

Trading-off Simulation Fidelity and Optimization Accuracy in Air-Traffic Experiments using Differential Evolution

Rubai Amin*, Jiangjun Tang*, Mohamed Ellejmi[†], Stephen Kirby[†], and Hussein A. Abbass*

*School of Eng. and IT, University of New South Wales, Canberra, Australia
Email: r.amin@adfa.edu.au, j.tang@adfa.edu.au, h.abbass@adfa.edu.au

[†]Eurocontrol Experimental Centre, Brtigny, France
Email: m.ellejmi@eurocontrol.int, s.kirby@eurocontrol.int

Abstract—In many engineering applications, black-box optimization relies on the use of a simulation to obtain a numeric evaluation or a score for a proposed solution. The cost of optimization is mostly a reflection of the cost of running this simulation environment. On the one hand, the higher the fidelity of the simulation environment, the longer it is likely to take to evaluate a single solution. Consequently, less solutions are allowed to be evaluated given a time constraint on the running time of the optimization algorithm.

On the other hand, the lesser the fidelity of the simulation environment, the more likely more solutions could be evaluated within the same time constraint. However, understanding the relationship between fidelity and the quality of the final solution obtained by the optimization method is largely an unexplored area of research.

In this paper, we present an approach for adjusting taskload of Air traffic controllers (ATC) in real time by using three different shadow simulators of increasing fidelity and Differential Evolution (DE) as the evolutionary optimization algorithm. According to air traffic conditions, DE optimizes a goal programming model to steer the taskload up or down towards a predefined taskload target by generating two ATC requests every 10 minutes.

The results demonstrate how a high fidelity simulator can help DE to achieve better solution quality in the absence of any time constraint on running the experiments. However, when there is a tight time constraint imposed, lower fidelity simulators allow DE to explore more solutions in the search space by cutting down on the extra time needed when higher fidelity simulators are used.

I. INTRODUCTION

Air traffic controllers (ATC) are an important part of the air traffic management system as they are responsible for maintaining a safe and efficient flow of air traffic in a controlled airspace. At any point of time, an ATC could be responsible for simultaneously managing multiple aircraft, and therefore the safety of hundreds of people. The complex process of air traffic control relies significantly on and is limited by human performance [1]. Like many complex and dynamic systems, the ATC can't temporarily halt the air traffic system to take a break when the taskload becomes too complex. For this purpose, a strategy needs to be in place to shift this load between the humans and machines in order to maintain an

ideal level of complexity according to the ATC's capabilities [2]. Presently, air traffic flow management strategies rely on centralised systems to produce routes for aircraft [3]. This is conducted over a large time frame, ranging from one hour to one year in advance and often encompass large regions, such as the entirety of the Australian airspace. This causes the system to be slow in responding to developing localised weather conditions, airport conditions and other uncertainties, which could potentially lead to local delays [3]. These local delays could grow to form larger regional congestion. The increased congestion in turn may exceed the initially planned level of traffic within certain sectors and the capacity of the sector; and at the same time exceed the capabilities of the ATCs allocated to these sectors. In order to handle these uncertainties we require a method by which to adjust the ATC's taskload in real-time.

To facilitate the adjustment of the ATC's taskload in real-time, it may be necessary for involved parties (ie. ATCs, pilots, etc.) to take actions to correct from their current or planned states. To generate these actions and evaluate their effectiveness, we require an optimisation system. Optimisation can be a time-consuming activity, especially when a simulation is required to evaluate possible solutions. A number of advanced air traffic management simulators currently exist. Examples include the Future ATM Concepts Evaluation Tool (FACET) [4], Total Airport and Airspace Model (TAAM) [5] and the Air Traffic Operations and Management Simulator (ATOMS) [6]. All of these simulators provide high level of fidelity for modelling the airspace and the aircraft. For example, the ATOMS system incorporates atmospheric, weather and wind modelling; and complex aircraft modelling which includes various aircraft performance parameters such as fuel flow computation, aircraft acceleration/deceleration, aircraft banking angle calculation and the calculation of the effect of wind on the aircraft. This simulator, along with other advanced simulators within the air traffic domain, are themselves time-consuming and operating these large computationally expensive simulator is not practical for real-time applications [7], especially when

the optimisation must be completed within a specified time frame. Although high fidelity simulators are necessary as predictive tools for some applications, the solution, for real-time optimisation applications, is to run the simulation with only a limited number of parameters. In this paper, we demonstrate the effectiveness of incorporating simple, low fidelity, simulation systems for use in real time optimisations. We compare the effectiveness of these simple simulators to the results achieved when using a much more complex, high fidelity, simulator, ATOMS, in real-time optimisation. The system will use ATOMS as a visual simulator which will periodically send updates to an optimisation component. This optimisation component will use a shadow simulator, a low fidelity simulator, to generate a set of actions in order to adjust the taskload of the current air traffic state.

Evolutionary algorithms are a common method of solving optimization problems as the algorithm simultaneously works with a set of possible optimal solutions in a single run instead of a series of separate runs as required by some other methods [8]. Using evolutionary algorithms for optimization also has the potential to produce multiple solutions in one run. Differential evolution (DE) is one such evolutionary algorithm. DE uses direction information to guide the search and compares the fitness of an offspring directly to the fitness of the corresponding parent which results in faster convergence speeds than other EAs [9]. In addition, DE is also easy to use, requires fewer control parameters and can find near optimal solutions regardless of the initial parameter values [10]. DE has been applied to a range of topics in science, engineering and management, such as logistics [11] and crew rostering for airlines [12].

In this paper, we present an approach for adjusting the ATC taskload in real time by using shadow simulators and DE. Here, the ATC taskload is measured by one of the well-known Dynamic Density (DD) metrics [13]. A taskload target is defined as the average taskload of the whole period from a given air traffic scenario. According to the air traffic conditions, DE applies goal programming to push the taskload up or down towards the target by generating two requests every 10 minutes for the main simulator, such as climbing to a given flight level and skipping some navigation points. Instead of the main simulator, a shadow simulator is used as an evaluator in DE. Three shadow simulators with different modeling fidelities including ATOMS are investigated here in order to evaluate the performance of them when working with DE.

II. PROBLEM DEFINITION

An initial scenario is required as an input consisting of flight plans for a set of aircraft $A = \{a_i\}_{i=1}^N$ where

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

The input flight plan consists of the aircraft's route (r), activation time (T_a), initial speed (S) and requested flight level (RFL). The aircraft's route, $r = W_1, W_2, \dots, W_j$, contains j waypoints, W , each with a latitude-longitude coordinate and optionally an elevation. Waypoint W_1 is the activation point

for the aircraft and W_j is the deactivation or final point. Both of these points could either be an airport at ground level or an en route waypoint.

At various stages of the real-time visual simulation, requests are generated and evaluated for the suitability for incorporation into the real-time simulation. The requests are generated as a list of requests $Q = \{q_k\}_{k=1}^M$, where $q_k = (a_i, T, Y, \delta)$. 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 is to be made, Y is the type of request to be made and δ is a value specific to the request, for example the number of feet to climb.

The aim of the optimisation is to generate a number of requests that is likely to influence and change the taskload complexity measured in the real-time simulation for a specific period of time towards a predefined target level. Taskload complexity can be calculated from an air traffic state using a number of methods [13]. One widely used methods is a method commonly known as the NASA-II method [14]. This method takes into account various characteristics of the traffic state of a specified area, the measured sector. This includes the number of aircraft in the area, the number of aircraft with heading changes, speed changes and altitudes changes greater than 15° , 10 kts and 750 ft respectively in the last two minutes and several groupings of aircraft pairs based on their lateral separation distances (0-25NM, 25-40NM, 40-70NM) and three dimensional distances (0-5NM, 5-10NM). The complexity of the traffic state is calculated periodically by summing these metrics after being multiplied by a weight. By using the weights, aircraft pairings which fall into the 0-25 NM lateral distance group make a bigger contribution to the complexity figure than the aircraft pairings which fall into the 40-70 NM group, as the aircraft pairings falling into the 0-25NM group are a bigger concern for the ATC than the pairings in the 40-70NM group. To determine if the taskload complexity at a particular time has met the predefined target we use Equation 1, where x is the taskload complexity, T is the target level of complexity, d_i^+ is over achievement of the target and d_i^- is under achievement of the target.

$$x - d_i^+ + d_i^- = T \quad (1)$$

As the aim is to change the taskload complexity of a specific period of time, we use Equation 2 to determine how well the complexity measurements for this period meets the target. In Equation 2, f is the objective or fitness function, which is calculated by taking the sum of the deviations of each of the complexity measurements from the target level in the specified time period. As all deviations are non-negative, a global optimal solution for this optimisation problem occurs when the objective value is zero.

$$f = \sum_{i=1}^N d_i^+ + d_i^- \quad (2)$$

III. METHODOLOGY

A. Overview

As illustrated in Figure 1, two major components exist in our system: a shadow simulator and DE.

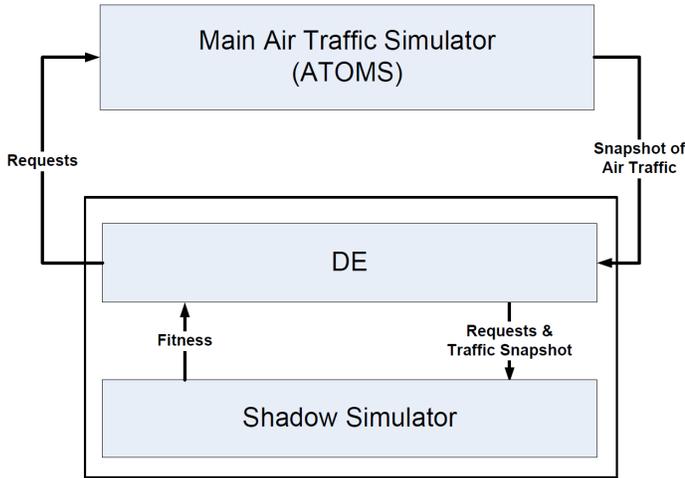


Fig. 1. System overview

The main simulator sends snapshots of air traffic to our system and the DE produces possible solutions (requests) effecting the future taskload, which are evaluated by a shadow simulator. Based on the fitness, the best requests are sent to the main simulator and are executed. The details of each component are discussed in the following sections respectively.

B. Simulation

A number of different simulation systems were used as part of this study. The simulation systems used included the Air Traffic Operations and Management Simulator (ATOMS) [6], a basic simulator which does not include many of the domain specific features of ATOMS and another basic simulator which includes a limited number of the domain specific features found in ATOMS. All three of these systems use an agent-based system architecture for simulation.

ATOMS can simulate end-to-end airspace operations and air navigation procedures for conventional air traffic, as well as for free flight. A complex aerodynamic model based on the Base of Aircraft Data (BADA) [15], which includes the detailed aircraft climb/descent, acceleration/deceleration, and turning models, supports ATOMS to provide high fidelity simulation results for any level operation analysis and optimisation exercises [16]. However, the computational cost of ATOMS for such detailed aerodynamic modelling may not be suitable to use it as an evaluator when involving population based optimisation method for a time constrained optimisation problem.

The two basic simulators were developed as agent-based systems. The aircraft in these simulators are modelled as point masses in 3-D space and use the equations of motion to model the movement of the aircraft. The first simulator (Basic

Simulator 1) does not take into account any of the aerodynamic properties, aircraft performance, atmospheric conditions or weather conditions which are considered in ATOMS. This simulator uses a fixed rate of climb/descent (ROCD) and a fixed rate of acceleration/deceleration (ROAD) for all aircraft, irrespective of aircraft model, elevation and speed. The second basic simulator (Basic Simulator 2) is an extension of the first basic simulator, but takes into account an additional feature which is available in ATOMS. This simulator uses aircraft performance data available from BADA to determine an aircraft's rate of climb and descent and acceleration rate based on its model and elevation. The major differences of these three simulators are summarised in Table I.

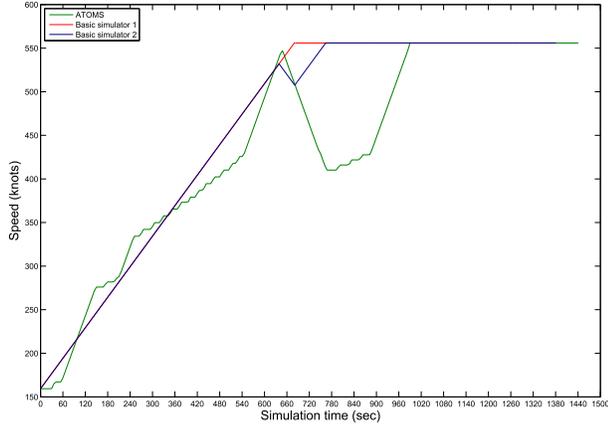
TABLE I
COMPARISON OF MAJOR THE DIFFERENT SIMULATORS

Feature	ATOMS	Basic Simulator 1	Basic Simulator 2
Climb	Aerodynamic Model	42 ft/s	BADA tables
Descent	Aerodynamic Model	35 ft/s	BADA tables
ROAD	Aerodynamic Model	0.3 m/s ²	BADA Tables
Turning	3°/s	Direct turn	Direct turn
Navigation	Fly pass	Fly over	Fly over

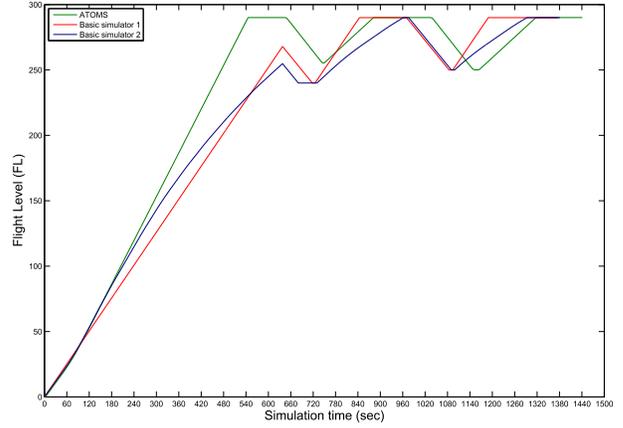
A comparison of a sample aircraft's speed and elevation along time can be seen in Figure 2 when being simulated with ATOMS, the Basic Simulator 1, and the Basic Simulator 2. The sample aircraft's flight plan was as follows: Take off from an elevation of 0ft, climb to a cruise elevation of 29,000ft, when the fifth waypoint is reached descend to 24,000ft, when the sixth waypoint is reached return to cruise altitude, when the seventh waypoint is reached descend to 25,000ft and finally when the eighth waypoint is reached return to cruise altitude.

It can be seen from Figure 2 that the speed and elevation profile for this flight varies between the three simulators. The average speeds when using both Basic Simulators are higher than when using ATOMS. As a result, waypoints are reached earlier in both Basic Simulators, so the climbing and descending requirements from the flight plan are carried out and fulfilled earlier. From Figure 2(b), it can be seen that the climb rate used in the three simulators are also different and that it is not possible for the aircraft to descend to the required altitude in ATOMS before the waypoint is reached. Figure 2(b) also shows that the rate of climb/descent (ROCD) of Basic Simulator 2 is more similar to ATOMS than Basic Simulator 1. In addition, Basic Simulator 2 has similar behavior to ATOMS in terms of speed profile when the aircraft is descending, which is illustrated by the window of 600 to 800 seconds in Figure 2(a).

In all experiments in this study, two simulations were operated simultaneously. One of the simulations was a real time visual simulation conducted with ATOMS, while the second simulation, the shadow simulation, operates in a fast-time mode in the background without visualisation. The shadow simulation runs at a much faster rate than real-time. The aim of the shadow simulation is to evaluate and estimate the impact of dynamically changing the complexity of the measured sector by allowing the aircraft to make requests to deviate from their



(a) Comparison of speed



(b) Comparison of altitude

Fig. 2. Comparison of the speed and altitude of an aircraft using ATOMS and the Basic Simulator 2

TABLE II
TIME WINDOWS FOR OPTIMISATIONS AND REQUESTS

	Request 1	Request 2
Look ahead time start	$T_{LS} = \text{Most recent traffic snapshot time}$	
Look ahead time end	$T_{LE} = T_{LS} + 10 \text{ minutes}$	
Optimisation cut-off time	$T_{OE} = T_{LS} + 2\text{minute}$	
Execution time range start	$T_{ES1} = T_{OE} + 1 \text{ minute}$	$T_{ES2} = T_{EE1}$
Execution time range end	$T_{EE1} = T_{ES1} + (T_{EE2} - T_{ES1})/2$	$T_{EE2} = T_{LE2} - 1 \text{ minute}$

original flight plan at specific periods of time. The shadow simulation and the aircraft requests will be discussed in more detail in the following sections.

C. Requests

An aircraft travelling through the measured sector can be chosen to make one of eight requests:

- Climb: The aircraft can request to climb 2000ft
- Descend: The aircraft can request to descend 2000ft
- Change speed: Exit the sector up to 5 minutes earlier
- Change speed: Exit the sector up to 5 minutes later
- Turn right: Change heading by 5° in a clockwise direction, then return to original path after 2 minutes
- Turn left: Change heading by 5° in a counter-clockwise direction, then return to original path after 2 minutes
- Skip upcoming waypoints: Skip a number of upcoming waypoints such that it results in a net heading change
- Emergency landing: Immediately begin descent to a flight level lower than the sector

The request to be made by a particular aircraft is generated using the shadow simulation and a set of probabilities optimised using differential evolution. The execution time of the requests is also generated from differential evolution.

In order to obtain a wider variety of requests, only one instance of each request is allowed to be sent for execution within a predefined period in the real time visual simulation.

The emergency landing request is allowed to be executed only once during the entire real time visual simulation as this event occurs rarely relative to the other listed requests in real world traffic conditions. In air traffic management environments the pilot may need approval from the air traffic controller before some of these requests can be executed. But for the purpose of this study, it is assumed that all requests are approved by the ATC.

D. Shadow simulation

While the visual ATOMS simulation is operating in real time, another set of simulations, the shadow simulations, are run in the background. The shadow simulation is run without visualisation and at a much faster rate than real time. The aim is to use the shadow simulation, in conjunction with differential evolution, to dynamically change the complexity of a measured sector by allowing the aircraft to make requests to deviate from its original flight plan at specific periods of time.

The optimisation system receives a snapshot of the current traffic state (ie. each aircraft's positions, speed and heading) at regular intervals, $I_{snapshot}$, from the real-time environment. When the snapshot is received, the optimisation system is triggered. The optimisation system aims to produce up to two requests which are to be executed at two different times. The execution time for each request is determined from a time range relative to the simulation time received from

the snapshot, which can be seen in Table II. The execution time and the request are determined using the probabilities generated from the differential evolution component. When the optimisation cut-off time is reached, the optimisation is stopped and the best generated set of requests thus far is sent to the real time simulation. A one minute gap between the cut-off time and the execution time of the second request is kept in order to accommodate transfer time of the request to the real time environment.

Request 1 is generated first and then Request 2 is generated from the same set of probabilities. Request 2 is constrained to be a different request from that of Request 1. It is also constrained to be generated for a different aircraft from the aircraft that was associated with Request 1. If it is not possible to generate Request 2 with the previous two constraints, and after ten unsuccessful attempts, then the simulation goes ahead with only Request 1.

Before any request generation can occur, the traffic scenario is simulated using the shadow simulation from the most recent traffic snapshot time, T_{LS} , to the look ahead end time, T_{LE} . During this simulation, all the aircraft which entered the measured sector between the times T_{LS} and T_{LE} are recorded and the aircraft which did not enter during this period are disregarded from use in further simulations for the generation of this particular request. This is done in order to reduce the total number of aircraft being simulated which do not have a bearing on the final outcome of the optimisation; only aircraft which enter or are already inside the measured sector during this period impact complexity. Additionally, as there are fewer aircraft being simulated, the overall simulation time of the shadow simulation will be faster than simulations which included these inconsequential flights and will therefore allow for more simulations before the time T_{OE} is reached.

Multiple lists of probabilities are produced using differential evolution (which will be discussed in more detail in the following sections). These lists contain the probability that each of the requests will be executed and a probability for each aircraft for making the request. Each list of probabilities is used separately as an input into the shadow simulation. The shadow simulation then uses these probabilities to generate a request-aircraft pairing. The scenario is then simulated from time T_{LS} to T_{LE} with the generated requests being executed. At the end of the simulation, the measured complexity for this period is used to evaluate the effect of implementing these requests using Equation 2.

Once time T_{OE} has been reached in the real-time simulation, the best set of requests is sent to the real time environment for execution. The request is automatically executed when the request's execution time is reached.

E. Request Probabilities

The aircraft which makes a request and the nature of the request is based on given probabilities, obtained from the chromosomes which were generated using differential evolution. The given probabilities include a probability 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 ID i ; and another probability for each aircraft, $A = \{P(a_j)\}_{j=1}^M$, where M is the number of aircraft entering the measured sector between T_{LS} and T_{LE} , and a_j is aircraft number j . The probability of a particular request being made from a certain aircraft at a given time is found using Equation 3.

$$P(r_i|a_j) = P(r_i) \times P(a_j) \quad (3)$$

The execution time for each request were also obtained from chromosomes generated using differential evolution.

F. Differential Evolution

T_1	T_2	R_1	...	R_8	A_1	...	A_n
Time for execution of requests		Probability of a request being issued			Probability of a request being issued to aircraft		

Fig. 3. Chromosome representation for use in the differential evolution process where T is the execution time for each request, R is the probability for a request and A is the probability for each of aircrafts

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 to generate a list of requests for the shadow simulation. Based on feedback from the shadow simulation environment, each list of requests was then evaluated to determine the fitness of each list. DE search for 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 eight request types. There were also two parameters for the execution time for each of the requests to be generated. The eight parameters for the request types represent the probability of that request being made at a particular time while the parameters for the aircraft represent the probability that the request is coming from the corresponding aircraft. The parameters are initialised with random values. The minimum and maximum values for the execution time are obtained from Table II while the minimum and maximum values for the remaining parameters are 0 and 100 respectively. If a particular request is not allowed, then its maximum value is set to 0, thus making it impossible to be selected using Equation 3;

DE searches the space by using 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 using a shadow simulation.

IV. EXPERIMENT DESIGN

Three different experiments were conducted as part of this study. Each experiment used each of the simulators introduced in the previous section for the shadow simulation, while all experiments used ATOMS for the real time environment. A breakdown of the experiments can be seen in Table III.

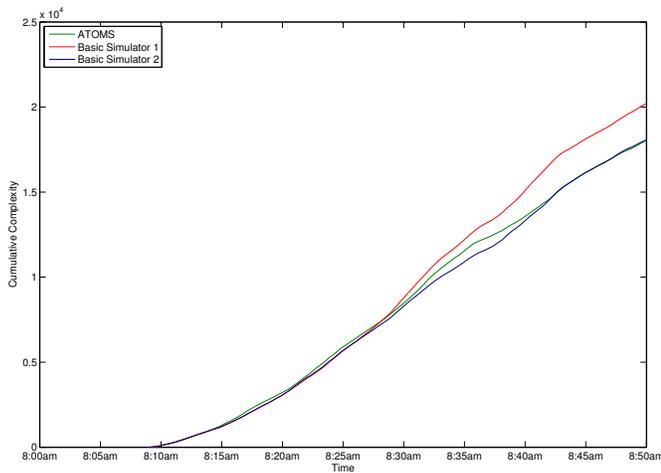


Fig. 4. Cumulative Taskload complexity from the three different simulators for the input scenario

TABLE III
EXPERIMENTS SETUPS

Experiment	Visual Simulation	Shadow Simulation
AB1	ATOMS	Basic Simulator 1
AB2	ATOMS	Basic Simulator 2
AA	ATOMS	ATOMS

Each of the experiments were conducted with the same input scenario. This input scenario contained 42 aircraft with a range of different characteristics. Some aircraft were activated outside the measured sector, some had origin airports within the sector and some had destination airports within the sector. The measured sector was defined in 3D space above 25,000ft. Most aircraft entering the sector had an intersecting route with another aircraft that was also within the sector, but intersection may not have necessarily resulted in a violation of separation required to increase the taskload complexity. A screenshot of the real-time environment can be seen in Figure 5. This figure shows the measured sector and the flight plans of some aircraft in the input scenario. The real time environment was run for 50 minutes in each of the three experiments. Each request type was only allowed to be executed in the real-time environment once in the first 25 minutes and again in the second 25 minute block.

The traffic state snapshots were sent from the real-time environment once every five minutes. The air traffic complexity was measured for a rolling window for the previous five minutes with a frequency of ten times a minute in all simulations. A comparison of the taskload complexity from the three different simulators can be seen in Figure 4. It shows the complexity of the scenario as simulated by ATOMS for the 50 minute period used in the experiments. It can be seen from this figure that the taskload complexity slowly builds up from around 8:10 and reaches a peak at around 8:30 as more aircraft become active and ready to enter the measured sector. A number of key differences can be seen between the taskload complexity plots from the different simulators. This

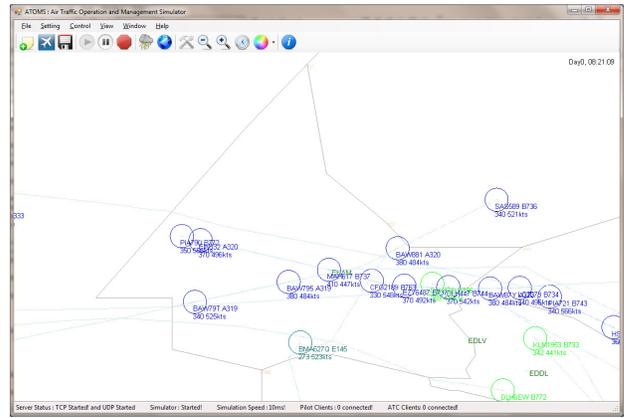


Fig. 5. Screenshot of the realtime visual simulation

is a result of the fidelity of the different simulators and the way each of them handle heading changes, elevation changes and speed changes; and the flow on effects of causing aircraft pairings to be classed under different groups when calculating the complexity. When simulating between T_{LS} and T_{LE} with a snapshot from the real-time environment, these differences are lowered, but not entirely eliminated. When we look at the cumulative complexity in Figure 4, the differences between ATOMS and the Basic Simulator 2 is much smaller.

The target, T , for Equation 2 was set to 45, which is the average taskload complexity from the input scenario when using ATOMS as seen in Figure 4.

For each snapshot, differential evolution was operated for 100 generations with a population size of 20 individuals. The parameters for the differential evolution was set at the recommended initial values for the crossover constant (CR) of 0.9 and the amplification factor (F) of 0.5 [17]. If the differential evolution run was completed before time T_{OE} (2 minutes), it gets re-initialised with the same snapshot, but with a different seed for the random number generator. This process continued until time T_{OE} was reached in the real-time environment. At this point of time, the best list of requests generated from all differential evolution runs for this snapshot were selected for use in the real-time simulation.

V. RESULTS

The results obtained from the three experiments show that the system was successful at meeting the aim of changing the taskload complexity of the real-time environment as can be seen in Figure V. Comparisons among different shadow simulators on the cumulative complexity are presented in Figure 7.

It can be seen in Figure 7 that the cumulative taskload complexity largely follows that of the input scenario when the Basic Simulator 1 was used as the shadow simulator. When ATOMS was used as the shadow simulator, there was a greater effect on taskload complexity, as can be seen in Figure 7, particularly during the first 25-minute period when the taskload complexity was higher than the baseline, a time when the baseline complexity was lower than the target level.

However, the complexity could not be pushed down enough to the level which the Basic Simulator 2 has achieved in the second 25 minutes period. In fact, it has resulted in an increase in complexity for parts of this period when compared to the baseline. The most dramatic change on the cumulative taskload complexity can be seen when using Basic Simulator 2, particularly for the period around 8:30. If we see Figure V, we can see a significant drop in the taskload complexity when using Basic Simulator 2 during this period relative to the other shadow simulators and the baseline.

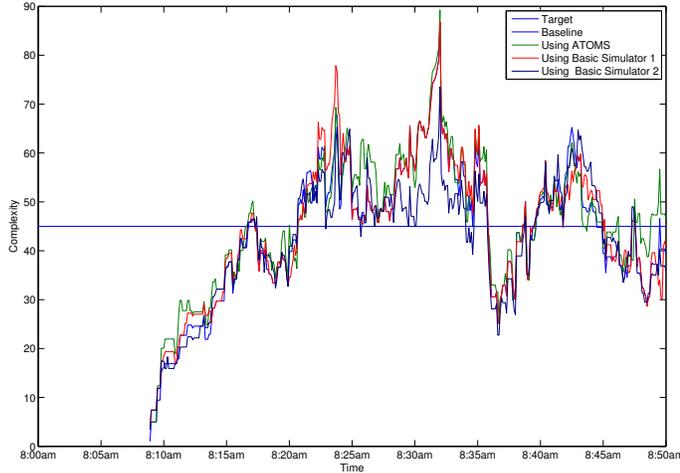


Fig. 6. Taskload complexity as measured by the ATOMS real-time simulation for each experiment

As shown in both Figure V and 7, it seems that the Basic Simulator 2 has a better overall performance among the three shadow simulators.

TABLE IV
AVERAGE DEVIATION FROM THE TARGET LEVEL OF COMPLEXITY

Experiment	Average deviation for 10 minutes starting at					Overall average
	8:00	8:10	8:20	8:30	8:40	
Baseline	37.24	14.16	7.90	12.58	7.36	11.21
ABS1	34.88	12.88	9.47	13.15	7.52	11.40
ABS2	34.93	13.96	6.27	8.44	8.44	9.97
AA	35.37	11.37	10.00	13.29	4.98	10.57

The average deviation of the complexity measurements from the target level of complexity is calculated using Equation 2 for the entire 50 minute session and dividing the obtained value by the total number of measurements, N . The average deviation from the target level of complexity for each experiment can be seen in Table IV. From this table, we see that using Basic Simulator 1 and ATOMS as the shadow simulator resulted in an overall average deviation greater than that of the unaltered input scenario, the baseline. Although ATOMS as the shadow simulator achieved smaller variations in the first 20 minutes, the Basic Simulator 2 has much smaller variations for the remaining periods which corresponds to the observations from Figure V.

It is expected that using Basic Simulator 1 as the shadow simulator would yield results with a greater average deviation

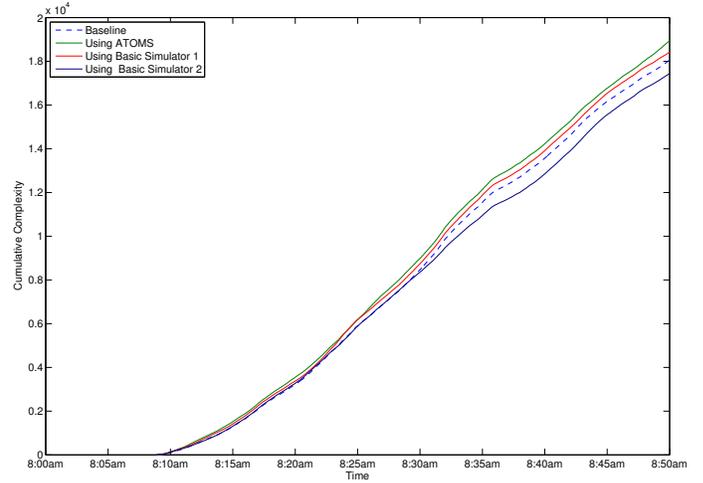


Fig. 7. Cumulative taskload complexity as measured by the ATOMS real-time simulation for each experiment

than Basic Simulator 2 as the fidelity of Basic Simulator 2 is greater than that of Basic Simulator 1. Using ATOMS as the shadow simulator may have resulted in a greater average deviation due to the time constraints placed on the optimisation. Figure V demonstrates that the number of evaluations completed when using ATOMS as the shadow simulator is significantly smaller than that when using the two basic simulators. This is expected as ATOMS is a significantly higher fidelity simulator and therefore requires more time to complete simulations consisting of a similar number of aircraft when compared to the basic simulators. ATOMS can achieve better results when the number of aircraft (agents) are smaller. When number of aircraft (agents) increases, the explored search space by DE with ATOMS is limited by the time constraint. The number of aircraft (agents) affect the other two basic simulators somehow as shown in Figure V, but the built-in low fidelity model enables DE to handle more aircraft and to explore more of the search space.

The experiments were repeated with varying parameters for the differential evolution to understand the sensitivity of these parameters on the system. The CR was varied to three levels: 0.3, 0.6 and 0.9; and the F was also varied between three levels: 0.1, 0.3 and 0.5; giving a total of nine different combinations. The three experiments were repeated with 20 runs for each combination of CR and F. Due to resource constraints these experiments were conducted on a number of different computers and as such the two minute optimisation limit would not have given fair results, particularly those using ATOMS as the shadow simulator. So the optimisation was instead limited to a maximum number of evaluations. The average number of evaluations for each shadow simulator was determined from Figure V and then halved. The resulting figure was the evaluation limit used in place of the two minute limit for the respective shadow simulator. The average of the average deviations from the target for each experiment and CR and F combination can be seen in Table V. From this table

it can be seen that for all but one combination, the ATOMS shadow simulator has the lowest average of average deviations from the target.

TABLE V
AVERAGE OF AVERAGE DEVIATIONS FROM TARGET LEVEL OF COMPLEXITY FOR DIFFERENT DIFFERENTIAL EVOLUTION PARAMETERS USING THE THREE SHADOW SIMULATORS

CR	F	BS1	BS2	ATOMS
0.3	0.1	11.64 ±1.34	11.28 ±0.61	11.20 ±1.49
0.6	0.1	11.58 ±0.87	11.56 ±0.74	11.06 ±1.38
0.9	0.1	11.90 ±1.04	11.65 ±0.87	10.70 ±0.68
0.3	0.3	11.54 ±1.19	11.52 ±1.08	11.10 ±0.84
0.6	0.3	11.32 ±1.15	11.49 ±1.05	11.62 ±1.63
0.9	0.3	11.53 ±1.30	11.40 ±1.23	10.95 ±1.40
0.3	0.5	11.33 ±1.26	11.65 ±0.77	10.56 ±0.78
0.6	0.5	11.88 ±1.17	11.73 ±0.78	11.43 ±1.72
0.9	0.5	11.49 ±1.19	11.50 ±0.74	10.95 ±0.92

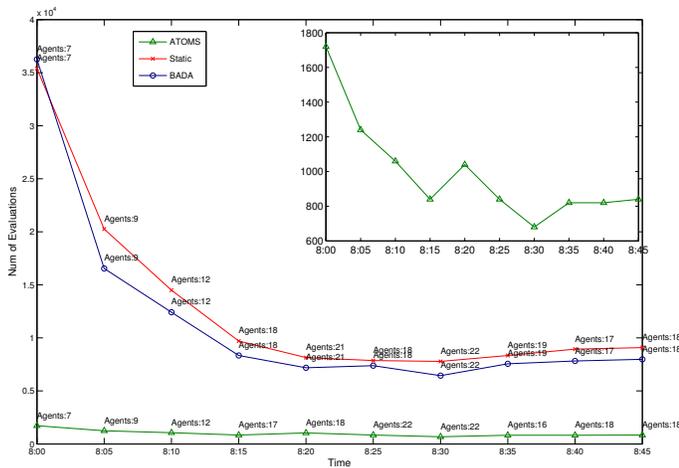


Fig. 8. Number of evaluation completed by each shadow simulation method for each snapshot

VI. CONCLUSION

In this paper, we proposed an approach to optimise ATC taskload in real time by using shadow simulators and DE. Three simulators with different fidelities are used as shadow simulators. As demonstrated by the results, DE can utilise all shadow simulators to produce acceptable solutions for adjusting ATC taskload in real time. However, the performance of each shadow simulator is different from the others.

A high fidelity simulator can help DE to achieve better results if no time constraint was applied. On the other hand, a lower fidelity simulator allows DE to evaluate more solutions; thus producing acceptable solutions in a short time window.

In summary, we demonstrate that a lower fidelity simulator can be beneficial when there is a time constraint on the optimisation process.

REFERENCES

[1] 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.

[2] H. Abbass, D. Tucek, S. Kirby, and M. Ellejmi, "Brain-traffic integration," *Air Traffic International*, pp. 34–37, 2013.

[3] K. Tumer and A. Agogino, "Distributed agent-based air traffic flow management," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 2007, p. 255.

[4] K. D. Bilimoria, B. Sridhar, G. B. Chatterji, K. S. Sheth, and S. Grabbe, "Facet: Future atm concepts evaluation tool," *Air Traffic Control Quarterly*, vol. 9, no. 1, 2001.

[5] D. J. Bodoh and D. F. Wieland, "Performance experiments with the high level architecture and the total airport and airspace model (taam)," in *Proceedings of the seventeenth workshop on Parallel and distributed simulation*. IEEE Computer Society, 2003, p. 31.

[6] S. Alam, H. A. Abbass, and M. Barlow, "Atoms: Air traffic operations and management simulator," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 209–225, 2008.

[7] Y. Wan, S. Roy, and B. Lesieutre, "Uncertainty evaluation through mapping identification in intensive dynamic simulations," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 5, pp. 1094–1104, 2010.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] P. Kopardekar and S. Magyarits, "Measurement and prediction of dynamic density," in *Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar*, 2003.

[14] B. Sridhar, K. S. Sheth, and S. Grabbe, "Airspace complexity and its application in air traffic management," in *2nd USA/Europe Air Traffic Management R&D Seminar*, 1998.

[15] A. Nuic, *User Manual for the Base of Aircraft Data (BADA) — Revision 3.6*, 3rd ed., EUROCONTROL Experimental Centre, Bretigny, France, 7 2004.

[16] S. Alam, W. Zhao, J. Tang, C. Lokan, M. Ellejmi, S. Kirby, and H. Abbass, "Discovering delay patterns in arrival traffic with dynamic continuous descent approaches using co-evolutionary red teaming," *Air Traffic Control Quarterly*, vol. 20, no. 1, p. 47, 2012.

[17] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," *Advances in intelligent systems, fuzzy systems, evolutionary computation*, vol. 10, pp. 293–298, 2002.