Algorithm Evaluation Technical Report

Document information	
Project Title	ASHiCS
Project Number	E.02.05
Project Manager	Rob Alexander, University of York
Deliverable Name	Algorithm Evaluation Technical Report
Deliverable ID	D 3.1
Edition	00.00.03
Template Version	03.00.00
Task contributors	
University of York	

Abstract

An assessment of the current state and future direction of the ASHiCS search harness algorithms is given, alongside details of new additions to the Stage 2 scenario (simulation of severe weather impact) that allow us to increase the complexity of the air sector and workload of controllers. By applying the NASA complexity measures originally intended for multi-sector planning to Stage 2's single sector, we gain an objective measure of complexity which we can add to the compound risk measures already used in our fitness function. Our initial results suggest that the search is able to find targeted conflicts within the increased complexity. However, there seems to be an increase in the frequency of very severe conflicts which can sometimes skew the heuristics away from an intended target (this is on-going work). Finally we propose a solution to the event frequency problem by using a two stage search process. The first part enables discovery, the second part will provide statistical context. Our hope is that the statistical information can then be used in a safety analysis of the sector and the recommended implementation of safety barriers.

Authoring & Approval

Prepared By - *Authors of the document.*		
Name & Company	Position & Title	Date
Dr Kester Clegg	Research Associate	17/12/12
Dr R Alexander	Lecturer	

Reviewed By - *Reviewers internal to the project.*		
Name & Company	Position & Title	Date
Dr Rob Alexander	Lecturer	18/12/12

Reviewed By - *Other SESAR projects, Airspace Users, staff association, military, Industrial Support, other organisations.*		
Name & Company	Position & Title	Date
Prof Tim Kelly	Professor	19/12/12

Approved for submission to the SJU By - *Representatives of the company involved in the project.*		
Name & Company	Position & Title	Date
Dr Rob Alexander	Lecturer	19/12/12

Rejected By - *Representatives of the company involved in the project.*		
Name & Company	Position & Title	Date

Rational for rejection
None.

Document History

Edition	Date	Status	Author	Justification
00.00.01	24/12/12	Draft	Dr Kester Clegg	Initial draft
00.00.02	1/1/13	Submitted to Eurocontrol		Passed internal review
00.00.03	7/2/13	Final		Amended in response to comments from EUROCONTROL.

Intellectual Property Rights (foreground)

This deliverable consists of SJU foreground.

Table of Contents

EXECUTIVE SUMMARY **5**

1 INTRODUCTION **6**

1.1 PURPOSE OF THE DOCUMENT 6
1.2 INTENDED READERSHIP 6
1.3 INPUTS FROM OTHER PROJECTS 6
1.4 GLOSSARY OF TERMS 6
1.5 ACRONYMS AND TERMINOLOGY 7

2 AN OVERVIEW OF THE ASHICS SEARCH ALGORITHM **9**

2.1 INCREASING THE COMPLEXITY OF STAGE 2 9

3 ADDITIONS THE STAGE 2 SCENARIO **11**

3.1 IMPLEMENTATION OF THUNDERSTORM 11
3.2 STORM PARAMETERS 12
3.3 MEASURING COMPLEXITY 13
3.3.1 Initial results *14*

4 ISSUES RELATED TO QUANTITATIVE INFORMATION ABOUT SEARCH RESULTS **20**

4.1 TRYING TO DETERMINE EVENT FREQUENCY 20
4.2 MACRO DISCOVERY, MICRO SAMPLING? 21
4.3 PROPOSED "NEAR NEIGHBOUR SAMPLING" OF SEARCH RESULTS 21

5 CONCLUSION **23**

6 REFERENCES **24**

APPENDIX A **25**

A.1 IMPLEMENTATION OF STORM RESTRICTED ZONE 25

APPENDIX B **28**

A.2 COMPLEXITY FACTORS USED IN RAMS PLUS 28
A.2.1 Aircraft Count (ACT) *28*
A.2.2 Aircraft Density (DNS) *28*
A.2.3 Airspace Structure (STR) *28*
A.2.4 Climbing or Descending (CoD) *28*
A.2.5 Closest Points Approach (CPA) *28*
A.2.6 Aircraft Proximity to Sector Boundary (PRX) *29*
A.2.7 Variance in Directions of Flight (VDF) *29*
A.2.8 Convergence Angle (ANG) *29*
A.2.9 Conflict Near Sector Boundary (PRC) *29*
A.2.10 Aircraft Neighbouring Conflict (NBR) *29*
A.2.11 What is the NASA Complexity Algorithm? *30*

List of tables

There are no tables in this document.

List of figures

Figure 1: An example of dynamic resectorisation as a severe weather event changes traffic patterns [1]. 11
Figure 2: Various storm start and end points implemented as time-limited restricted zones in RAMS. 12
Figure 3: NASA Complexity Factors within RAMS Plus. 14
Figure 4: Complexity measure of final scenario over time. Series 1 shows raw data (multipliers set to one). Series 2 shows same scenario measured using NASA default multipliers shown in Figure 3... 15
Figure 5: Disruption due to severe weather simulation in Stage 2. The pictures show successive states of the airspace as the storm moves across the sector. 16
Figure 6: Fitness scores of ten best in each generation, using pop. size = 50, mutation range = 30s, weighting for CPLoss = 25. 17
Figure 7: Fitness scores of ten best in each generation, using pop. size = 50, mutation range = 90s, weighting for CPLoss = 10. 17
Figure 8: Fitness of best ten per generation, using pop. size = 100, mutation range = 20s, CPLoss weighting = 20. 18
Figure 9: Best scenario from search shown in Figure 8. The vectoring operations are shown in advance while in the final screenshot we can see CPLoss starting its emergency descent. 19

Executive summary

An assessment of the current state and future direction of the ASHiCS search harness algorithms is given, alongside details of new additions to the Stage 2 scenario (simulation of severe weather impact) that allow us to increase the complexity of the air sector and workload of controllers. By applying the NASA complexity measures originally intended for multi-sector planning to Stage 2's single en-route sector, we gain an objective measure of complexity which we can add to the compound risk measures already used in our fitness function.

Our initial results suggest that the search is able to find targeted conflicts within the increased complexity. However, there seems to be an increase in the frequency of very severe conflicts which can sometimes skew the heuristics away from an intended target (this is on-going work). Finally we propose a solution to the event frequency problem by using a two stage search process. The first part enables discovery, the second part will provide statistical context. Our hope is that the statistical information can then be used in a safety analysis of the sector and the recommended implementation of safety barriers.

1 Introduction

ASHiCS has demonstrated the use of search heuristics on simulation outputs to find scenarios of high risk for a given air sector. Our previous reports have described how weighted heuristics are able to focus on specific incident types, flight paths or aircraft so that the search can effectively target those within the solution space.

As ASHiCS does not have a specific case study on which to do comparisons of algorithm types, and given that we have already discussed the choice of risk measures (in D2.1) and the search's implementation and performance (in D2.2), in this deliverable we discuss the effects of increasing the complexity of the Stage 2 scenario through the addition of a severe weather simulation. We end by looking at future work to explore a solution to the problem of not having contextual information related to the search result, as the Stage 2 solution space cannot be exhaustively searched due to its size.

We propose that the search uses a two stage process. The initial search is guided by compound risk measures and the NASA complexity function, with weightings applied to the focus of interest. Once a result is discovered that interests the safety analyst, we propose that an intensive sampling is conducted around the near neighbourhood of the result. This means sampling occurs within a specific range around the entry times of aircraft with all other factors remaining constant. This would permit reasonable statistical information about the near context of the result, in particular whether specific aircraft are critical to the event concerned or whether there is a large number of near configurations that will result in similar outcomes.

This approach should mean that the search result is accompanied by a degree of confidence relating the likelihood of the search outcome given a variance in the input configuration. Such information would aid safety analysts who need this type of information to assess the implementation, cost and effectiveness of safety barriers.

1.1 Purpose of the document

To describe the algorithms used by the ASHiCS project to search for high levels of risk in a complex air traffic scenario.

1.2 Intended readership

This document's intended readership are ATM planners, modellers and safety analysts interested in automated searches for hazards using fast time ATM simulation software such as RAMS Plus.

1.3 Inputs from other projects

We have had no input from other projects or technical advisors for this deliverable.

1.4 Glossary of terms

Evolutionary search

Form of search algorithm that uses selective pressure and mutation to improve a population of candidate solutions over many generations.

Evolutionary strategy

Pragmatics of evolutionary search relating to rate, range and restrictions of mutation, crossover, combination or other means of furthering good genes, population size, fitness selection policy, number of generations, etc.

Fitness function

Process used to select individuals from the population of candidate solutions by a ranking score assigning to each solution.

Search heuristics

Means of effectively guiding the search algorithm through the search space.

Search Landscape

Imaginary visualisation of a search space in which the fitness of each individual in a set's population is shown as a measure of vertical height with individuals of similar fitness being placed close together. By plotting a curve between the heights of individuals a landscape can be drawn with peaks representing areas in the solution space that contain the fittest individuals. This visualisation is extremely pervasive within the search literature, however it has many theoretical problems: i) there are no horizontal axis which can place the individuals geometrically within a set so the notion of similar solutions lying close to one another is hard to justify; ii) the visualisation breaks down completely in high dimensionality (i.e. where many factors may affect fitness levels), as there are likely to be areas of "impossible" gene combinations that cannot be realised in a solution.

Weighted fitness function

In a multi-objective fitness function, it is possible to assign greater "weight" to certain factors within the fitness evaluation so that the search favours solutions presenting those characteristics over others.

1.5 Acronyms and Terminology

Term	Definition
ANSP	Air Navigation Services Providers
API	Advanced Programming Interface
ARMS	Aviation Risk Management Solutions (working group)
ASHiCS	Automating the Search for Hazards in Complex Systems
ATC	Air Traffic Control
ATCos	Air Traffic Controllers
ATM:	Air Traffic Management
ATOMS	Air Traffic Operations and Management Simulator
CGP	Cartesian Genetic Programming
CFIT	Controlled flight into terrain
CRT	Computational Red Teaming
CPA	Closest point of approach (between two aircraft)
CSV	Comma separated values
DFS	Deutsche Flugsicherung
EC	Evolutionary Computation
ECAC	European Civil Aviation Conference
eDEP	Early Demonstration & Evaluation Platform
EFT	Evolutionary functional testing

Term	Definition
ERC	Event Risk Classification
ESD	Event sequence diagram
FAA	Federal Aviation Administration
FL	Flight level (given in hundreds of feet)
FTS	Fast time simulation
IRP	Integrated Risk Picture
ISA Software	Innovation for Sustainable Aviation Software
MOGA	Multi-Objective Genetic Algorithms
NASA	National Aeronautics and Space Administration
NATS	National Air Traffic Service (UK)
NSGA	Non-dominated sorting genetic algorithm
PUMA	Performance and Usability Modelling
RAMS	Re-organized ATC Mathematical Simulator
RTS	Real time simulation
SDAT	Sector Design and Analysis Tool (FAA)
SESAR	Single European Sky ATM Research Programme
SID	Standard instrument departure
SoS	System of Systems
SJU	SESAR Joint Undertaking (Agency of the European Commission)
SJU Work Programme	The programme which addresses all activities of the SESAR Joint Undertaking Agency.
SESAR Programme	The programme which defines the Research and Development activities and Projects for the SJU.
SSE	Safety significant event
SSMT	System Safety Management Transformation (internal program of FAA)
STAR	Standard arrival
TAAM	Total Airspace and Airport Modeller
TMA	Terminal Area

2 An Overview of the ASHiCS Search Algorithm

The various components of the ASHiCS search harness have been described in previous deliverables. We have gone into detail into how the basket of risk measures is calculated (D2.1) and the mechanics of constraining the random mutation operator so that reasonable variants of a scenario are generated (such as restricting the separation between aircraft on the same flight path, see D2.2).

When trying to design a search process, constraints for the search are often required to limit the amount of time to reach a solution or to ensure the solution meets certain criteria. In the Stage 2 scenario for ASHiCS, it was decided that we could allow variable aircraft distribution between the two major flight paths, as we felt that the liberal distribution and timing of aircraft was critical in exploring the solution space. However, given the combinatoric possibilities that this decision permits, we realised that the solution space was far too large to search exhaustively and that this would result us being unable to say much about the context of the search results. In our last deliverable (D2.2) we explored using linear regression and potentially principal components analysis (PCA) to reduce the size of the solution space to areas that are likely to provide high fitness scenarios. This work was unsuccessful in finding a link between the input variables and fitness scores that would have allowed us to try and reduce the dimensionality of the solution space, partly due to the difficulty of aligning the input data so that it would amenable to statistical analysis such as PCA. After an internal review, we decided the amount of work this entailed could not be justified, given that we had not had any indication that our initial efforts using forms of linear regression would be successful. However, we believe that the benefits of dimension reduction (i.e. allowing an exhaustive or near exhaustive search of the solution domain) can be achieved using a different approach.

In the following sections, we describe our current process and our proposals to extend the scenario to increase its complexity by simulating the impact of a severe weather disruption to air traffic. We then describe how we believe the search process can be split into two stages. The first stage permits discovery of high risk, highly complex scenarios with the option to target certain conflicts. The second stage provides context to the first result such that we can examine “near” variant scenarios for risk and types of conflict, and gauge their levels of severity or similarity.

2.1 Increasing the complexity of Stage 2

The Stage 2 scenario has been described in previous deliverables (please see D2.2 in particular). However, it was noted in meetings with SJU representatives that they felt the scenario although potentially complex did not really represent a system of systems (SoS). Within the modelling environment provided by RAMS Plus (modelling traffic flow and control of air sectors), it is difficult to incorporate other subsystem information feeds into a RAMS scenario without extensive bespoke development provided by ISA Software Ltd.[1] In addition to this there would need to be extensive domain consultation to establish what type of information subsystems might feed directly into an ATC of future systems and to work out what impact erroneous information might have on the handling of aircraft by ATC.

Given the issues with developing a true SoS of this type within the RAMS Plus environment, we decided instead to try to increase the complexity of the scenarios by enabling another configurable subsystem that would interact with the simulation yet form part of the input configuration to the search. The easiest method to achieve this was to implement a weather effect within the scenario that would impact traffic flow, workloads and potentially associated risk. Bringing in more inputs to the simulation naturally increases the search space considerably and so some restraints are applied to restrict the degree of freedom allowed to the search.

The second issue with trying to implement weather effects within the scenario is that we needed to have some objective measure that could demonstrate greater complexity had been achieved by bringing in another configurable subsystem for the search to exploit. It could be possible that the weather effects would have very little impact on complexity, and by extension on safety, and we

[1] We have previously commented on some of the short-comings of the SIMC- API provided by ISA Software to the ASHiCS project. The reliability problems were such that we were unable to pursue the development of a subsystem service provider of information to feed into RAMS simulations using SIMC.

needed to have some measure that the search could use to drive it towards scenarios containing both greater complexity *and* higher risk.

3 Additions the Stage 2 Scenario

While there is already an add-on to RAMS Plus that permits the simulation of wind (the ATMOS weather server), there is very little information on how to use it and no indication of how its effects can be judged (other than looking at arrival delays of aircraft, not an area we were interested in). Furthermore, it seemed that the ATMOS weather server is intended more to model the effects of constant, strong head or cross winds over large areas, rather than a specific weather event that could cause disruption to the management of aircraft over a given sector.

Our solution was to work in terms of the impact of severe weather events on ATM, which essentially result in aircraft being vectored around or over the weather as if it were enclosed in a restricted no-fly zone. Restricted zones can be modelled manually in RAMS relatively easily and they can be given time limits for their operation. There are no limits to their shape, height or the number of zones used. It also possible to specify whether aircraft vector around bad weather or attempt to fly over it. After some initial research on the common sizes of thunderstorms and their maximum speeds, we decided we could create a series of restricted zones that would have the same impact as a thunderstorm moving rapidly across our en-route sector in Stage 2.

3.1 Implementation of thunderstorm

Real world thunderstorms are interesting events to manage with respect to ATM as they represent a moving restricted zone that can change shape, speed and severity. Figure 1 shows how an approaching storm changes shape as it moves across several air sectors. In this example, the disruption to the traffic causes the controllers to dynamically re-sector the air space to account for the changes in traffic flow.

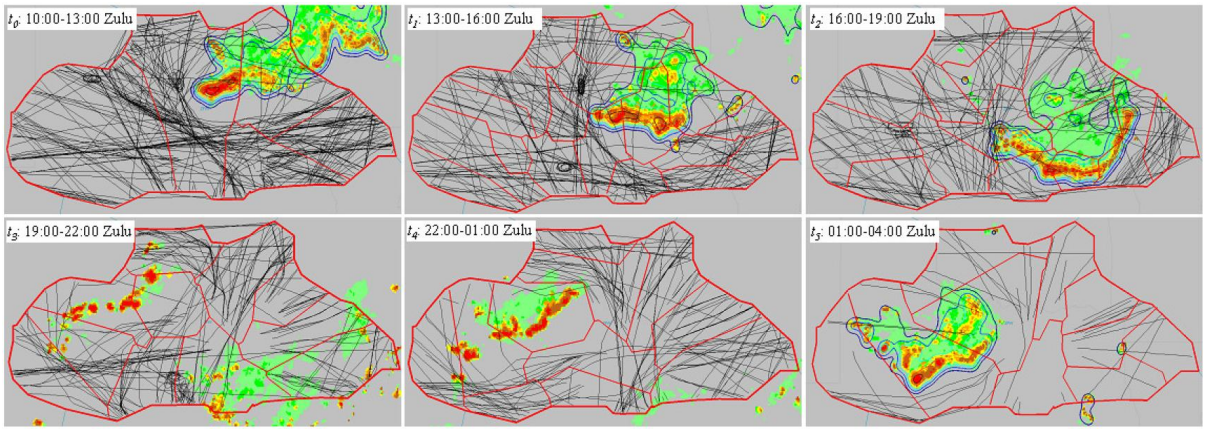

Figure 1: An example of dynamic resectorisation as a severe weather event changes traffic patterns [1].

As part of the process of investigating the shape and speed of thunderstorms so that we could implement them into our Stage 2 scenario, we also came across the technique of dynamic resectorisation to manage the changes in controller workload due rerouting traffic around severe weather (Figure 1). This was of interest to us, as we were already aware of the multi-sector planner functions that had been implemented in RAMS Plus. What we hadn't been aware of is that as part of this function, ISA Software had also implemented the NASA Complexity Factors as a means of automating a multi-sector planning. These factors are used to trigger the multi-sector planning function given certain thresholds of complexity. Such a measure proves extremely convenient for us to use in our fitness function (see Section 3.3).

Clearly attempting to realistically simulate the changing shape of the severe weather event shown in Figure 1 would be beyond the resources available to ASHiCS. In addition, we would be faced with a decision related to the shape configuration parameters that would need to be either restricted or exposed to a random mutation operator as part of the search function. Allowing the search to select a wide variety of storm behaviour would further increase our problems related to the total size of the

solution space (see Section 4). Some means of restricting the combinatorial possibilities is needed to ensure the solution space remains tractable. We decided that the principal increases in complexity were due to the rerouting of aircraft around the storm, so attempting to reproduce in high fidelity the shape fluctuations of real world weather events is not as important as representing the storm as something which is mobile over a given time frame, creating new traffic flows and perhaps unexpected conflicts.

3.2 Storm parameters

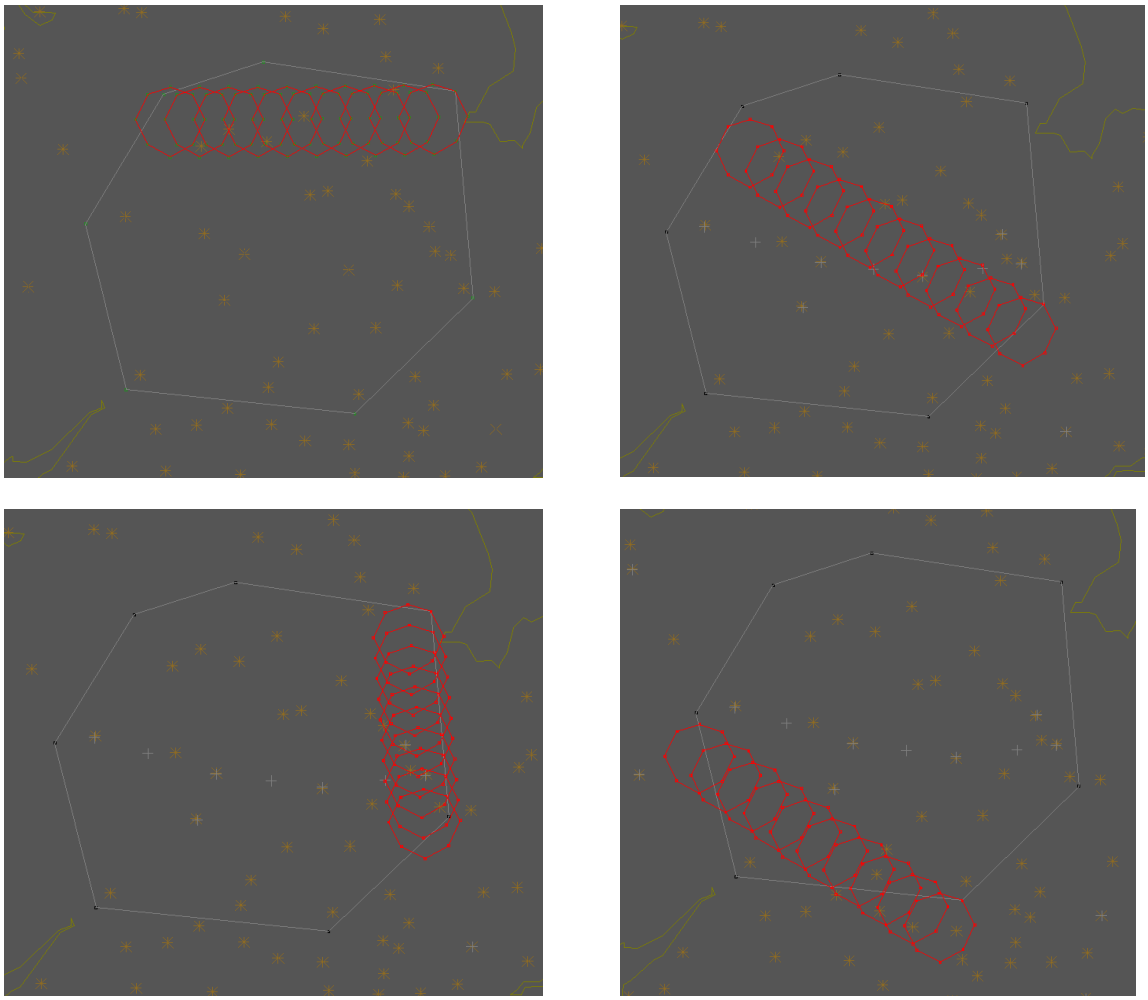

Figure 2: Various storm start and end points implemented as time-limited restricted zones in RAMS.

Using our existing air sector, an easy implementation was to allow the storm a limited series of trajectories across the sector by selecting straight line trajectories from one waypoint to another that defined the sector corners. By determining the widest distance between the air sector corners, we could work out what would be the maximum possible speed of the storm over a two hour period. This turns out to be 50km/h, which is within the range of fast moving winds that accompany severe weather events. However, if we selected a trajectory between two adjacent corners, then the distance the storm moves over the two hour period is greatly reduced and likewise its speed.

The seven waypoints that form the corners of the air sector can be selected by a random operator as start and end points for the storm's trajectory, a solution that also selects the storm's direction of travel across the sector.

The next part of our implementation was to determine the size of the storm. As previously mentioned, we thought it would be too complicated for a project of this length and manpower to implement a shape changing representation. We selected a fixed polygon with a maximum diameter of 18km (this

figure was based on information from the National Oceanic Atmosphere Administration website[2]). The polygon takes its top left hand corner as having the same co-ordinates as the initial waypoint. A straight line is drawn between this waypoint and the end waypoint. That distance is divided into ten segments, the starting point of each forming the same top left hand corner of the storm polygon as it moves along its path. We then calculate the corners of each of the storm polygons and create a series of ten restricted zones with those co-ordinates. The co-ordinates form corners which are written to `corner.dat`, with the corners linking to form polygons that are written to `boundary.dat`. Each restricted zone is listed in `restrictedboundary.dat` and finally the start and end time of each zone is recorded in `restriction.dat`. The code listing for the implementation (apart from the file writing) is shown in Appendix A.

Once the trajectory of the storm is calculated from the initial random operator's selection of waypoints, we can determine the restricted zone time limits that define the storm's progress across the sector. Currently, we simulate for a period of two hours. Our current implementation therefore just divides the two hour period into ten zones that last for 12 minutes each. The further apart the waypoints, the faster the storm moves across the sector. At this point the storm configuration is fixed (in a similar way to the distribution of aircraft on flight paths, or the waypoint that triggers the emergency cabin pressure event to CPLoss). However, we are planning to allow the search the freedom to choose a start time for the storm's entry into the sector, in which case the evolutionary operator will be able to mutate the storm's entry time in a similar fashion to how it currently mutates the entry times of aircraft. Naturally this would also affect the storm speed, potentially making it travel much more slowly than it does currently (as the storm always starts at the beginning of the simulation), potentially causing a wider range of disruption to aircraft.

We ran tests on the storm implementation by running the search as before, with a combined risk measure that contained weightings towards conflicts with CPLoss. The fitness function now comprises:

- total number of conflicts (weighted in favour of scenarios that included CPLoss);
- total number of resolutions by ATCo;
- conflict separation percentage (only worst case used);
- total ATCo task workload (measured in seconds);
- NASA sector complexity measure (highest score).

The search results found scenarios that generated increased workloads for the controller due to extensive vectoring of aircraft around the storm. While these results were pleasing, we had no guarantee that what we had added to the sector was creating greater complexity in the sense of making the controller's job more difficult, potentially reducing safety and increasing risk. Rather than attempt to adjust our current heuristics that looked for particular conflicts, or to seek a new measure of risk that might capture better the increased workloads, we decided to add the NASA Complexity Factors to our fitness function to give us an objective, user-tuneable measure of complexity.

3.3 Measuring complexity

The complexity function within RAMS Plus is intended to be used as part of the multi-sector planning tools. The main task of a multi-sector planner (MSP) is

"to resolve workload imbalances in the sectors under its responsibility ahead in time of the planning controller and the tactical controller.. Controller workload is often measured as the number of tasks to be performed by the controller. This measurement must consider the controller working method, (far beyond the scope of this manual) and therefore may not be easily measured. As many of the ATC specialists consider the sector complexity as the major source of the controller workload, the MSP has the optional functionality to compute ATC complexity to evaluate the controller loadings." [2].

It is this functionality, which can be applied to a single air sector (with certain changes to the configuration) that we will use to compute the complexity of the scenario.

The complexity measure itself is comprised of a set of measurements (see A.2 for the full definition) that look ahead a certain amount at each instant in time to calculate the predicted complexity. For

[2] http://www.noaawatch.gov/themes/severe.php

single sector assessments, this look-ahead time is reduced to zero, as there is no need to re-sector the air space. Each complexity factor can be given a weighting (described in the RAMS Plus manual as a "multiplier", but which works in exactly the same way as weightings for our risk measures, see D2.2) to increase or decrease its influence on the final complexity score. This means that the search is able to use these weightings to direct to discover scenario whose complexity is more likely to contain measurements of a particular type. The currently implemented complexity factors in RAMS Plus are shown in Figure 3 (and fully defined in A.2). For the current implementation we have not changed the weightings from their default NASA values.

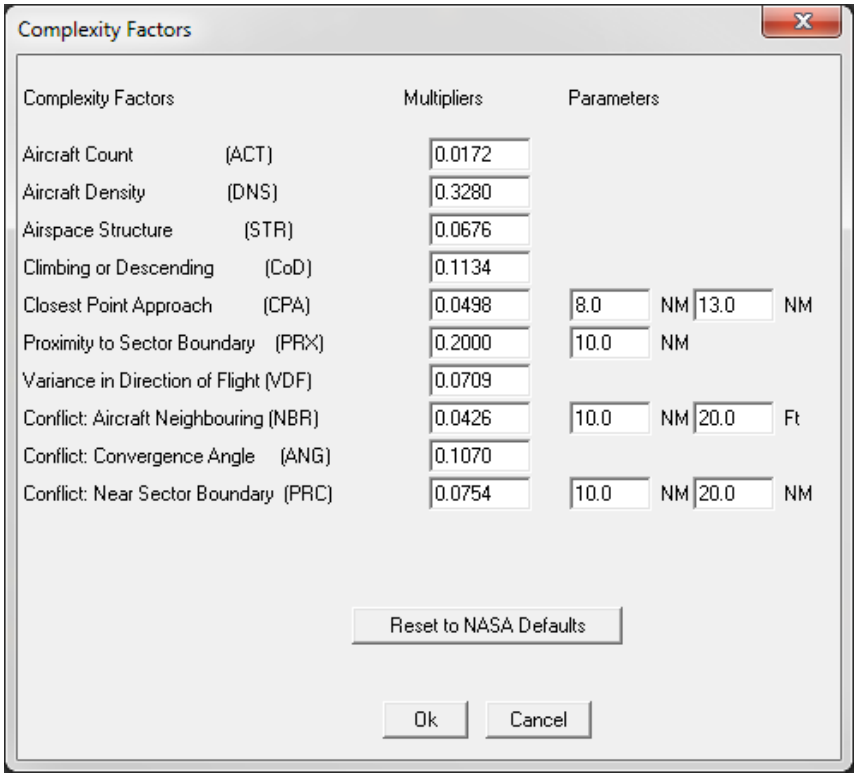

Figure 3: NASA Complexity Factors within RAMS Plus.

Because the complexity factors are calculated every *n* instants (where *n* is a number of seconds, in our case n=60) during the simulation, we end up with a series of measurements that chart the rise and fall in complexity over the time. Unfortunately, and in a similar fashion to some of the issue we raise over the generic risk measures in Section 4, a rise in this value for short time may represent a critical period that would be of interest to us to explore. However, as this makes parsing the log output file difficult, we take the greatest value recorded for the combined complexity score. It may be that this level of complexity lasted for several minutes or an hour; currently we do not have a way of estimating the greater severity of disruption and difficulty for the controller over long periods of high complexity, other than trying to weight the overall task load more heavily to factor in long periods of high complexity.

3.3.1 Initial results

A typical graph of the complexity from a final search result is shown in Figure 4. From our initial experiments, it would seem that this shape of chart is relatively typical. Rather than a series of peaks, we tend to see sustained periods of high complexity as the storm moves across the scenario. Of course, these figures depend heavily on the distribution and entry times of aircraft, and the trajectory of the storm. But given our limited domain experience with complexity measures and the difficulties of implementing a more sophisticated measure, we think that the plot's shape suggests it is reasonable use the highest value recorded during the simulation. This means we can very easily parse the `complexityfunction.out.1` log file without resorting to heuristics that would sum the total length of time at high complexity over a given threshold.

Using the NASA default multipliers gives a relatively low total complexity value which we are still experimenting with to see if it requires weighting when incorporated with our risk measures. The introduction of another large measure naturally offsets previous weightings for specific targets, such as conflicts involving CPLoss (reducing their impact). Getting the right weightings for a particular search objective requires a lot of experimentation.

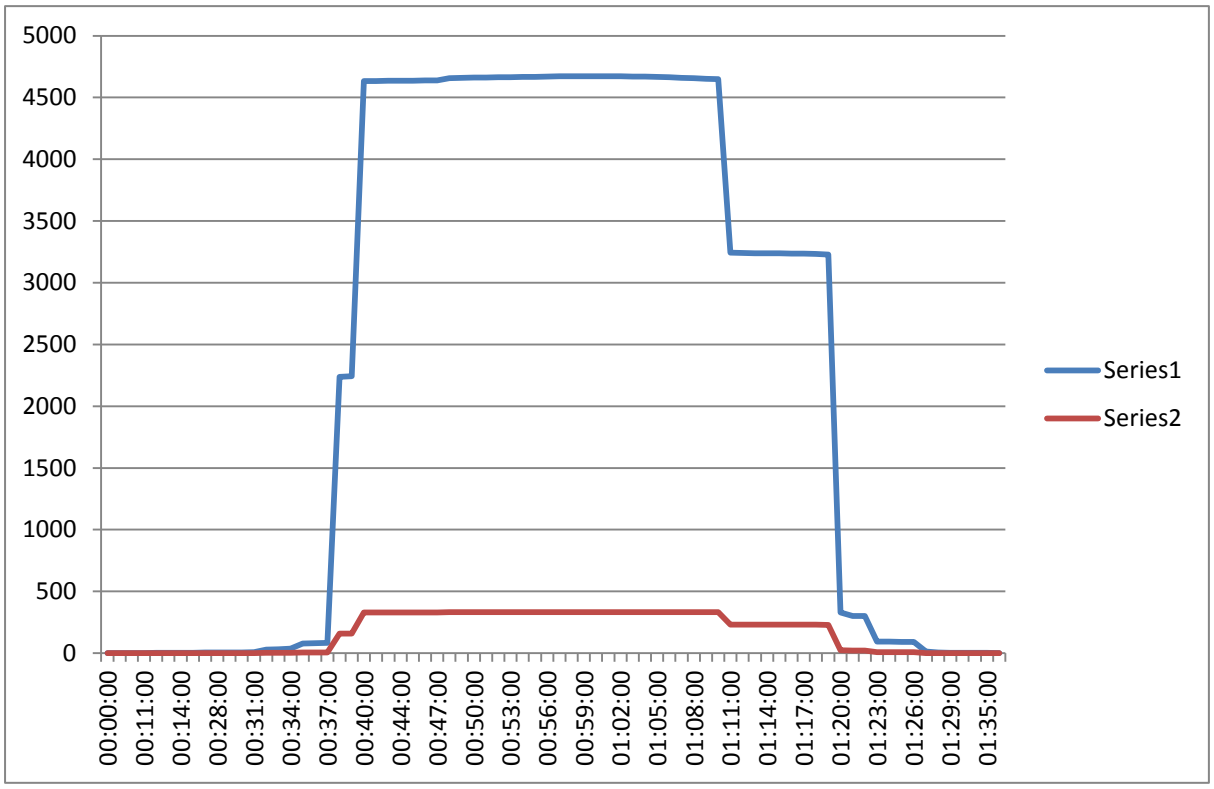

Figure 4: Complexity measure of final scenario over time. Series 1 shows raw data (multipliers set to one). Series 2 shows same scenario measured using NASA default multipliers shown in Figure 3.

Unfortunately the addition of the complexity measure means that the non-graphical version of RAMS Plus takes much longer to generate its output, requiring us to increase "pauses" in the code while we wait for all processes to finish writing to output files before our code starts to parse the outputs. This approximately doubles the search time; making an run with population size of 50 take about 20 hours to complete 300 generations (our previous population size of 100 now takes more than 24 hours to complete 300 generations).

The screenshots in Figure 5 below show the additional vectoring required to divert aircraft from the thunderstorm as it moves across a sector. Unfortunately we are not able to judge whether these vectoring operations by the controller are reasonable or not, particularly with regard to some vectoring instructions whose function seems solely to delay certain aircraft so that all aircraft can re-join their flight path in the original sequence after the storm has passed. We have been advised that this may not be realistic[3], however we are not able to change the default resolution or vectoring behaviour within RAMS Plus, other than the changing the priority of conflict types which is intended as part of the MSP functionality. Changing the controller rule base behaviour within RAMS Plus is beyond the scope of our expertise.

When we look at the fitness scores over generations (Figure 6 and Figure 7), we can see that the maximum fitness levels (shown as "plateaus", as the best scenario is carried over from one generation to the next) are reached rather rapidly, perhaps as a result of using smaller population sizes. Another possibility is that some of our scenarios have had very close conflicts, as measured in severity by the conflict separation percentage (CSP). This value, which indicates the amount of available separation available as a percentage (i.e. closest point of approach (CPA) divided by the

[3] Discussion with EUROCONTROL.

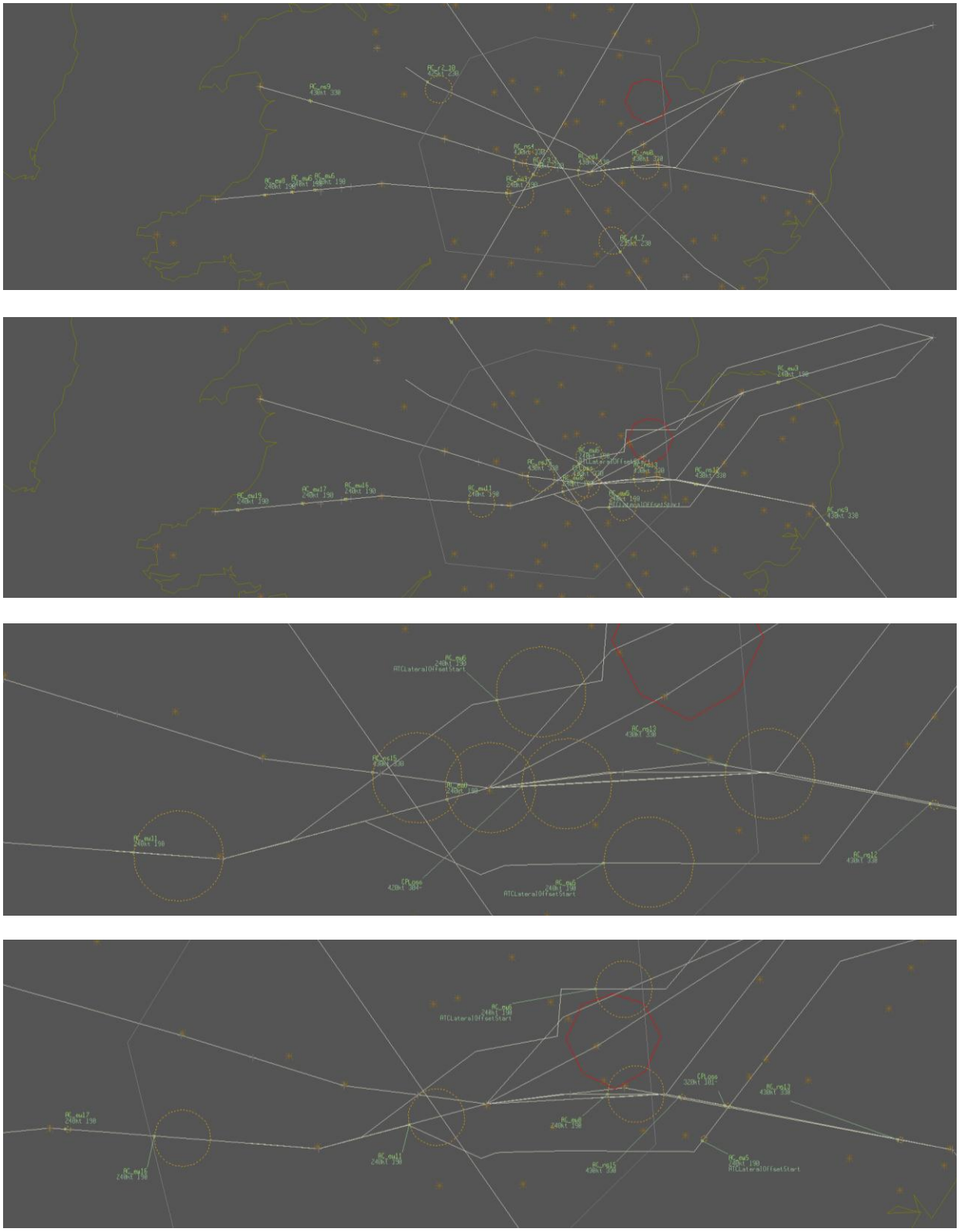

Figure 5: Disruption due to severe weather simulation in Stage 2. The pictures show successive states of the airspace as the storm moves across the sector.

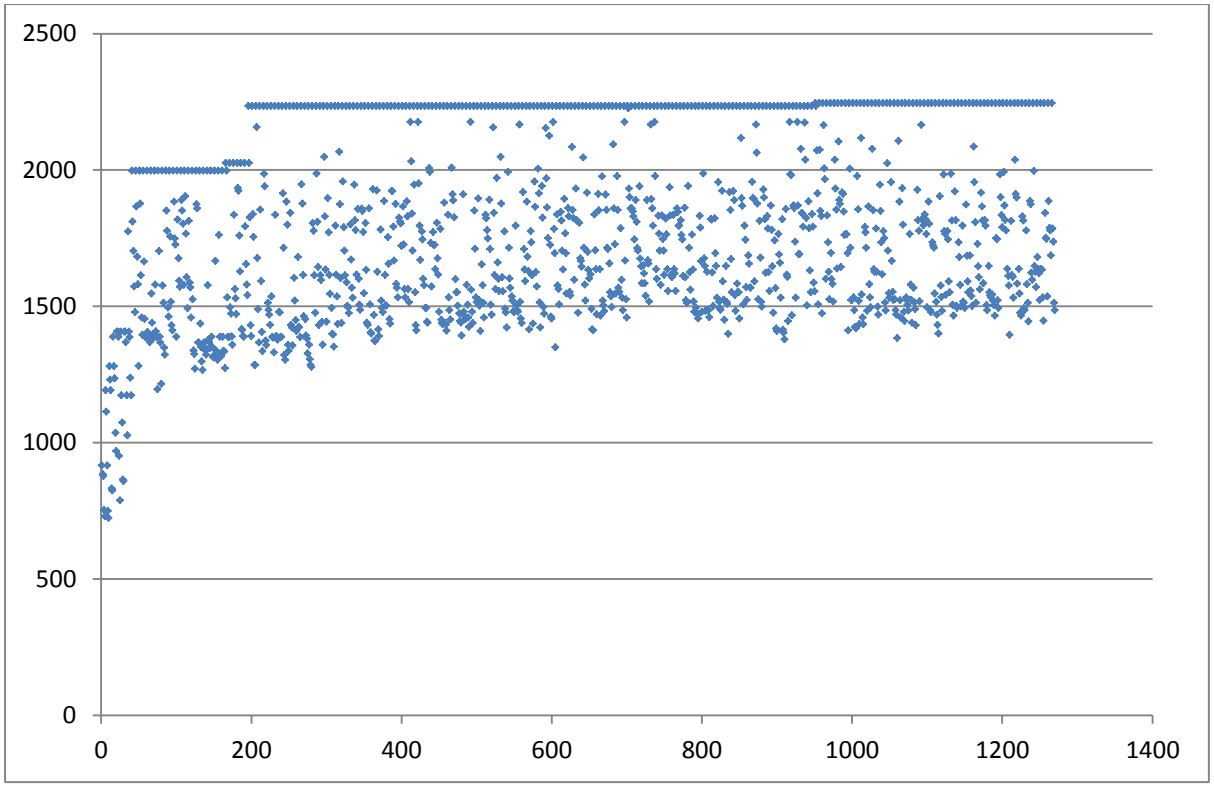

Figure 6: Fitness scores of ten best in each generation, using pop. size = 50, mutation range = 30s, weighting for CPLoss = 25.

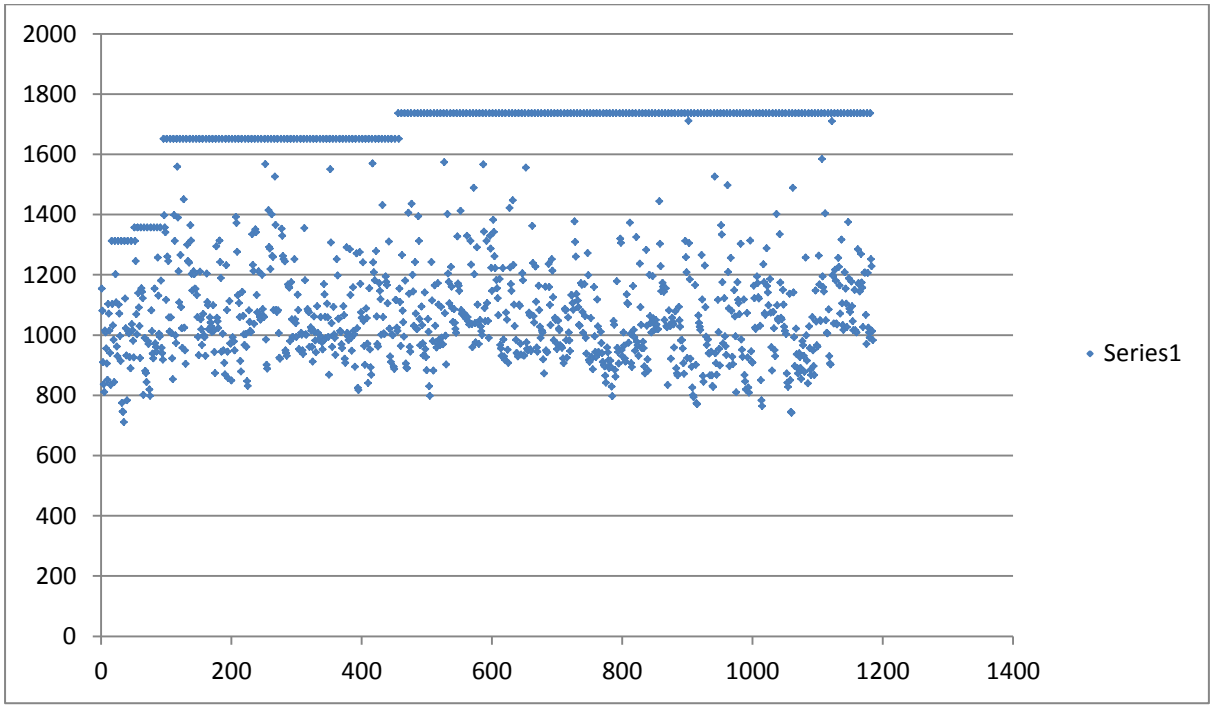

Figure 7: Fitness scores of ten best in each generation, using pop. size = 50, mutation range = 90s, weighting for CPLoss = 10.

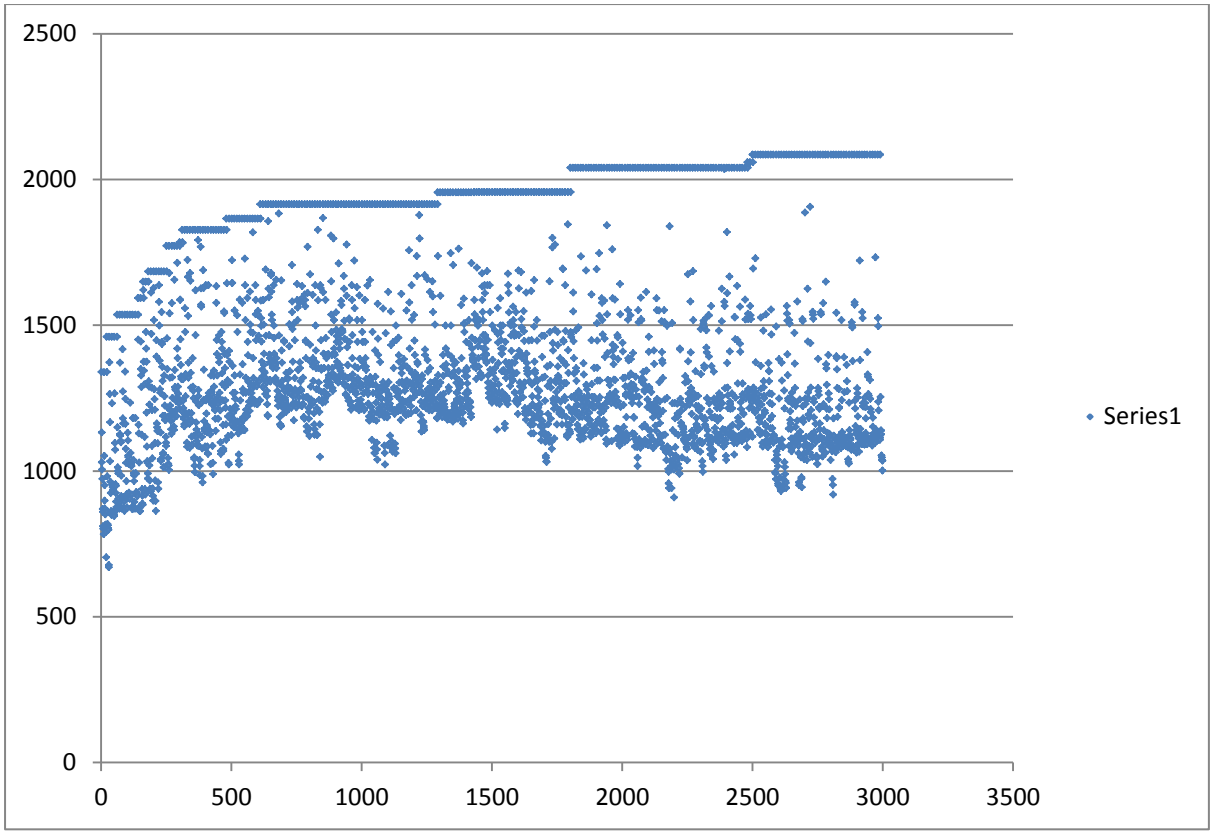

Figure 8: Fitness of best ten per generation, using pop. size = 100, mutation range = 20s, CPLoss weighting = 20.

minimum separation specified for the sector), can sometimes be a very low number if two aircraft are allowed to get exceptionally close to one another before being resolved. We apply a logarithmic multiplier to try and offset the wide variance in values that the CSP can take. However, even using a logarithmic multiplier is insufficient on occasion and in these cases the search can become "stuck", as the smallest mutation is likely to move the aircraft in conflict apart and result in a reduced fitness score for a scenario.

This is an area we are currently working on to see if there is an increased number of results that feature severe conflicts as a result of the additional complexity and disruption of the storm, or whether it's the case that the complexity measure itself has offset other weightings in the fitness function, perhaps preventing the search from progressing. It is worth noting that in Figure 7 we are again seeing destructive mutation (i.e. most mutations produce significantly worse scenarios), which is reminiscent of problems we had in D2.2. We are still investigating why this might be happening.

When we look at Figure 8, using a small mutation range of 20 seconds, but using a population size of 100 (i.e. double the number of simulations), we can see we get a much better fitness function curve that gradually increases over time. However we can also so that towards the end, we are again having mostly destructive mutations despite a continuous very gradual improvement in fitness of the best scenario. When we examine the best scenario discovered by the search, it's fairly obvious that the position of the storm moving from north to south on the western edge of the sector causes major disruption, particularly to the lower of the two main flight paths. The screenshots in Figure 9 show some of vectoring operations diverting aircraft from the path of the storm. Again we're seeing some rather odd vectoring decisions by the controller modelled within RAMS Plus but we are unable to comment on whether the ATC operations are realistic or correct. Note that CPLoss is shown as being diverted north around the storm in the 3rd screenshot (it has yet to commence its emergency descent). However, the diversion means that it then conflicts directly with AC_ns13, itself diverted around the southern side of the storm. Within a few moments CPLoss starts its descent where it will then conflict with both AC_ew7 and AC_ew6. The additional disruption and complexity the storm brings to the Stage 2 scenario is self-evident, but what we can also see in this series of screenshots is that we get

much better results using larger population sizes and small mutation ranges. The downside is the twofold increase in search time.

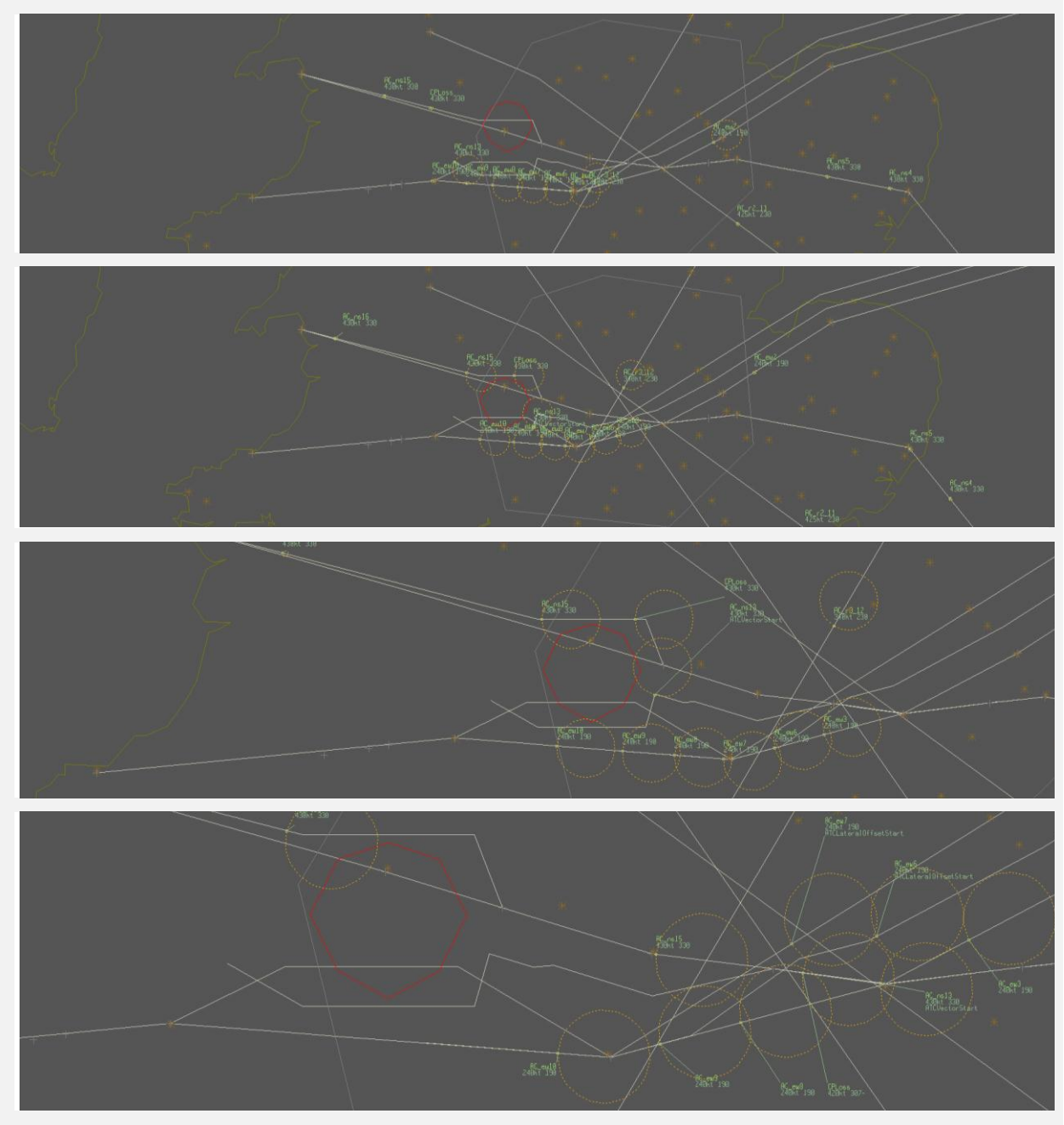

Figure 9: Best scenario from search shown in Figure 8. The vectoring operations are shown in advance while in the final screenshot we can see CPLoss starting its emergency descent.

4 Issues related to quantitative information about search results

As described in D2.1, ASHiCS uses a basket of risk measures to provide us with a compound fitness score with which we can judge a scenario's risk level. However, it should be noted that this is an overall risk assessment. It maybe that two very different scenarios, one with a single severe incident, and one with a consistently overloaded controller, have similar overall fitness scores with respect to their risk. This can be avoided by weighting the search heuristics such that a specific incident will score highly and make it more likely that a scenario containing that incident will be selected during the search. However the risk in taking that approach is that the search may focus on that incident type to the exclusion of others. By using a generic basket of risk measures, the search is not guided by particular incident types or conflicts with certain aircraft, and therefore has greater freedom to find perhaps unexpected combinations of conflicts.

We believe that both of these approaches are equally valid. The choice is determined by what the safety analyst wants to examine within the airspace. However, whichever way is chosen for the first approach, the result is unlikely to be of practical use to safety analysts without some contextual information. The barrier safety model proposed by SESAR and NextGen [3] suggests that safety hazards and incidents are constructed through a sequence of events that must pass through a series of barriers designed to prevent them. The barriers themselves are implemented under the assumptions that the events occur with a given likelihood, and therefore the barrier needs to be strongest and most robust against those safety events with the highest frequency or with the most potential to do harm. However, the cost of implementing such barriers is determined to a large extent by a simple formula that uses the cost of the outcome multiplied by the probability of the event occurring.

The question for ASHiCS is what does a search result represent? Is it a series of events with given likelihoods that can then be analysed to establish the probability of that incident occurring in a real world scenario? The answer would appear to be no unless the search parameters that govern the freedom of the search space are based on real word data. As the approach ASHiCS offers is far more likely to be adopted as part of the ATM planning phase, such data may not be available. But even if it is, and those who have constructed the scenario are relatively confident that they are exploring possible real world outcomes, it would still be very difficult to have confidence in the sum of probabilities attached to a sequence of simulated events discovered by the search. If we take the position that the simulation is an abstraction of the real world but still represents a theoretical outcome, the possibility remains that the search has discovered a rare input configuration that would be extremely unlikely to occur, or perhaps worse, that the search has found a single example of many similar outcomes, suggesting that there is a strong probability that a range of input configurations will nearly always result in the same hazardous outcome.

4.1 Trying to determine event frequency

Outcome frequencies are difficult determine without sampling data, and even here it may be the case that the simulation imposes bias on certain outcomes. However, we feel that a degree of inaccuracy is acceptable in order to discover more information about the context of a search result. In D2.2 we looked at some dimension reduction techniques using forms of linear regression over the input variables. The data in our analysis came from a million random samples of the Stage 2 scenario inputs. While this is a minute fraction of the combinatorial total, it was the most we could manage given the project resources. However, the approach had several weaknesses. Firstly its size was unrepresentative of the total solution space, which may have been a factor in our failure to find a relationship between high fitness scores and certain input variables. Secondly, the sampling required over a week's processing time, which would render the technique impractical in many working environments (for example, if mistakes are made and the sampling has to be repeated).

However, the approach of using sampling data to provide frequency information about event outcomes remains the only way of getting a statistical context to the search result. Rather than try to sample the whole solution space as we did in D2.3, we now propose to conduct sampling on a much smaller area *around* the search result.

In a similar fashion to keeping “near neighbour” fitness scores to check that the evolutionary mutation was improving the average fitness of the search population (see D2.2), we can extend sampling of the near neighbourhood of a search result to conduct an intensive exploration of the area around the search results. This may provide us with a number of high fitness scenarios, often varying very little in their fitness scores from the original result. However, by measuring the degree of mutation required to significantly drop the fitness scores we can get idea of how likely the event outcome is within a given input range. While this may sound relatively straightforward to implement, there remain a number of problems to be solved.

4.2 Macro discovery, micro sampling?

There are two main problems to solve if we wish to extensively sample the parameter space around a search result. Firstly is the range of the mutation operator. This is likely to be determined pragmatically: too large and we require high numbers of samples and run into sampling times becoming excessive, too small and we may get an inaccurate statistical picture of the event outcomes. The second problem is that we must restrict the search in a similar way we originally wanted when we tried dimension reduction.

For example, imagine that the search finds a scenario configuration in which two conflicts happen in short succession with the aircraft undergoing cabin pressure loss (CPLoss). In this case it would seem clear that we can exclude altering the entry times of many of the outlying aircraft, as changing their times will have no impact on the two CPLoss conflicts. We could “freeze” these, and concentrate sampling the input configuration around the entry times of those aircraft directly involved in the conflicts.

However, it is not always easy to identify all the factors that have gone into a scenario getting a high fitness score, particularly now that we include a raft of complexity measures. Some conflicts may involve multiple aircraft, and we must be careful not exclude aircraft whose position may have a subtle effect on the outcome of certain configurations. Choosing which aircraft to focus the second stage sampling process on requires domain expertise to be done effectively, perhaps using a set of rules established by an ATC specialist. Once this has been done, it may be possible to automate the selection of which aircraft to include in the second stage sampling and which to exclude. For ASHiCS, which is only intended to demonstrate a proof-of-concept of the approach, there may be neither the time nor resources to implement the automatic selection and exploration of aircraft for the second stage, and some manual intervention may be necessary. But while the introduction of domain expertise to draw up a rule base to allow the eventual automation of the second stage sampling process would be the best solution, in the absence of the necessary domain expertise we can adopt a halfway solution that perhaps lacks efficiency but is easiest to implement in the remaining time available to the project.

4.3 Proposed “near neighbour sampling” of search results

It is possible to convert our search harness to a random sampling process of the search space, as we did with the sample plotted in D2.2. That code was a fork from the main search harness code that uses the random operator to repeatedly sample the solution space (as the current search does on the first generation, and then subsequently for a proportion of every generation). We propose that the new implementation does not follow the same process, as we wish to instead to extensively sample the near neighbourhood of an existing solution. In effect, this is the equivalent of creating many mutations of one scenario using a small range for the mutation operator for all aircraft in the scenario. The other factors in the scenario remain unchanged (aircraft distribution across flight paths, the waypoint that triggers CPLoss, the start and end point of the severe weather no fly zones).

In order to try and automate the process as much as possible, we impose a generation limit of 300 after which point we keep the final solution discovered by the search. The second stage sampling then starts using the input configuration of the final solution for the mutation base. Instead of the usual search near neighbour limit of 3 mutants of the original, we create several thousand near mutants. Each of these is assessed for fitness using the same criteria as the original search. The mutation range that decides the size of the neighbourhood is yet to be decided, but we feel that we can use our previous results from D2.2 as an initial guide.

Our intention is to keep the input configurations for the highest scoring scenarios in the second stage, particularly any that outscore the original search result, so that we can try to use our domain knowledge to subjectively assess how close the events are that represent the moments of highest risk, and whether the mutants can provide anything new related to the original solution. For example, we may find that the vast majority of mutants are variants of the same conflicts, indicating that the conflict is likely to happen within a relatively broad range of input values. However if the nature of the conflicts changes, or there is wide spread of fitness values, this may indicate that the search discovered a relatively rare event. Exact frequency distributions for the conflicts discovered in the original scenario can only be estimated, as it is impossible for us to analyse in detail all samples taken around the original solution.

Our plan of work for the remaining time of the project is therefore to extend our search to a two-stage process, implementing an extensive random sampling of the near neighbourhood around the final result of the first stage. Although this will obviously increase the time required to get results, we believe what will be obtained will be of much greater value to safety analysis, as the contextual results will give confidence in the original search result and also allow analysts to look at near variants of the original result that carry more risk. These can be analysed to see how they differ and can give an input parameter range for aircraft times within which those levels of risk can be expected.

We hope that this approach will provide contextual information about the search results and therefore be of greater use to safety analysts. We are continuing to hope for assistance from ATC specialists who could advise on search topics for our heuristics, and help us use the complexity multipliers to direct the search towards certain conflict types or situations. If we could get such assistance and demonstrate the effectiveness of the approach, we believe we could raise the interest of the wider ATM safety community.

5 Conclusion

In this deliverable we have described the additions we have made to the Stage 2 ASHiCS scenario, with the aim of creating greater complexity and using an objective measure of that complexity to help direct the search to complex, high risk scenarios. We have introduced and implemented a severe weather event to the search inputs, with further options to increase the search space by allowing it to select the storm's entry times into the air space. We have attempted to address concerns related to the lack of contextual information of our search results that we first highlighted in D2.2 and again at SID 2012. Our additions remain on-going work, and we hope our work for the final stages of the ASHiCS project has the potential to provide safety analysts with some confidence about the nature of search results, and whether such results can be viewed as rare or expected outcomes for a given range of input values to an air sector.

6 References

[1] A. Yousefi, R. Hoffman, M. Lowther, B. Khorrami and H. Hackney, “Trigger Metrics for Dynamic Airspace Configuration,” in *9th AIAA ATIO Conference*, 2009.

[2] ISA Software, *RAMS Plus User Manual Version 5.36,* 2011.

[3] E. Perrin, B. Kirwan and R. Stroup, “A Systemic model of ATM Safety: the integrated risk picture,” in *7th ATM Seminar*, Barcelona, 2007.

[4] K. De Jong and W. Spears, “An analysis of the interacting roles of population size and crossover in genetic algorithms,” in *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 1991, pp. 38-47.

[5] D. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, 1989, pp. Goldberg, D. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

[6] J. R. Koza, M. Keane and M. Streeter, “Evolving inventions.,” *Scientific American,* p. 52–59., 2003.

[7] C. F. Lima, D. E. Goldberg, K. Sastry and F. G. Lobo, “Combining competent crossover and mutation operators: A probabilistic model building approach,” in *Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO '05)*, New York, 2005.

[8] S. Borenor, “Innaxis Complex World - Key note presentation: Complexity and safety performance - Will circular accident causality result from NextGen and SESAR?,” 2011. [Online]. Available: http://complexworld.innaxis.org/presentations/borener.pdf.

[9] D. White and S. Poulding, “A rigorous evaluation of crossover and mutation in genetic programming,” in *12th European Conference, EuroGP 2009*, Tübingen, Germany, 2009.

Appendix A

A.1 Implementation of storm restricted zone

Code below shows the creation of the polygon that represents the storm.

```
void FileHandler::writeRestrictedZonesCorners(int scenarioNumber, bool mutateZones)
//call after write traffic.  Scenario already initialised. Routine creates direction for storm by
//defining a series of restricted zones moving from one corner point in the air sector to another.
//Any two corners can be chosen, provided they are different.  Distance between corners is calculated
//and divided by ten.  This gives increment of movement.  Storm therefore moves fastest between those
//corners that are the furthest apart.  Typically a zone has a 12 min window before it moves on. Gives
//an approximate max. speed of about 50mph.  Restricted zone size approx 15nm across.  We may allow
//evolutionary mutation of start times to provide some selection via variation in the storm's effects.
//Flights resolved around restricted zones may add to workload tasks, but this is a minor weighting
//against measures such as conflicts involving CPLoss or severity of conflict.  We only generate two
//files directly: corner.dat and restriction.dat
{
        try
        {

        //rnd
        Random^ rndNumber = gcnew Random();

        //main air sector boundary corner co-ordinates (lat, long) for Stage 2 en route air sector
        const int lat=0, lng=1;
        double c1[] = {53.1350150796, -1.3354370849};
        double c2[] = {53.0052949047, -2.0005078425};
        double c3[] = {52.4861085701, -2.5130166995};
        double c4[] = {51.8227949275, -2.2401489379};
        double c5[] = {51.7222316928, -0.7592755586};
        double c6[] = {52.1754512422, 0.0231939035};
        double c7[] = {53.0044158306, -0.0575166331};

        //main air sector boundary corner co-ordinates array
        double sectorCorners[][2] = {
                {c1[lat],c1[lng]},
                {c2[lat],c2[lng]},
                {c3[lat],c3[lng]},
                {c4[lat],c4[lng]},
                {c5[lat],c5[lng]},
                {c6[lat],c6[lng]},
                {c7[lat],c7[lng]}
        };

        //badly drawn octagon for restricted zones defined using offsets from single sector corner
        //NB. (lat, lng), mercator projection means you can't define a simple square and rotate it!
        double offs[8][2] = {
                {0.0, 0.0},             //C8 0, 0  first corner (top right) is one of the sector
                                        //corners, go round anti-clockwise
                {0.03, -0.15},          //C9
                {0.0, -0.3},            //C10
                {-0.1, -0.38},          //C11
                {-0.2, -0.3},           //C12
                {-0.25, -0.15},         //C13
                {-0.2, 0.0},            //C14
                {-0.1, 0.08}            //C15
        };

        int noOfRestrictedZones = 10;

        // path and RAMS traffic file names
        String^ cornerFile = String::Concat(FileGlobals::scenarioPath, scenarioNumber,
FileGlobals::corner);


        //open corner.dat file for writing
        StreamWriter^ swcorners = gcnew StreamWriter(cornerFile);

        Int32 startPoint = G::allResults->generation[scenarioNumber]->scenario->RestrictedZonePath[0];
```

```
        Int32 endPoint = G::allResults->generation[scenarioNumber]->scenario->RestrictedZonePath[1];

        //check if this is initial call to set storm trajectory, if so, use random storm paths
        //otherwise, regenerate previous storm path.
        if (mutateZones) {

                //Assign storm track corners
                startPoint = rndNumber->Next(1, 8);
                G::allResults->generation[scenarioNumber]->scenario->RestrictedZonePath[0] =
startPoint;

                //give time for clock reset
                Thread::Sleep(10);
                endPoint = rndNumber->Next(1, 8);

                //ensure start point and end point are different
                do
                {
                        Thread::Sleep(10);
                        endPoint = (endPoint == startPoint) ? rndNumber->Next(1, 8) : endPoint;
                } while (startPoint == endPoint);

                G::allResults->generation[scenarioNumber]->scenario->RestrictedZonePath[1] = endPoint;

        }

        //array length
        int corners = sizeof sectorCorners / sizeof sectorCorners[0];

        ///loop through corners, writing each one to file for main air sector
        for (int i=0; i < corners; i++) {
                String^ cornerCoOrd = String::Concat("  ", Convert::ToString(sectorCorners[i][lat]), "
", Convert::ToString(sectorCorners[i][lng]));
                swcorners->WriteLine(String::Concat("C", i+1, cornerCoOrd));
        }

        //we have are corner start and end points for the restricted zones that will represent the
storm.  First create offset increment.
        //increment subtracted from initial corner co-ords to start each restricted zone.
        double increment[2];
        double firstCorner[2];

        //offset each restricted zone by equal distance between boundary 0 corners divided by
noOfRestrictedZones
        increment[lat] = (sectorCorners[startPoint][lat] - sectorCorners[endPoint][lat]) /
noOfRestrictedZones;
        increment[lng] = (sectorCorners[startPoint][lng] - sectorCorners[endPoint][lng]) /
noOfRestrictedZones;

        //create remaining restricted zones using offsets
        for (int i=0; i < noOfRestrictedZones; i++){

                //first boundary is set from one of boundary 0 corners, so apply no offset
                if (i == 0) {
                        //offset first corner (top right), ref point for all other corners
                        firstCorner[lat] = sectorCorners[startPoint][lat];
                        firstCorner[lng] = sectorCorners[startPoint][lng];
                }else {
                        //offset all other corners by increment
                        firstCorner[lat] -= increment[lat];
                        firstCorner[lng] -= increment[lng];
                }

                //create co-ords for hexagon usings offsets from top right corner
                double restrictedZone[8][2] = {
                        {firstCorner[lat], firstCorner[lng]},
                        {firstCorner[lat] + offs[1][lat], firstCorner[lng] + offs[1][lng]},
                        {firstCorner[lat] + offs[2][lat], firstCorner[lng] + offs[2][lng]},
                        {firstCorner[lat] + offs[3][lat], firstCorner[lng] + offs[3][lng]},
                        {firstCorner[lat] + offs[4][lat], firstCorner[lng] + offs[4][lng]},
                        {firstCorner[lat] + offs[5][lat], firstCorner[lng] + offs[5][lng]},
                        {firstCorner[lat] + offs[6][lat], firstCorner[lng] + offs[6][lng]},
                        {firstCorner[lat] + offs[7][lat], firstCorner[lng] + offs[7][lng]}
```

```
                };


                //loop through corners, writing each one to file
                for (int j=0; j < (sizeof restrictedZone / sizeof restrictedZone[0]); j++) {
                        String^ cornerCoOrd = String::Concat("   ",
Convert::ToString(restrictedZone[j][lat]), "     ", Convert::ToString(restrictedZone[j][lng]));
                        corners++;
                        swcorners->WriteLine(String::Concat("C", corners, cornerCoOrd));
                }

        }


        swcorners->Close();

        }

        catch (Exception^ e)
        {
                Console::WriteLine("Problem in writeRestrictedZonesCorners.  Error: {0}", e);
        }

}
```

Appendix B

A.2 Complexity factors used in RAMS Plus

The following complexity factors are reproduced for the convenience of the reader from the RAMS Plus User Manual, pages 300-304 [2].

A.2.1 Aircraft Count (ACT)

This is a count of the number of aircraft within the lateral and altitude boundaries of the sector at an instant of time.

A.2.2 Aircraft Density (DNS)

Aircraft Density is the aircraft count divided by the usable amount of sector airspace. This complexity factor provides correlation with the flexibility that a controller has with each aircraft in its sector, due to the amount of airspace that is available on a per aircraft basis.

DNS = ACT(i) / Sector Volume, where sector volume is square miles.

A.2.3 Airspace Structure (STR)

This complexity factor measures the conformance of the traffic flow through a sector to the geometry of the sector. In general, sectors are designed to conform the major traffic flow. For example, arrival sectors are generally designed to be longer and narrower than normal sectors, and are oriented toward the arrival terminal area, so the aircraft fly in the same general direction through the length of the sector. A controller's complexity can be increased if there are aircraft flying against the major traffic flow.

The computation of the STR, at an instant in time, requires:

• Get the sector major axis

• Calculate the aspect ratio for the sector. The aspect ratio is the:

Maximum (Length, Width) / Minimum (Length, Width)

• Calculate the difference in heading between each aircraft and the major axis, using radian degrees.

• Squared the difference in heading

• Weight the squared deviation by the aspect ratio (aspect ratio x square deviation) and then summed over all aircraft in the sector.

A.2.4 Climbing or Descending (CoD)

This complexity factor is a count of the number of aircraft that are in climb or descent at an instant in time.

A.2.5 Closest Points Approach (CPA)

This complexity factor is weighting of the number of aircraft that are within a threshold separation of each other at any instant in time. This complexity factor is predicting potential losses of separation and therefore implies a high monitoring between the two aircraft by the controller. This complexity factor is only nonzero at a time, n, at which the aircraft are actually predicted to be within a given threshold, rather than being non-zero if there is a predicted conflict sometime after n.

This complexity factor needs two separations parameters to be defined:

• CPA Distance 1 (default of 8 miles) which is used as an indication of a predicted separation that would cause action on the part of the controller.

• CPA Distance 2 (default of 13 miles) which is used as an indication of a predicted separation that would cause heightened separation monitoring between the two aircraft by the controller.

In the computation of this complexity factor, one unit is added to the CPA factor at any time at which two aircraft are predicted to be within Distance 1 miles of each other. One half unit is added to the CPA factor at any time at which two aircraft are predicted to be more than Distance 1 miles apart, but less than Distance 2 miles.

A.2.6 Aircraft Proximity to Sector Boundary (PRX)

This complexity factor is a count of the aircraft that are within a threshold distance of a sector boundary at a given time instant. When aircraft are near a sector boundary, a greater amount of coordination and monitoring is required, which can increase controller complexity. The PRX parameter defines the threshold distance, in nautical miles.

A.2.7 Variance in Directions of Flight (VDF)

This complexity factor is a measure of the variability of heading of all of the aircraft in the sector at a time instant. A higher heading variability of the traffic provides less organisation of the traffic flow and therefore higher controller monitoring.

This function computes the variability of heading of all of the aircraft in the sector at time instant i.

$VDF(i) = 1/n(n - 1) X SUM (hdg(k) - hdg(l))^2$

Where n = ACT(i) and hdg(k) is the aircraft heading, in degrees. The difference of the aircraft heading is between 1 and 180 degrees.

A.2.8 Convergence Angle (ANG)

This complexity factor is a measurement of the severity of each conflict situations based on the conflict geometry. Considered as a potential complex conflict, conflicts with small convergence angle between aircraft and Head-on conflicts. This function computes the convergence angle (severity) of each conflict situation based on the conflict geometry at an instant in time i. A score of one unit is assigned to a conflict with an intercept angle of zero degrees. As the convergence angle increases to 90 degrees, the component score decreases. The score then again increases with convergence angle back to one full unit for a head-on convergence angle. Therefore, for any given conflict, the convergence angle has a value between 0 and 1.

A.2.9 Conflict Near Sector Boundary (PRC)

This complexity factor is a count of the predicted conflicts that will occur within a threshold distance of a sector boundary. This function computes the number of the predicted conflicts that will occur within a threshold distance of a sector boundary. This function increase the PRC by one unit for each conflict within PRC Distance 1 (default of 10 miles) of the sector boundary, and one half unit for each conflict that is within PRX Distance 2 (default of 20 miles) of the sector boundary. The PRC parameters define the Distance 1 and Distance 2, in nautical miles.

A.2.10 Aircraft Neighbouring Conflict (NBR)

This complexity factor is a count of other aircraft that are close to the area of the potential conflict. This complexity factor is used to model the reduction in flexibility that a controller has in order to resolve a conflict when specific aircraft are within the region of conflict. Consider a situation where two or more aircraft are predicted to be in conflict at an instant in time, this function computes the number of the other aircraft that are within the general area of conflict.

For the computation of this function, the general area of conflict needs to be defined by two parameters:

• NBR Lateral which defines the lateral radius of the general area of conflict.

• NBR Vertical which defines the vertical height of the general area of conflict.

The general area of the conflict is defined as the centre of the bounding box area of the conflict separation zone.

A.2.11 What is the NASA Complexity Algorithm?

The following complexity formula is a result of the NASA study that defined the sector complexity factors. The weight assigned to each complexity factor has been verified by the NASA study as an appropriate method to measure sector complexity. Consider m is the look-ahead time. At an instant n, the final complexity formula is expressed as:

0.0172 x MAX(ACT(n),...,ACT(n+m))

0.328 x MAX(DNS(n),...,DNS(n+m))

0.0498 x SUM(CPA(n),...,CPA(n+m))

0.1070 x SUM(ANG(n),...,ANG(n+m))

0.0426 x SUM(NBR(n),...,NBR(n+m))

0.0754 x SUM(PRX-C(n),...,PRX-C(n+m))

0.1134 x SUM(CoD(n),...,CoD(n+m))

0.0709 x MAX(VDF(n),...,VDF(n+m))

0.0 x MAX(VAS(n),...,VAS(n+m))

0.2 x SUM(PRX(n),...,PRX(n+m))

0.0676 x MAX(STR(n),...,STR(n+m))

0.2564 x MAX(INT(n),...,INT(n+m))

= OVERALL COMPLEXITY

The overall complexity is a weighted sum of contributions from individual complexity factors as described above. Each of the complexity factors contributes to the overall complexity through either a maximum (MAX) value of the complexity factor between the time n and the time n+m or a summation (SUM) of values computed during the look-ahead time interval.

-END OF DOCUMENT-