the subject's response to different mix of air traffic conflicts. The human subject can de-conflict these conflicts very early in the experiment by changing speeds of aircrafts, requesting controllers in the other sectors to delay an aircraft, or vectoring the aircraft, to name a few. Consequently, what needs to be tested in an experiment can't be tested because of the core nature of uncertainty caused by the diversity of possible strategies and behaviors of human subjects. To reduce the impact of this interference caused by human actions, a subject matter expert may deny the human subject some actions. This can be undesirable and sometimes causes frustration of the human-subject.

In this paper, we propose a Computational Red Teaming (CRT) environment to run in parallel to the real-time simulation session. CRT is a computer system that plays devil's advocates. The CRT environment monitors the real-time session, detects deviations from designed effects, then runs simulation and optimization engines to find the best set of actions required to counteract the actions causing the deviations. In other words, CRT will play devil advocates against the human subject to ensure that the session is proceeding as designed. The actions proposed by CRT are communicated to pilots and/or other air traffic controllers in the experiment as an advice. These players communicate these actions to the human-subject as normal operating requests.

We use goal programming to model the optimization problem. When a deviation from a target number of events is anticipated, the optimization problem is formulated to minimize deviations from those targets. Evolutionary goal programming is used to solve this problem by manipulating the scenarios to meet one or more criteria of target number of conflicts. Goal programming will attempt to balance across four conflict angle groups in order to maintain the target difficulty of the scenario being simulated. The system was able to successfully meet the requirements in six different experimental setups spanning the space of possible problems.

II. The concept

A human performance experiment using real-time simulation within the air traffic domain normally contains subject and non-subject players. Subject players are the human subjects that are being the focus of the analysis in an experiment

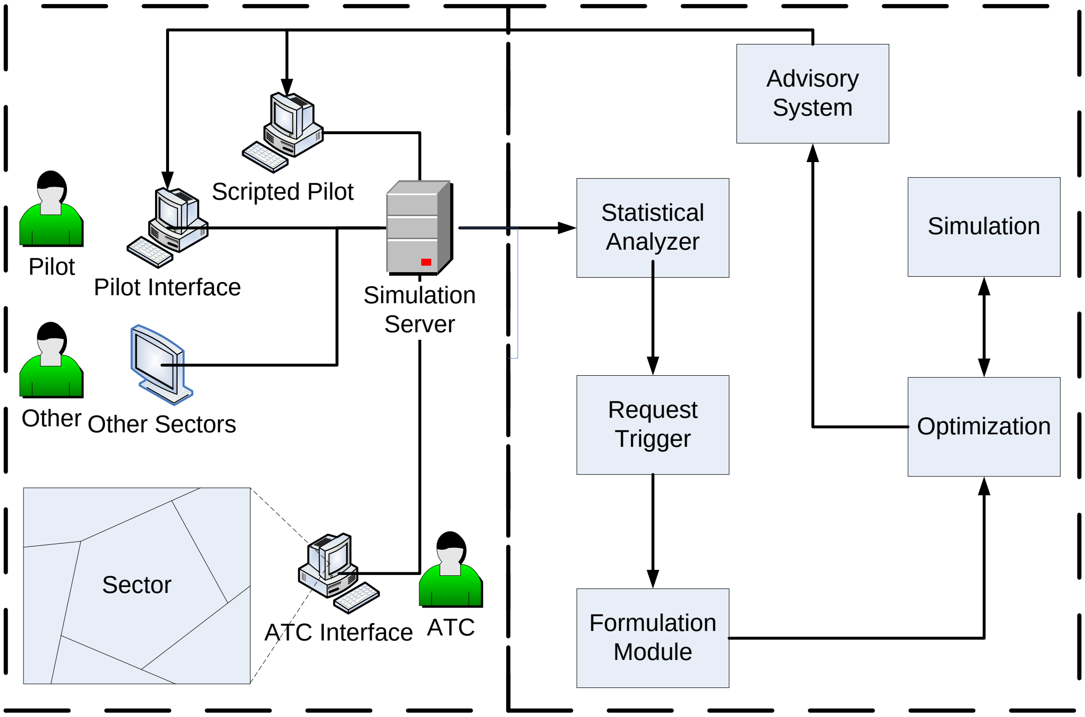


Fig. 1. A conceptual diagram of the proposed concept. The typical human performance air-traffic testing environment on the left hand side is augmented with CRT environment on the right hand side.

and are normally responsible for specific sectors. Non-subject players include pilots, other air traffic controllers, and supervisors. They are needed to recreate and imitate the operational environment in a realistic manner.

Figure 1 has two main components: the dotted boxes on the left and right sides, respectively. The dotted box on left depicts a typical real-time simulation experimental environment. A subject player is responsible for a specific sector called the measured sector. Non-subject players include air traffic controllers (this position is normally called “other” and is responsible for unmeasured sectors) controlling the sectors feeding traffic to the measured sector and human pilots controlling individual flights. Sometimes the flights are scripted, eliminating the need for human pilots. It is also common to have one or two pilots controlling all aircrafts in the experiment and one experienced air traffic controller controlling all traffic in unmeasured sectors. A simulation server provides the simulated air traffic to all other subsystems. The server responds to the interactions of pilots and ATC. It is possible for non-subject players to issue requests to the subject players to steer back the scenario to those events that are critical for the human performance study.

The concept presented in this paper is a novel automated advisory system that proposes requests to non-subject players to correct deviations from those important designed events. This concept extends the environment on the left hand side with the Computational Red Teaming (CRT) [6] environment on the right hand side. The CRT monitors all events. As soon as it detects a possible deviation from some target designed events, it activates the optimization engine.

The optimization engine commences by formulating a goal programming problem that minimizes the deviations between the existing characteristics of the system - such as predicted number of conflicts - and the target characteristics - such as the analyst pre-designed target number of conflicts. This cycle is explained in more details below.

During the experiment, the statistical analyzer generates various statistics including number of conflicts so far, which are fed to a Request Trigger component. If the analyzed results do not meet certain criteria then a number of steps are trigged to steer the traffic towards the required events. The formulation and optimization components are then triggered. These two components attempt to model and optimize several permutations of the current simulation state in order to meet the given criteria. Once a permutation has been found which meets the criteria, the actions for the original real-time simulation environment required to create these events are communicated to non-subject players. The non-subject players attempt to generate requests to the subject players with these actions to steer the scenario back to the originally designed events.

The optimization problem is best modelled as a goal programming problem. Goal programming is a common technique in optimization and is particularly useful when it is required to simultaneously consider satisfying multiple goals. There are several methods available for optimizing problems using goal programming. One of these methods is the weighted sum goal programming method where the optimization is done by assigning weights to each goal and then minimizing the weighted sum of the deviations from targets [9]. Several alternative goal programming optimization methods include the MINMAX and Lexicographic methods [10]. In the MINMAX method, the maximum deviation from the target is minimized instead of minimizing the weighted sum of the deviations. This method also makes use of weight factors. The lexicographic method assigns priorities for different goals and goals with the highest priority are considered first [10].

When a simulation environment is used to evaluate the goals and/or constraints in the system, the problem becomes a black-box goal programming problem. The name black-box comes from the fact that we do not know the explicit mathematical formulation of the behavior of the simulation. As such, it is impossible to differentiate or study the exact mathematical properties of these functions in the general case. These types of problems are most suited for evolutionary algorithms [12]. Evolutionary algorithms simultaneously work with a set (sample) of possible solutions in a single run instead of a series of separate runs as required by more traditional optimization methods [13]. This characteristic gives evolutionary algorithms the strength to produce multiple solutions in a single run.

We use differential evolution (DE) [14] as the evolutionary algorithm for this paper. DE approximates implicit direction information to guide the optimization [15]. DE compares the fitness of an offspring directly to the fitness of the corresponding parent which results in faster convergence speeds than other EAs [15]. In addition, DE is also easy to use, requires fewer control parameters and can find near optimal solutions regardless of the initial parameter values [16]. DE has been applied to a range of topics in science, engineering and management, such as logistics [17], [18] and crew rostering for airlines [19].

To test the above concept, an approach has been devised to dynamically manipulate a set of human performance scenarios by using DE to solve the goal programming problems. The aim is to change the difficulty of the scenarios by increasing or decreasing the number of conflicts throughout the scenario. This aim is achieved by searching the space of possible actions that if requested and accepted by the subject players, certain traffic events will increase. On the contrary to traditional flight scheduling [7], [8] which aims at eliminating conflicts, our objective here is equivalent to optimizing a set of actions to cause changes to scheduling flights to generate a target number of conflicts.

III. Problem Definition

We require an initial input scenario consisting of a flight plan which includes a set of aircraft $A = \{a_i\}_{i=1}^{N}$ where

$$a_i = (r, T_a, S)$$

The input information for each aircraft includes its route (*r*), activation time (T_a) and initial speed (*S*). The aircraft's route, $r = W_1, W_2, ..., W_j$; consists of *j* waypoints, *W*, each with *x*, *y* and *z* coordinates, which must be visited by the aircraft in a sequential order. Waypoint W_1 is the activation point for the aircraft and W_j is the deactivation or final point.

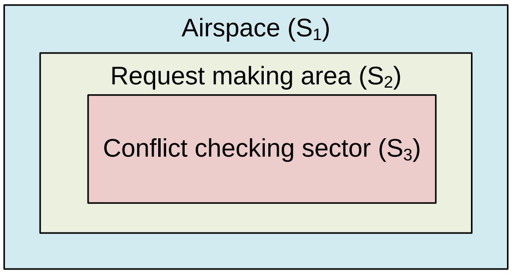


Fig. 2. A conceptual diagram of the relative positions of the request making area (S_2) and the conflict checking area (S_3) both contained within the airspace S_1

The simulated airspace, S_1, is further subdivided into two nested areas, S_2 and S_3. S_2 is an area which contains the measured sector(s) S_3, but is entirely contained within S_1. Aircraft can only make requests when they are inside S_2. S_3 is a sector of the airspace S_1 within which a target number of conflicts is required with a specific distribution of conflict types. A conceptual diagram of the positioning of S_1, S_2 and S_3 can be seen in Figure 2.

TABLE I
Classification of conflicts based on the angle of conflict

Description	**Criteria**
In-Trail (IT)	$\theta \leq 0°$ + tolerance
Crossing Narrow (CN)	$0°$ + tolerance $< \theta \leq 90°$ + tolerance
Crossing Wide (CW)	$90°$ + tolerance $< \theta \leq 180°$ - tolerance
Head-on (HO)	$180°$ - tolerance $< \theta \leq 180°$

Conflicts among aircraft are a common design-element when designing human performance experiments for ATC. In this study, a conflict is defined as the distance between two aircraft being less than or equal to 5NM while the difference in elevation between the two aircraft is less than or equal to 1000ft. Conflict can be grouped into one of four categories, as shown in Table I, based on the relative heading angles of the two aircraft engaged in the conflict.

The aircraft in the simulation is capable of making requests at various times throughout the simulation. The requests are recorded throughout the simulation as a list of requests $Q = \{q_k\}_{k=1}^{M}$ where $q_k = (a_i, T, Y, \delta)$ is produced. M is the total number of requests in the list, a_i is the aircraft which made the request, T is the time the request was made, Y is the type of request made and δ is a parameter list specific to the request, for example the number of feet to climb that is being requested.

The aim of the optimization problem is to identify the minimum number of requests that are likely to steer the simulation towards a specific number and distribution of conflicts. In this paper, we assume that we wish to achieve a uniform distribution of conflicts. To determine if the number of conflicts in a group has met the target for that group we use Equation 1 where x_n is the number of conflicts in group n, T is the target number of conflicts for each group (in goal programming language, this target is called the aspiration level), d_i^+ is over achievement for the group and d_i^- is under achievement for the group.

$$x_n - d_i^+ + d_i^- = T \tag{1}$$

To determine if all four groups consist of the target number of conflicts, we use Equation 2 where *f*, the objective or fitness function, is the sum of the deviations of the number of conflicts in each group. Because all deviations are non-negative, a solution is an optimal solution for this optimization problem if the corresponding objective value is zero. The deviations in this objective function can be weighted if required.

$$f = \sum_{i=1}^{N} d_i^+ + d_i^- \tag{2}$$

IV. Methodology

A. Simulation

A multi-agent system was developed to simulate an airspace based on an input scenario. The multi-agent approach was used as it allows for the mapping of an environment to individual agents capable of autonomous action in the environment to meet some design objectives [20]. The agents in this system were individual aircraft. The input scenario contains flight plans for a number of aircraft including their waypoints, speed and activation times. The information extracted from the flight plans was used to construct a number of routes in the airspace. The agents were simulated to travel based on their selected routes in one second time steps until all of aircraft reach their deactivation point. A point-mass approach using the equations of motion is used to model the movement of aircraft.

In this system, two simulations of the same scenario are conducted simultaneously. One of the simulations is a real time visual simulation of the scenario while the second simulation is a background simulation without visualization. The background simulation runs at a much faster rate than the real time visual simulation. The background simulation aims to dynamically change the difficulty of the scenario being simulated by allowing the aircraft to make requests to deviate from their flight plan during a specific period of time. The background simulation and the aircraft requests will be discussed in more details in the following sections.

In the real world, aircraft do not always travel in a perfect great circle route between two points due to factors such as error in meteorological conditions. These deviations are simulated through a random noise using artificial deviation points centered around the original path. Pilots can also request deviations from the set flight plan. If these requests are granted, the aircraft is allowed to deviate from the original flight plan as being discussed in the next section.

B. Aircraft Requests

An aircraft travelling through the area S_2 has the ability to make one of four requests: change flight level, reschedule arrival time, avoid an area on the route and skip a waypoint.

1) Change flight level: An aircraft can only request a change of elevation in 1000ft increments. For example, if an aircraft is travelling at FL300 then it can only request a change to either FL310 or FL290. In addition an aircraft cannot make a request to fly below FL200 or higher than FL400 which are the typical ranges of aircraft in the cruise phase. In this study, we assume that the rate of climb and rate of decent (ROCD) are fixed for all aircraft at all flight levels.

2) Reschedule overfly time: The aircraft can also make a request to delay its arrival at the deactivation point or request to arrive at the deactivation point earlier by a set amount of time, Δt, than its planned time of arrival, t_p. The new arrival time at the deactivation point is calculated using Equation 3.

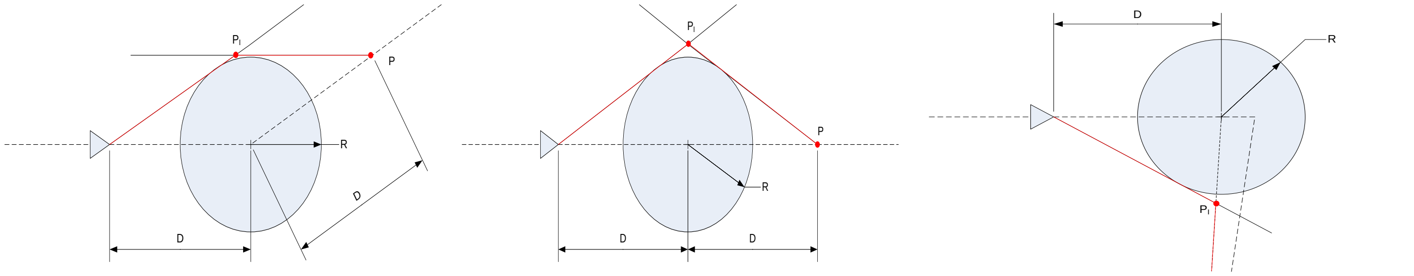
$$t_T = t_p + \Delta t \tag{3}$$

If this request is made, then the aircraft is given a target speed which the aircraft will accelerate or decelerate to, within a set time, t_a, and maintain this speed until the deactivation time is reached. The time t_a is set in such a way that the aircraft will be required to accelerate or decelerate to the target speed at the maximum acceleration or deceleration rate (ROAD). If the maximum or minimum speed for the aircraft is reached before the target speed can be reached then the aircraft maintains the maximum or minimum speed, respectively, and this means the aircraft will not be able to arrive early or delay its arrival at the deactivation point by a time of Δt.

3) Avoid an area ahead on the route: If an aircraft makes a request to avoid an area ahead on the route, such as an area of severe weather, then a circular avoidance area is created around this area with the center of the circle positioned along the route. Several intermediate artificial waypoints are added to the flight plan to guide the aircraft around this circle by taking the shortest path to rejoining the original route. If a waypoint in the original flight plan is located within a distance *D* of the center of the circle then it will not be visited by the aircraft. If the deactivation point for the aircraft is located within a distance *D* of the circle then the point cannot be avoided. The avoidance circle is always given a radius of 15NM while the aircraft will only begin deviating from the original route to avoid the circle once it is within a distance *D*, where *D* = 20NM in this paper, of the center circle.

The method for determining the position of the additional intermediate artificial waypoints is dependant on the angle created by extending a line from the aircraft's current position to the center of the circle and another line extended from the center of the circle to the next waypoint in the flight plan which is located at a distance greater than *D* of the center of the circle.

4) Skip an upcoming waypoint: If an aircraft makes a request to skip a waypoint then the upcoming waypoint in the flight plan is ignored and the aircraft begins travelling towards the next waypoint in the flight plan. For example, an aircraft has the route plan shown in Equation 4 where W_1 is the most recently visited waypoint, W_2, ..., W_{j-1} are the next waypoints in the route plan and W_j is the deactivation point.



(a) $\alpha > 90$ (b) α = 180 (c) $\alpha < 90$

Fig. 3. Shortest path taken to avoid an area (Dashed line = original path based on flight plan, solid black line = tangents and P_I and P are the additional waypoints)

While travelling between W_1 and W_2 the aircraft requests to skip the next waypoint then W_3 will be skipped and the aircraft will begin travelling towards W_4. If there are no waypoints in the route plan after W_2 other than W_j then the aircraft begins to travel towards W_j. If there are no waypoints in the flight plan after W_1 other than W_j the aircraft cannot skip any waypoints.

$$r = W_1, W_2, ..., W_{j-1}, W_j \tag{4}$$

C. Aircraft Request Time

Decisions on requests are made at a randomly selected time from a uniform distribution within a five minute window of the most recent request. The relationship between one request time and the following request time can be seen in Equation 5 where t_i is the request time that is being selected, t_{i-1} is time of the previous request and Δt_w is the length of the request time window.

$$\begin{gathered} t_i \in [t_{i-1}, t_{i-1} + \Delta t_w] \\ t_{i+1} \in [t_i, t_i + \Delta t_w] \end{gathered} \tag{5}$$

D. Aircraft Request Probabilities

The aircraft which makes a request and the nature of the request is based on given probabilities, which will be further discussed in the following subsection. The given probabilities include a set of probabilities for each request type, $R = \{P(r_i)\}_{i=1}^{N}$, where *N* is the number of request types and r_i is request type number *i*; and another set of probabilities for each aircraft, $A = \{P(a_j)\}_{j=1}^{M}$, where *M* is the number of suitable aircraft in the airspace and a_j is aircraft number *j*.

If a request and an aircraft are probabilistically chosen, this aircraft needs to be able to carry out that request. The aircraft is suitable to carry the request if none of the following is true:

- The aircraft is currently undertaking another request;
- The aircraft would violate airspace or navigational constraints such as the aircraft is currently travelling at FL400 and the selected request is to climb 1,000ft which is not possible as discussed above;
- The aircraft is outside the selected sector, S_3

During the first time step of the simulation, if there is at least one active aircraft within the selected sector then a request is made by one of these aircraft based on the given probabilities. At the next request time another aircraft and request are selected from the aircraft within the selected sector based on the given probabilities. This process continues until all aircraft have reached their deactivation point. The probability of a particular request being made from a certain aircraft at a given time is found using Equation 6. If an aircraft is not inside the selected sector or does not meet the conditions for making the request then it has a probability of zero of being selected.

$$P(r_i|a_j) = P(r_i) \times P(a_j) \tag{6}$$

Once a request and aircraft has been selected, the request is approved instantly and the aircraft adjustments are made instantly to the aircraft's flight in order to fulfil the request.

E. Real time correction of traffic events

While the visual simulation is running in real time, there is another set of simulations which are run simultaneously in the background without visualization and at a much faster rate than the real time visual simulation. The aim of these background simulations is to dynamically change the difficulty of the scenario by increasing or decreasing the number of conflicts in the scenario through new requests made by the aircraft in the flight plan.

Assume that the time when the frequency of events in the scenario needs adjustment is T_{R1}. The scenario is simulated from a time $T_{R1} + 5minutes$ to a time when all of the aircraft in the flight plan have reached their deactivation point.

Multiple lists of probabilities are generated using differential evolution (which will be discussed in the following sections). These lists include the probabilities for each of the requests and the probability of each of the aircraft making a request. Each of the lists of probabilities are separately used as an input to the simulation of the agents from time $T_{R1} + 5minutes$ to the end. The probabilities determine the type of the requests that is being made and the aircraft making the request.

During each simulation only the active aircraft can make requests. It is these requests which will either cause an increase or decrease in the number of conflicts in the scenario and therefore increase or decrease the difficulty of the scenario respectively. The requests can only be made by aircraft after the time $T_{R1} + 5minutes$ and when the aircraft are located within a selected area of the airspace, S_2. The area S_2 is located entirely within the simulated airspace, S_1, but encompasses the entire conflict checking sector, S_3. Once a list of requests has been generated from one simulation, it is evaluated using Equation 2.

There is a maximum time limit of 5 minutes for the generation and evaluation of the lists of requests. At the end of the 5 minute period the list of requests which achieved the best fitness score is added to the real time visual simulation. The 5 minute time limit guarantees that a list of requests will be available for insertion into the real time simulation when it reaches the time $T_{R1} + 5minutes$.

If the fitness score for the selected list of requests is not zero then the request generation process is repeated but this time from a time $T_{R1} + i \times 5minutes$ where i = 2. The requests from the current list between the time $T_{R1} + 5minutes$ and $T_{R1} + i \times 5minutes$ are retained and new requests are only generated after $T_{R1} + i \times 5minutes$. If a list of requests is generated which results in a fitness score equal to zero or is closer to zero than the currently selected list, then it is selected to replace the list of requests in the visual simulation after the time $T_{R1} + i \times 5minutes$. If none of the lists produce a score better than the current best then they are discarded. This process continues with an incremental increase of i by 1 until $i \times 5minutes$ equals 60 minutes or a list of requests has been found which results in a fitness score of zero, whichever is first.

A flowchart of this process can be seen in Figure 4 and an example of the selection process of the list of requests can be seen in Figure 5. At the time $T_{R1} + 60minutes$ it is again decided if the difficulty is required to be changed. If so, the request generation process is repeated again, but this time from $T_{R2} = T_{R1} + 60minutes$, in a same fashion as the process from T_{R1}. The requests which have already been inserted into the real time simulation from time T_{R1} to T_{R2} are retained and new requests are only generated after $T_{R2} + 5minutes$.

F. Differential Evolution

Differential evolution (DE) was used as the search technique to optimize the objective function. A list of un-normalized probabilities was generated randomly then was used as an input for the simulation to generate a list of requests. Based on feedback from the simulation environment, each of the lists of requests were then evaluated using the objective function to determine the fitness of each list. DE is used to find the list of probabilities which can generate the minimum value for the objective function.

Each solution (called a chromosome in DE) is represented naturally as a vector of real numbers. As shown in Figure 6, the chromosome used in this system included one parameter for each aircraft and another for each of the four requests types. The four parameters for the request types represent the probability of that request being made at a particular time while the parameters for the aircraft represents the probability of the request coming from the corresponding aircraft.

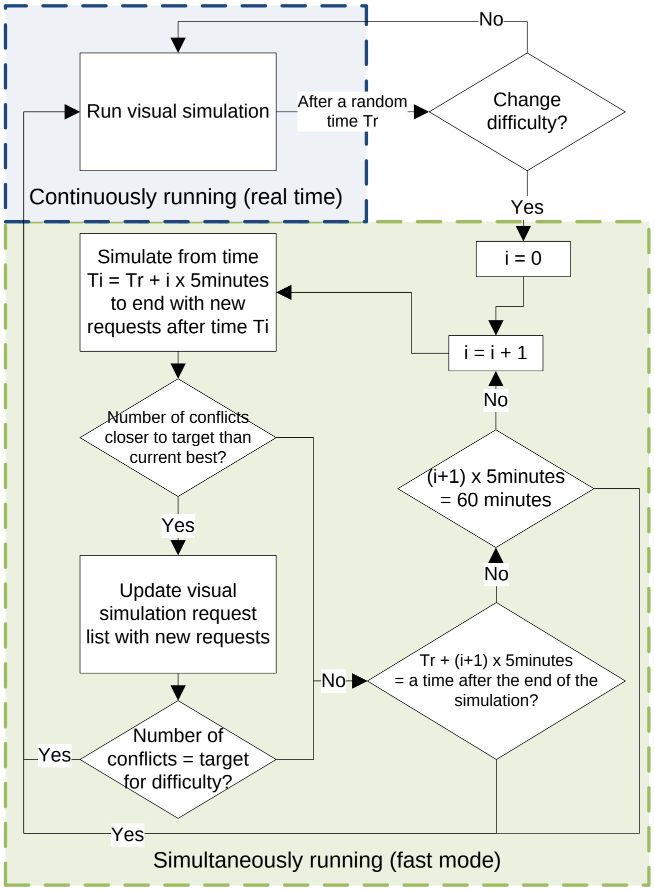
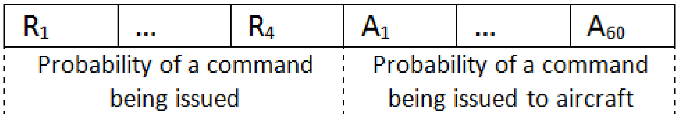

No
Run visual simulation
After a random time Tr
Change difficulty?
Continuously running (real time)
Yes
i = 0
Simulate from time Ti = Tr + i x 5minutes to end with new requests after time Ti
i = i + 1
No
Number of conflicts closer to target than current best?
(i+1) x 5minutes = 60 minutes
Yes
Update visual simulation request list with new requests
No
No
Tr + (i+1) x 5minutes = a time after the end of the simulation?
Yes
Number of conflicts = target for difficulty?
Yes
Simultaneously running (fast mode)

Fig. 4. Flowchart of background simulations



R_1	...	R_4	A_1	...	A_{60}
Probability of a command being issued			Probability of a command being issued to aircraft		

Fig. 6. Chromosome representation for use in the differential evolution process where R is the probability for a request and A is the probability for each of aircrafts

DE searches the space by used existing solutions to decide on possible directions where the fitness function will improve. Each time a new potential solution is generated, its fitness is determined through simulation. During the simulation, checks for potential conflicts are conducted at the end of each time step in the simulation between every active aircraft inside the airspace using conventional separation standards [21]. When a conflict occurs between two aircraft the position of the two aircraft are recorded along with the angle between the two aircraft at the beginning of the conflict period.

V. Experiment Design

Four input scenarios were used for testing the system. All four of these scenarios were based on the same route structure and consisted of 60 aircraft with 10 on each of 6 routes. The scenarios differed from each other as the aircraft in each of

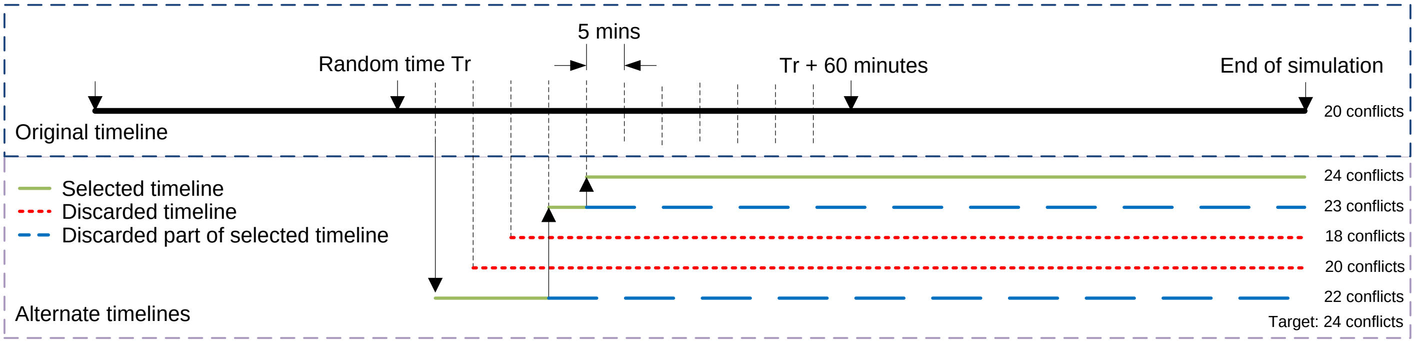


Fig. 5. Example of selection of alternate time lines for changing difficulty

the scenarios had different activation times, different speeds and different amounts of deviation from the route due to navigational error. The flight plan for each of the four scenarios would have resulted in the same number of conflicts, 20 with 5 conflicts in each of the four groups, had they been simulated without the interference caused by any requests. Each of the aircraft in the flight plan had a starting speed between 400 and 500kts which remained constant throughout the scenario as did the aircraft's elevation.

All aircraft simulated in the system were given identical performance envelops. The rate of acceleration and deceleration (ROAD) for all the aircraft were set to 2.2ft/s^2 while the rate of climb and decent (ROCD) was set to 21 ft/sec and 37 ft/sec respectively. These characteristics are typical characteristics for common commercial aircraft in the cruise phase. The maximum and minimum speeds for the aircraft were set to 500 and 400kts respectively.

Differential evolution was conducted with a population size of 20 individuals. Each run for an experiment used a different seed for the random number generator to which follows a uniform distribution in the range [0,1]. A crossover rate of 0.3 was used for all runs and the chromosomes in DE process were initialized randomly.

A. *Experiments*

Six different experiments were conducted using the four input scenarios. Each of the experiments had a different combination of changes in difficulty requirements at times T_{R1} and T_{R2} which are as follows:

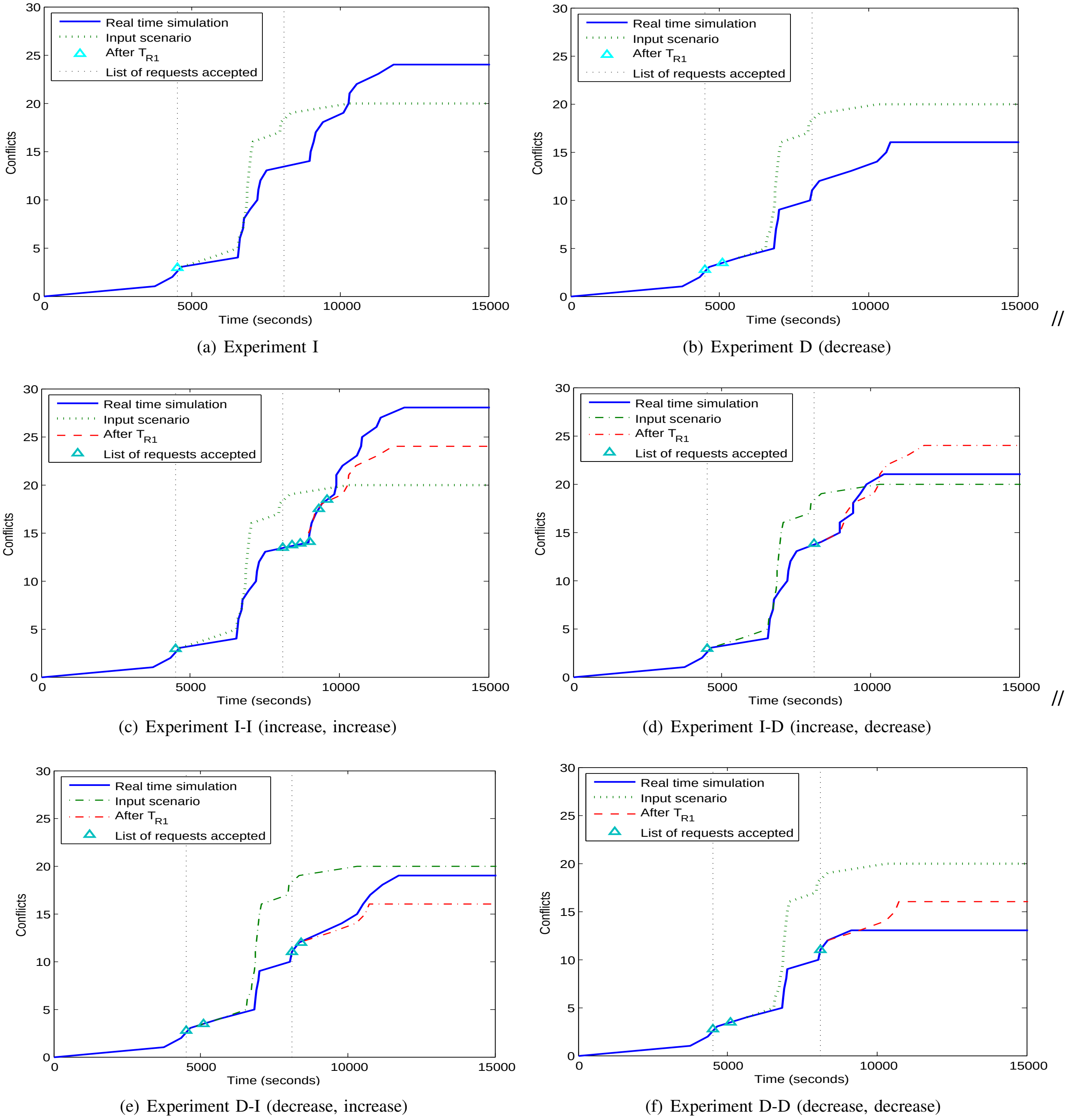
- I - T_{R1}: increase
- D - T_{R1}: decrease
- I-I - T_{R1}: increase, T_{R2}: increase
- I-D - T_{R1}: increase, T_{R2}: decrease
- D-I - T_{R1}: decrease, T_{R2}: increase
- D-D - T_{R1}: decrease, T_{R2}: decrease

In Experiment I-I an increase at time T_{R1} means an increase in the number of conflicts in the scenario. If the scenario was to have had 20 conflicts with the input flight plan, then the aim is to generate requests such that at least 24 conflicts occur in the entire simulation. Then at time T_{R2} the target is again increased to 28 conflicts in the entire scenario. With all four experiments the aim is to generate a number of requests which will results in the target number of conflicts while the conflicts are also evenly divided among each of the four groups shown in Table I.

VI. Results Analysis

The results obtained from each of the six experiments indicate that the system was successful at carrying out the requirements for each experiment. The cumulative number of conflicts for input scenario 3 was plotted as a function of time for each experiment and can be seen in Figure 7. In the plots shown in this figure, the solid line indicates the cumulative number of conflicts which will occur in the real time visual simulation, the dotted line indicates the number of conflicts which would have occurred in the original input scenario if it remained unchanged and the dashed line indicates the number of conflicts that would occur if the difficulty had only been changed at time T_{R1}. The vertical dotted lines indicate times T_{R1} and T_{R2}, the two times when it was requested to change the target number of conflicts. The conflicts which occur between these dotted lines occur after the insertion of the list of requests from the aircraft at time T_{R1}. The conflicts which occur to the right of the second dotted line occurs after the insertion of the list of requests from the aircraft at time T_{R2}. A triangle indicates a time when a new list of requests has been accepted for insertion into the real time simulation which improves the number of conflicts produced by the most recently inserted list towards the target number.

Table II shows the number of conflicts which would have occurred in each scenario without any modification of the flight plans. It also shows the total number of conflicts which would have occurred if there has been a change of difficulty at only time T_{R1} and the total number of conflicts which will occur in the real time simulation as a result of also changing the difficulty at time T_{R2}. From this table we see that in all cases the total number of conflicts which occurred meets the expected requirement for the corresponding experiments. That is, if the requirement was to increase the number of conflicts at T_{R1} and then at T_{R2} again increase the number of conflicts, then we see that for all four scenarios the number of conflicts after T_{R1} increases from the original 20 conflicts and again there is another increase after T_{R2}. While the change in difficulty requirement is met in all experiments, the target



(a) Experiment I

(b) Experiment D (decrease)

(c) Experiment I-I (increase, increase)

(d) Experiment I-D (increase, decrease)

(e) Experiment D-I (decrease, increase)

(f) Experiment D-D (decrease, decrease)

Fig. 7. Cumulative conflicts for time with scenario 3 as input

number of conflicts is not always met. As we can see in experiment 3 for scenario 4, the requirement is to increase the number of conflicts at T_{R1}. The target at this stage should be 24 conflicts, but only 22 conflicts were possible. This is a result of the time constraints put in place for the optimisation algorithm to produce the lists of requests in a real time environment.

The distribution of the sum of the deviations from the target number of conflicts from each group (the fitness from Equation 2) produced by each of the scenarios in each of the experiments after both of the difficulty changes can be seen in Figure 8.

If a list of requests inserted into the real time simulation produced a set of conflicts which results in the sum of the deviations from the target number of the conflicts from each group to equal a number other than 0 then the conflicts are not evenly distributed among the four groups. If the result is 1 then this means that in one of the four groups the number

TABLE II
NUMBER OF CONFLICTS WHICH WOULD HAVE OCCURRED IN EACH SCENARIO WITHOUT ANY MODIFICATION OF THE FLIGHT PLANS, THE TOTAL NUMBER OF CONFLICTS AFTER THE CHANGE OF DIFFICULTY AT T_{R1} AND THE TOTAL NUMBER OF CONFLICTS AFTER CHANGING THE DIFFICULTY AT TIME T_{R2}

Experiment	Scenario 1			Scenario 2			Scenario 3			Scenario 4		
	Input	T_{R1}	T_{R2}	Input	T_{R1}	T_{R2}	Input	T_{R1}	T_{R2}	Input	T_{R1}	T_{R2}
I	20	24		20	25		20	24		20	22	
D	20	16		20	16		20	16		20	19	
I-I	20	24	28	20	25	28	20	24	28	20	22	27
I-D	20	24	21	20	25	20	20	24	21	20	22	21
D-I	20	16	19	20	16	20	20	16	19	20	19	26
D-D	20	16	14	20	16	15	20	16	13	20	19	19

of conflicts does not meet the target while the remainder of the groups consists of the target number.

In Figure 8 we can see that 7 of the inserted lists of requests would have resulted in conflicts giving a fitness score of zero, another 7 produced a fitness score of 1 and 13 produced a score of 2. This means that more than 50% of the lists produced a fitness score within 2 of the target. If we further analyze the 7 lists of requests which achieved a fitness score of 0 we see that 3 of the lists were generated in 2, another 3 were generated in 11 minutes and the remaining list was generated in 6 minutes. These lists were generated within a significantly small fraction of the allocated 60 minutes for the generation and simulation of the lists. A breakdown of the deviations from the target number of conflicts for each group for scenario 3 can be seen in Table III for each experiment. From this table we can see that not all of the list of requests which were inserted into the real time simulation resulted or would have resulted in the target number of conflicts for each of the four groups. Again this goal was also difficult to achieve due to the time constraints put in place for the optimization algorithm to produce the lists of requests in a real time environment. Another factor which made this goal difficult to achieve is that the route structure provided more potential conflicts points (ie. crossings between routes) classified in crossing narrow than the other three groups.

TABLE III

Experiment	T_{R1}				T_{R2}			
	IT	CN	CW	HO	IT	CN	CW	HO
I	-1	+1	0	0				
D	0	0	0	0				
I-I	-1	+1	0	0	0	+1	-1	0
I-D	-1	+1	0	0	0	0	0	0
D-I	0	0	0	0	0	0	0	-1
D-D	0	0	0	0	0	0	0	+1

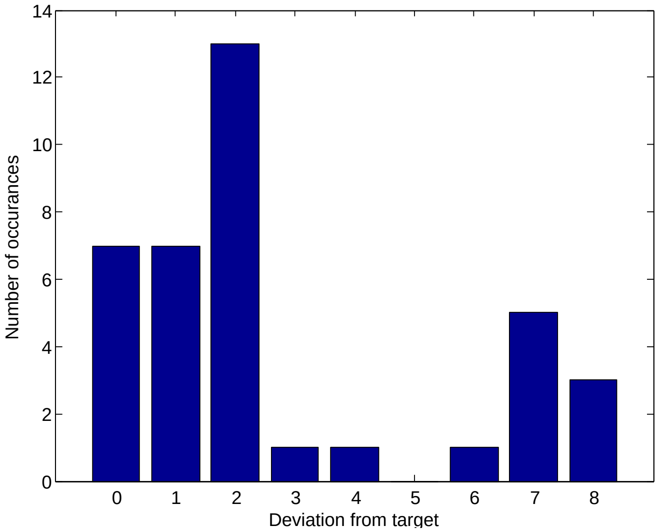


Fig. 8. Sum of the deviation from the target number of conflicts for each group

VII. CONCLUSION

In this paper, we presented a computational red teaming approach for the correction of air traffic events in real time for human performance studies. The presented approach has been shown to be capable of correcting the scenarios in real time by meeting the requirements of all six tested experiments for four different scenarios. The problem was modelled as a goal programming model, where deviations from goals are minimized. While the objectives of the experiments were met, it was not always possible to meet the target goal levels if the time were insufficient or scenario constraints will be broken. This suggests that analysts should not schedule events late in a real-time simulation environment to allow the proposed concept to correct for events.

The optimization approach used in this paper was fast, allowing solutions to be optimized between a time windows of 5 minutes. This approach can be made faster, but the details of evolutionary computations and tricks to speed it up are outside the scope of this paper.

This work has a huge potential for human performance studies. While the goals defined in this paper are related to the number of conflicts and distribution of conflict types, the optimization problem is very general. For example, cognitive and task load complexity indicators can be used instead. Moreover, the approach is applicable to many domains including human performance studies for command and control (C2), space, and other safety critical.

ACKNOWLEDGMENT

This work has been co-financed by the European Organisation for the Safety or Air Navigation (EUROCONTROL) under its University Research Grant programme. The content of the work does not necessarily reflect the official position of EUROCONTROL on the matter.

References

[1] S. Loft, P. Sanderson, A. Neal, and M. Mooij, "Modeling and predicting mental workload in en route air traffic control: Critical review and broader implications," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 49, no. 3, pp. 376–399, 2007.

[2] S. Inoue, K. Furuta, K. Nakata, T. Kanno, H. Aoyama, and M. Brown, "Cognitive process modelling of controllers in en route air traffic control," *Ergonomics*, vol. 55, no. 4, pp. 450–464, 2012.

[3] A. Vela, J. Clarke, E. Feron, N. Durand, and W. Singhose, "Determining the value of information for minimizing controller taskload: A graph-based approach," *USA-Europe ATM R&D Seminar*, 2011.

[4] Y. Wang, M. Hu, P. Bellot, F. Vormer, and V. Duong, "Spatial, temporal, and grouping behaviors in controller communication activities," *USA-Europe ATM R&D Seminar*, 2011.

[5] C. Moehlenbrink, A. Papenfuss, and J. Jakobi, "The role of workload for work organization in a remote tower control center," *Air Traffic Control Quarterly*, 2011.

[6] H. Abbass, A. Bender, S. Gaidow, and P. Whitbread, "Computational red teaming: past, present and future," *Computational Intelligence Magazine, IEEE*, vol. 6, no. 1, pp. 30–42, 2011.

[7] E. K. Burke, P. D. Causmaecker, G. D. Maere, J. Mulder, M. Paelinck, and G. V. Berghe, "A multi-objective approach for robust airline scheduling," *Computers & Operations Research*, vol. 37, no. 5, pp. 822 – 832, 2010, disruption Management. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054809000896

[8] M. Dunbar, G. Froyland, and C. Wu, "Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework," *Transportation Science*, vol. 46, no. 2, pp. 204–216, 2012.

[9] C.-F. Hu, C.-J. Teng, and S.-Y. Li, "A fuzzy goal programming approach to multi-objective optimization problem with priorities," *European Journal of Operational Research*, vol. 176, no. 3, pp. 1319 – 1333, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221705008817

[10] M. Bertolini and M. Bevilacqua, "A combined goal programmingahp approach to maintenance selection problem," *Reliability Engineering & System Safety*, vol. 91, no. 7, pp. 839 – 848, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832005001559

[11] C. A. Coello, "An updated survey of ga-based multiobjective optimization techniques," *ACM Comput. Surv.*, vol. 32, no. 2, pp. 109–143, Jun. 2000. [Online]. Available: http://doi.acm.org/10.1145/358923.358929

[12] K. Deb, "Solving goal programming problems using multi-objective genetic algorithms," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, 1999, pp. 3 vol. (xxxvii+2348).

[13] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information Systems*, vol. 1, pp. 269–308, 1998.

[14] D. GA and S. Okdem, "A simple and global optimization algorithm for engineering problems: differential evolution algorithm," *Turk J Elec Engin*, vol. 12, no. 1, 2004.

[15] D. He, F. Wang, and Z. Mao, "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect," *International Journal of Electrical Power & Energy Systems*, vol. 30, no. 1, pp. 31–38, 2008.

[16] A. Abou El Ela, M. Abido, and S. Spea, "Differential evolution algorithm for emission constrained economic power dispatch problem," *Electric Power Systems Research*, vol. 80, no. 10, pp. 1286–1292, 2010.

[17] C. Erbao and L. Mingyong, "A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands," *Journal of computational and applied mathematics*, vol. 231, no. 1, pp. 302–310, 2009.

[18] L. Mingyong and C. Erbao, "An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 2, pp. 188–195, 2010.

[19] B. Santosa, A. Sunarto, and A. Rahman, "Using differential evolution method to solve crew rostering problem," *Applied Mathematics*, vol. 1, no. 4, pp. 316–325, 2010.

[20] G. Weiß, *Mutiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, ser. Intelligent Robotics and Autonomous Agents Series. Mit Press, 1999. [Online]. Available: http://books.google.com.au/books?id=JYcznFCN3xcC

[21] R. Slattery and S. Green, "Conflict-free trajectory planning for air traffic control automation," 1994.

Rubai Amin received the Bachelor in Engineering from the Australian National University, Canberra, Australia. He is currently an ATM simulation engineer at the University of New South, Australia. His main research interests are environmental modelling, optimization-simulation, and multi-agent simulation of ATM operations.

Jiangjun Tang is a Senior ATM Research Fellow at the School of Eng. & IT , University of New South Wales@ADFA, Australia. He received his B.A. from Shanghai University (1999), M.Sc. from Australian National University (2004), and Ph.D. on dynamic sectorization from UNSW in (2012). From 2006 till now, he has been working on various projects at UNSW in the air traffic domain. His research interest are in modelling and simulation of advanced ATM concepts including dynamic sectorization, human performance, and RNP. He authored many journal articles and conference papers in ATM.

Mohamed Ellejmi PMP, is a project manager at EUROCONTROL research and experimental Centre at Brtigny sur Orge, France. He has been working for the last two years on the Single European Sky ATM Research programme (SESAR) in the airport domain. He has an engineering degree from the National Civil Aviation School at Toulouse France. His research interests focus on the airport domain (A-SMGCS and Collaborative Decision Making).

Stephen Kirby read Astrophysics at University College London (B.Sc.), then read Astronautics and Space Engineering at Cranfield, UK, (M.Sc.). After five years in operational research on defence problems (procurement, optimum use of equipement), he then moved into the air traffic management sector. For the past ten years he has worked mostly on problems concerning safety of air traffic services. He was a member of several ICAO mathematical and operational working groups. He is currently managing research and development work for the pan-European SESAR programme.

Hussein Abbass is a Full Professor at the School of Eng. & IT, University of New South Wales, Canberra, at the Australian Defence Force Academy, Australia. His research focuses on Modelling, Simulation and Cognitive Engineering of Air Traffic Management and Network Centric Operations. He is a Fellow of the Australian Computer Society, a Fellow of the Operations Research Society (UK), and a Senior Member of the IEEE. He has been the Research Leader on UNSW Strategic Alliance with Airservices Australia and projects funded by Australian Research Council, Defence Science and Technology Organization, Eurocontrol, and Thales.