Proceedings of the 2007 IEEE Symposium on
Artificial Life (CI-ALife 2007)

Biasing XCS with Domain Knowledge for Planning Flight Trajectories in a Moving Sector Free Flight Environment

Kuang-Yuang Chen* Hai Huong Dam** Peter A. Lindsay* Hussein A. Abbass**

*ARC Centre on Complex Systems,
University of Queensland
Australia
{kuang,p.lindsay}@itee.uq.edu.au
Tel: +61 7 3365 1195
Fax: +61 7 3365 1533

**ARC Centre on Complex Systems,
University of New South Wales
Australia
{h.dam,h.abbass}@adfa.edu.au
Tel: +61 2 6268 8158
Fax: +61 2 6268 8581

***Abstract*— Free flight is a new concept in air traffic management, where pilots are given more freedom in making decisions in the cockpit. This allows air traffic controllers to manage more flights. One of the concepts under investigation in the Australian airspace is moving sectors, where an air traffic controller becomes responsible of a moving volume of the space containing a group of airplanes. Planning flight trajectories in this group is a hard problem. In this paper, we show that XCS can be used as a reliable planning tool. We also propose a novel idea for incorporating hard constraints within XCS to increase its reliability.**

I. INTRODUCTION

Current Air Traffic Control (ATC) is a centralized system, where Air Traffic Controllers (ATCo) are responsible for critical decisions such as conflict resolution. An ATCo is normally responsible for a pre-defined fixed volume of the space known as a sector. However, several studies indicated that fixed-sector controllers may become a limiting factor in air traffic growth in the future [14]. This became a major task in current research activities to find better methodologies to improve the efficiency of future air transportation. One school of thought is to fundamentally change the airspace sectorization in order to improve the efficiency of air transportation and Air Traffic Management (ATM). This school includes a number of approaches such as the Sector-Less [5], SuperSector [8] [7] and The Tube Advanced Lane Control (TALC)[6] concepts. However, the detailed implementation of these concepts is not well-established as yet.

Another approach is Free Flight, which has been proposed as an alternative to current policy of allowing pilots very little freedom to choose optimal routes, altitude, speed, etc. In the Free Flight environment, pilots could profit from favorable winds to take faster routes to their destinations or avoid unfavorable weather conditions when provided with sufficient information. The benefits of Free Flight include fuel savings, reductions in flight delays, time-savings to destinations, and increase use of space [1]. There are still several outstanding research issues, such as how to maintain and guarantee safety, how to minimize cost to airlines, and how to manage congestion and predict airspace constraints [14].

The version of Free Flight being proposed for adoption in Australia and elsewhere is called *User Preferred Trajectories (UPT)*. Under a UPT regime, pilots (or more likely, Airline Operations Centres) would choose a 4D trajectory (route, altitude and times at waypoints) that best suits their operational imperatives. For example, if they want to save fuel, a great circle route would have the shortest ground distance, but may not have the shortest air-distance, due to the effect of head or tail winds, weather conditions, and special use air space. For some flights it may be more important to save time than fuel.

Using Australian air traffic as an example, it is clear that UPTs are likely to lead to bursts of activity in fixed sectors, due to many airlines wanting to fly similar routes at similar times. One possible approach proposed by the industry to decompose this problem is called the *Moving Sector* concept, whereby a group of aircraft flying similar trajectories are allocated to a single ground-based ATCo, for part of their journey at least. Specifically, a moving volume of airspace would be the responsibility of the ATCo. The issue addressed in this paper is how to provide a decision support system for off-line planning of trajectories so that airlines can try to optimise individual flight trajectories while at the same time optimising an overall system performance metric, in order to assure airspace safety and efficiency.

This is a hard optimization problem. There are two general ways of thinking on how to solve this problem. One is to build an optimizer in which the problem is represented mathematically or otherwise, and every time a new problem arises, the optimizer is run to find the new set of solutions. Another approach is to evolve a set of rules that can tell us how to find solutions. In other words, evolution is not used to optimize, but is used to find rules on how to optimize. This is the approach followed in this paper since it generates a set of rules that can be interrogated by a human expert for risk assessment. They can also be used to educate pilots on how to choose an option when faced with a conflict.

In this paper, we use XCS - an evolutionary classifier system - to evolve the set of rules that can be used for path planning in a moving sector problem. We investigate

1-4244-0701-X/07/$20.00

the effect of the number of populations and population size on the nature of evolved rules. This paper addresses two research questions; these are:

- How to handle long action chains? In the off-line planning system, the performance of the trajectories for all aircraft will be evaluated after all aircraft arrive at the destination. Learning Classifier Systems are not able to solve larger "Long Action Chain" (or Multistep) problems robustly [2] [4]. We believe that we can solve this problem efficiently if a carefully designed local reward function is designed. This paper presents one such local reward function.
- How to incorporate Hard Constraints? The path planning problem has hard constraints, such as ensuring the aircraft trajectories stay within the moving volume and stay safely separated in time and space. These constraints can be incorporated in the reward function by penalizing transitions that may break hard constraints. However, we hypothesize that it is more efficient to incorporate these hard constraints within XCS. Although the system is proposed to be an off-line system, time and computing resources are still very important. We propose in this paper a mechanism to initialize XCS with hard constraints and compare between XCS with and without these constraints.

The next section will introduce a short literature review followed by the approach. We then present the experiments, analyze the results, and conclude the work.

II. The Approach

A. The Search Technique

Learning Classifier Systems (LCS) [9] are rule-based systems, where the rules are usually in the traditional production system form of "IF condition THEN action". The rules provide a very general solution to a given problem. An evolutionary algorithm and heuristics are used to search the space of possible rules, whilst a credit assignment algorithm is used to assign utility to existing rules, thereby guiding the search towards better rules. The LCS formalism was introduced by Holland [9] and based around his more well-known invention - the Genetic Algorithm (GA). A few years later, in collaboration with Reitman [12], he presented the first implementation of an LCS. Holland then revised the framework to define what would become the standard system [10] [11] [3]. A very useful benefit to the use of LCS is the accessibility of the information that they obtain (or learn) through interaction with the environment. The explicit nature of the classifier's structure permits the possibility of subsequent higher-level interpretation and organization. In the research of Sen and Sekaran [15], action policies were developed to optimize environment feedback. According to their experimental results, classifier systems can be more effective than the more widely used Q-learning scheme for multi-agent coordination on a resource sharing problem and a robot navigation problem.

The "eXtended Classifier System" (XCS) was introduced by Wilson [16]. The difference between XCS and traditional LCS is that in traditional LCS, the rule fitness is based on the payoff received by rules but in XCS the fitness is based on the accuracy of predictions in payoff. Advantages of XCS include an ability to form accurate maximal generalizations and improved performance [17]. An example of the accurate, maximally general classifier corresponding to the inputs of 000000, 000001, 000010, 000011, 000100, 000101, 000110, 000111 is 000###. XCS contains rules (called classifiers), some of which will match the current input. An action is chosen based on the predicted payoffs of the matching rules.

The rule-based system consists of a population of condition-action rules or "classifiers". The structure is [17]:

$$< condition >:< action >\Rightarrow< prediction >$$

For example: 01#1## : 1 $\Rightarrow$ 943.2. The # acts as a wildcard allowing generalization so that the rule condition #011 matches both the input 0011 and the input 1011. For each action in $[M]$, classifier predictions are weighted by the fitness F to get system's net prediction in the prediction array. The estimates (keeps an average of) is the predictions of the payoff expected if the classifier matches and its action is taken by the system. Based on the system predictions, an action is chosen and sent to the environment. XCS is chosen for this research due to the following reasons:

- It provides a method to encode how to find solutions, instead of directly optimizing a solution; thus one can understand the process of finding solutions rather than merely finding them.
- It has enhanced readability [13]: XCS presents a symbolic representation that is easy to understand, so it should be helpful for us to analyze how the system finds solutions and to extract behavior rules for ATM from the rule set in the population.

B. The Environment

This paper uses a simplified version of the Moving Sector problem in order to develop the methodology. First we decompose the problem into multiple moving volumes of airspace: see Fig. 1. Each volume is structured into "lanes" which are arranged horizontally and vertically and separated by a minimum safe distance (5 nautical miles is the usual standard for radar-controlled airspace): see Fig. 2. The volume moves in a straight line at constant altitude and at a constant speed. In this highly simplified example, aircraft are also assumed to move at the same and constant speed, slightly greater than that of the volume, so that from a relative perspective they are moving steadily along the lanes. Finally, for simplicity, time will be discretised, and lane changes will be made only at discrete, evenly separated points in time; changing lanes will not incur a speed penalty. As a result of all these simplifications, the problem has been reduced to aircraft moving through a 3D grid of cells. A *scenario* will be specified by defining the lane in which each aircraft begins (its *entry waypoint*) and the desired lane in which it should finish when it reaches the other end of the volume

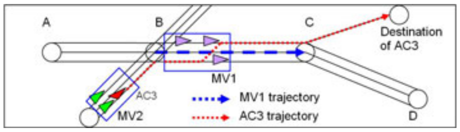

Fig. 1. Two Groups of Aircraft in Two Moving Volumes

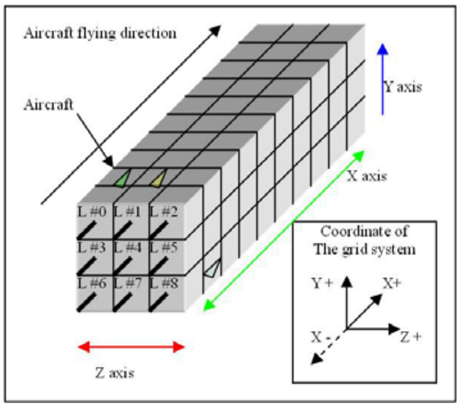

Fig. 2. The Structure of a Moving Volume

(its *exit waypoint*), for each aircraft in the Moving Volume (MV).

A trajectory in this case is thus a path through the cells. At each time step an aircraft can move into the cell in front of it, or to one of the cells adjacent to that (which corresponds to changing lanes): see Fig. 4. The challenge in planning trajectories has several parts: do not enter a cell that already contains another aircraft; do not cross the side boundaries of the volume; try to end up in the lane containing the exit waypoint: see Fig. 3. The optimisation problem is to try to minimise the overall number of lane changes.

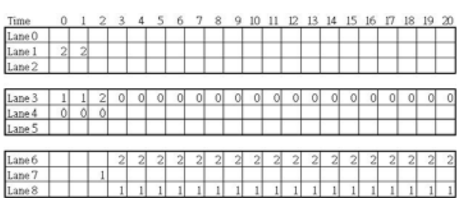

Fig. 3. Example of 3 Aircraft Trajectories in The Moving Volume

III. Rule Representation in XCS

In our experiments, it is assumed that there are nine lanes, namely Absolute Lanes #0 to #8, and the state the agent can sense is the state of the Adjacent Lanes, namely Adjacent Lanes #0 to #8, at the next time step. The Adjacent Lanes are defined relative to the lane the aircraft currently flies on, which will always be referenced as Adjacent Lane #4 (i.e. each aircraft is centered in the square). Thus, the Adjacent Lanes will be numbered as #0 to #8, as shown in Figure 4.

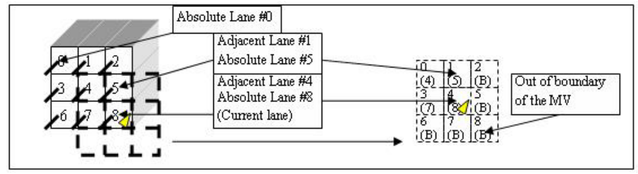

Fig. 4. The “Absolute Lanes” and “Adjacent Lanes”

Although, in current experiments, the speed of the aircraft is not changeable, one aircraft may sense other aircraft, due to the sequence (we assume a non-cooperative resolution policy) in which aircraft plan their trajectories. For example, if the sequence is AC2–AC0–AC1, the aircraft which AC1 may sense will be AC2 and AC0. The rules in XCS are used to represent the states of the environment. One rule may be used, only when the condition part can match the current state of the environment. So, the nodes of the condition and action part of the system may be:

TABLE I

The condition part of the XCS system

Node	Description	Option
0 to 8	The status of adjacent lanes at the next time step, Adjacent Lane #0 to #8 respectively.	0, 1, 2, #
9 to 17	The Adjacent Lane will be closer to or further away from or the same distance to the distance. Node 9-17 represents Adjacent Lane #0 to #8 respectively.	0, 1, 2, #

TABLE II

The action part of the XCS system

Node	Description	Option
0	Which Adjacent Lane will be chosen as the next step? The number, 0-8, represents Adjacent Lane number of the current position.	0, 1, 2, 3, 4, 5, 6, 7, 8

For nodes #0 to #8, the option of 0 means the adjacent lane at the next time step will be available, 1 means the lane will not be available due to another aircraft, and 2 means it is outside of the boundary of the volume. For node 9 to 17, the option of 0 means that by choosing the lane for the next time step, the aircraft will be closer to the destination, 1 means that the distance between the aircraft and the destination will remain the same and 2 means that the aircraft will be further away from the destination. In our experiments, the destination means the lane containing the exit waypoint. The ‘#’ is a “wildcard”, which means the node with ‘#’ will be ignored when the XCS system tries to match rules with the inputs of the environment. For the node of the action part, the number, between 0 and 8, of the option means the Adjacent Lane number the aircraft will choose for the next time step as its action. In the example of Figure 5, the best action should be Adjacent Lane #6.

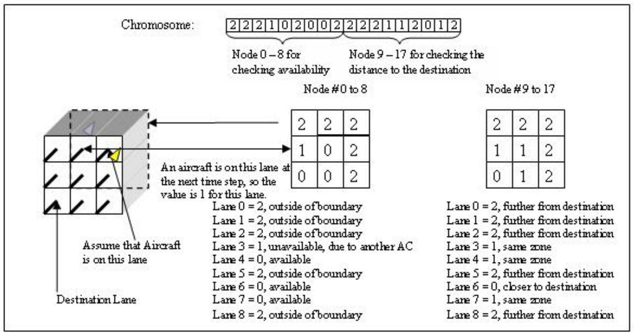

Fig. 5. The Chromosome of a Rule

A. Encoding hard constraints in XCS

The rules in the population are generated with the Covering function of XCS. However, the system may need a lot of training time and computing resources to learn a good set of rules. In this application, there are certain rules that we would like to enforce all of the time, namely: ensuring that aircraft do not collide and do not cross the volume side boundaries. We call such rule *hard constraints*. One approach is to add such rules to the system when it is initialized and tag them so that they do not get deleted. Also, these rules are given the highest possible fitness so that they always get selected from the match set, and the lowest possible prediction accuracy, so that they never get selected in action set. It is expected that with these rules, the system will learn what action should not be taken, so as to save the time on trying wrong actions. Figure 6 shows three of these rules.

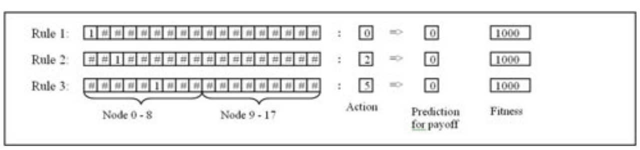

Fig. 6. Partial List of Initial Rules for Constraint

B. The Reward Function

Some researchers tried to solve Long Action Chain problems for XCS, but the results show that further investigations are still needed [2]. Our proposal is that in some problems, it might be possible to design local reward functions to guide the search algorithm to construct a long action chain. We define the search problem as the determination of an algorithm to find a path that is: A) continuous from origin to destination, B) has a minimal number of lane changes, C) has no collision with another aircraft. Therefore, a simple possible reward function is presented in Algorithm 1. This paper investigates the suitability of this reward function.

Algorithm 1 Reward Function

$AL \leftarrow$ the_Adjacent_Lane_chosen_for_the_next_time_step
$CL \leftarrow$ the_Current_Lane
if $AL =$ Not_Available **then**
  rewards = 0;
else if $AL =$ Outside_Boundary **then**
  rewards = 0;
else
  if $Dist\_to\_Destin(AL) < Dist\_to\_Destin(CL)$ **then**
    rewards = 1000;
  else if $Dist\_to\_Destin(AL) = Dist\_to\_Destin(CL)$ **then**
    if $AL = CL$ **then**
      rewards = 500;
    else
      rewards = 300;
    end if
  else
    rewards = 100;
  end if
end if

TABLE III
SCENARIOS USED IN EXPERIMENTS

Scenario	AC0		AC1		AC2	
	init lane	final lane	init lane	final lane	init lane	final lane
S0	#2	#7	#7	#0	#8	#1
S1	#1	#6	#5	#2	#6	#3
S2	#4	#3	#3	#8	#1	#6
S3	#5	#0	#6	#2	#8	#1

IV. Experimental Setup

We undertook two experiments to test the feasibility of our approach. There are common assumptions in these experiments: (A) All aircraft are trained to fly in the same direction and speed from origins to destinations. (B) All aircraft plan their trajectories in sequence. The sequence reflects the priority level assigned to each aircraft, where aircraft would normally have to negotiate their priority in a MV. The XCS parameters are the same for all experiments. 200 training scenarios generated by the system randomly are used to train the aircraft. Each training scenario will be used to train the aircraft for 5 epochs. The difference between these scenarios is the combination of the origins and destinations for all aircraft. Four testing scenarios (Table III) are designed manually for testing. Each experiment was repeated with 30 different seeds and the same 30 seeds are used in all setups. The evaluation criteria are: (1) safety performance measures: the number of crashes (i.e., two aircraft occupying the same cell), and the number of times aircraft crossed the boundary; and (2) efficiency performance measures: the number of lane changes for the whole team, the number of climb/descend, and the number of heading changes. A run is said to be *successful* if there are no crashes, no aircraft cross the boundaries, and all aircraft arrive at their exit waypoints.

The first set of experiments is designed to evaluate the utility of incorporating hard constraints. 18 initial rules are added to the population when the system is initialized. The results are compared with and without these hard constraints.

The second set of experiments is designed to compare between the use of a single population shared among all aircraft and the use of a different population for each aircraft.

V. Results

In this section, we will present different statistics on the experiments. Table IV presents the number of successful runs when using a single population and with/without hard constraints. It clearly demonstrates the usefulness of biasing the initial XCS population by incorporating hard constraints in the form of rules with high fitness and 0 prediction. The number of successful runs varied from one test scenario to another, but it is clear that the four different scenarios have different characteristics, which is desirable when using them for testing the generalization of XCS.

TABLE IV
Total number of successful runs when using a single population with and without hard constraints
(Out of 30 tests)

Scenario	Pop size	Without Constraints	With Constraints
S0	1000	0	0
	3000	2	14
	9000	14	25
	12000	14	25
S1	1000	0	0
	3000	9	21
	9000	26	28
	12000	26	28
S2	1000	0	0
	3000	4	17
	9000	15	24
	12000	15	24
S3	1000	0	0
	3000	4	16
	9000	20	29
	12000	20	29

TABLE V
Total number of successful runs when using three populations with and without hard constraints
(Out of 30 tests)

Scenario	Pop size	Without Constraints	With Constraints
S0	1000	0	0
	3000	4	15
	4000	5	19
	5000	4	17
S1	1000	1	4
	3000	7	25
	4000	18	27
	5000	21	29
S2	1000	0	0
	3000	4	13
	4000	4	20
	5000	4	24
S3	1000	0	2
	3000	2	13
	4000	7	21
	5000	3	25

TABLE VI
Average aircraft performance measures without constraint initialization and when using a single population

Scenario	Pop size	climb/ descend	heading changes	both	lane changes
S0	1000	20.00	20.00	20.00	20.00
	3000	18.76	18.69	18.71	18.82
	9000	11.14	10.68	11.12	11.61
	12000	11.14	10.68	11.12	11.61
S1	1000	20.00	20.00	20.00	20.00
	3000	14.38	14.14	14.29	14.81
	9000	3.52	2.68	2.97	3.83
	12000	3.52	2.68	2.97	3.83
S2	1000	20.00	20.00	20.00	20.00
	3000	17.40	17.42	17.64	17.80
	9000	10.13	10.37	10.37	10.87
	12000	10.13	10.37	10.37	10.87
S3	1000	20.00	20.00	20.00	20.00
	3000	17.60	17.40	17.67	18.00
	9000	6.91	6.91	7.71	8.20
	12000	6.91	6.91	7.71	8.20

TABLE VII
Average aircraft performance measures with constraint initialization and when using a single population

Scenario	Pop size	climb/ descend	heading changes	both	lane changes
S0	1000	20.00	20.00	20.00	20.00
	3000	11.14	10.91	11.31	12.01
	9000	4.18	3.34	4.16	5.01
	12000	4.18	3.34	4.16	5.01
S1	1000	20.00	20.00	20.00	20.00
	3000	6.77	6.06	6.62	7.43
	9000	2.26	1.34	1.66	2.59
	12000	2.26	1.34	1.66	2.59
S2	1000	20.00	20.00	20.00	20.00
	3000	8.88	9.09	9.27	9.90
	9000	4.22	4.58	4.58	5.38
	12000	4.22	4.58	4.58	5.38
S3	1000	20.00	20.00	20.00	20.00
	3000	9.82	9.83	10.53	11.51
	9000	1.01	1.01	1.93	2.62
	12000	1.01	1.01	1.93	2.62

Table V shows a similar comparison when each aircraft is assigned a different population. Here, the advantages of biasing the initial population with hard constraints are even more evident in this example. The number of successful runs with hard constraints is an order of magnitude better in most experiments. Also, refer to Table IV, by comparing a single population of size $3n$ and 3 populations with size n each, one can see real advantages when using a single population as compared to a separate population for each aircraft. The advantages are better success rate and smaller populations.

Tables VI and VII compare the performance of the different XCS populations in terms of the efficiency performance measures for all aircraft sharing one population. Where no solution was returned a value of 20 is used, which is equal to the worst performance (i.e. each aircraft changes lanes at every step). The definition hold true for Tables VIII and IX, where a separate population is used for each aircraft. The results for the cases with constraints (Table VII) are clearly better than without constraints (Table VI) in most cases (with statistical significance no more than 0.0143 and alpha=0.05).

TABLE VIII

AVERAGE AIRCRAFT PERFORMANCE MEASURES WITHOUT CONSTRAINT INITIALIZATION AND WHEN USING THREE POPULATIONS

Scenario	Pop size	climb/ descend	heading changes	both	lane changes
S0	1000	20.00	20.00	20.00	20.00
	3000	17.46	17.36	17.50	17.63
	4000	16.82	16.68	17.02	17.19
	5000	17.48	17.34	17.46	17.61
S1	1000	19.56	19.24	19.36	19.59
	3000	15.76	15.37	15.46	15.87
	4000	9.08	8.30	8.70	10.02
	5000	6.89	6.01	6.50	7.37
S2	1000	20.00	20.00	20.00	20.00
	3000	17.37	17.46	17.61	17.77
	4000	17.39	17.48	17.41	17.61
	5000	17.38	17.62	17.42	17.76
S3	1000	20.00	20.00	20.00	20.00
	3000	18.69	18.69	18.76	18.80
	4000	15.46	15.48	15.75	16.00
	5000	18.03	18.03	18.13	18.20

TABLE IX

AVERAGE AIRCRAFT PERFORMANCE MEASURES WITH CONSTRAINT INITIALIZATION AND WHEN USING THREE POPULATIONS

Scenario	Pop size	climb/ descend	heading changes	both	lane changes
S0	1000	20.00	20.00	20.00	20.00
	3000	10.49	10.08	10.72	11.27
	4000	7.98	7.37	7.96	8.63
	5000	9.23	8.67	9.24	9.80
S1	1000	17.47	17.39	17.44	17.61
	3000	4.44	3.87	3.91	5.50
	4000	2.91	2.04	2.33	3.27
	5000	1.61	0.69	1.22	2.17
S2	1000	20.00	20.00	20.00	20.00
	3000	11.52	11.80	11.88	12.51
	4000	6.87	7.62	7.51	8.64
	5000	4.20	4.60	4.61	5.40
S3	1000	19.04	18.78	18.74	19.21
	3000	11.50	11.50	11.90	12.21
	4000	6.26	6.29	7.10	7.63
	5000	3.64	3.64	4.42	5.03

Table X presents that the differences of values of lane changes between Tables VI and VII are significant, when tested with a hypothesis test, ttest2, in Matlab (trademark of The MathWorks, Inc). The exceptions are Scenario S1 with populations 9000 and 12000. Note that S1 is a somewhat simpler scenario than the others, since it requires only four lane changes and the aircraft shortest paths involve only one possible conflict. The XCS without constraints seems to have learnt how to avoid this conflict fairly successfully.

Tables XI and XII compare the number of macro-classifiers in each case. As would be expected, the number of macro-classifiers when the initial population is biased with hard constraints is less than the corresponding number when hard constraints are not used.

VI. CONCLUSION

This paper presented a first attempt at using XCS for path planning in a free-flight air-traffic control environment. We have shown a novel mechanism for incorporating hard constraints within XCS. Although the case study made some unrealistic simplifying assumptions, the results are promising and warrant further research. For future work, we plan to investigate better ways to encode hard constraints so that XCS does not produce any crashes. We also plan to scrutinize the macro populations to reduce their sizes and analyze the semantics of the evolved rules.

TABLE X

HYPOTHESIS TESTING FOR THE DIFFERENCE OF LANE CHANGE VALUES IN TABLES VI AND VII

Scenario	Pop size	h	P-Value
S0	3000	1	0.00042038
	9000	1	0.0024
	12000	1	0.0024
S1	3000	1	0.0012
	9000	0	0.3977
	12000	0	0.3977
S2	3000	1	0.0001628
	9000	1	0.00143
	12000	1	0.00143
S3	3000	1	0.00076708
	9000	1	0.0015
	12000	1	0.0015

TABLE XI

THE AVERAGE NUMBER OF MACRO-CLASSIFIERS WITHOUT/WITH CONSTRAINT INITIALIZATION AND WHEN USING ONE POPULATION.

Population size	Number of Macro classifiers without constraints	Number of Macro classifiers with constraints
1000	814.67	795.8
3000	2458.43	2532.53
9000	8580.07	5028.67
12000	8580.07	5028.67

TABLE XII

THE AVERAGE NUMBER OF MACRO-CLASSIFIERS WITHOUT/WITH CONSTRAINT INITIALIZATION AND WHEN USING THREE POPULATIONS.

Population size	Number of Macro classifiers without constraints	Number of Macro classifiers with constraints
3 x 1000	2604.70	2286.60
3 x 3000	6101.70	5522.03
3 x 4000	8024.30	6487.37
3 x 5000	8940.20	7565.77

Acknowledgment Work reported in this paper was funded by the ARC Centre on Complex Systems (ARC CEO0348249). The authors wish to thank Dr. David Fogel for coordinating the reviews of this paper.

REFERENCES

[1] Australian Strategic Air Traffic Management Group. *Air Traffic Management: A Strategic Plan for Australia.* ASTRA, 2003.

[2] Alwyn M. Barry. The stability of long action chains in XCS. *Journal of Soft Computing*, 6(3-4):pp. 183–199, 2002.

[3] Larry Bull and Tim Kovacs. Foundations of Learning Classifier Systems: An introduction. In Larry Bull and Tim Kovacs, editors, *Foundations of Learning Classifier Systems*. Springer, 2005.

[4] Martin V. Butz, David E. Goldberg, and Pier Luca Lanzi. Gradient descent methods in Learning Classifier Systems: Improving XCS performance in multistep problems. *IEEE Transaction on Evolutionary Computation*, 9(5):pp. 452–473, 2005.

[5] Vu Duong, Gilles Gawinowski, Jean-Pierre Nicolaon, and Darren Smith. Sector-less air traffic management. In *Proceedings of the fourth USA/EUROPE Air Traffic Management R&D Seminar*, 2001.

[6] Jean-Pierre Florent and Marc Brochard. Tube advanced lane control. Technical report, EUROCONTROL Experimental Centre, 2004.

[7] G. Gawinowski and V. Duong. Bridging the predictive and adaptive issues in air traffic management: the synchronous paradigm. In *Digital Avionics Systems Conference, 2003. DASC '03. The 22nd*, volume 1, pages 5.E.4–1–5.E.4–12, 2003.

[8] G. Gawinowski, J. Nobel, J.Y Grau, D. Dohy, L. Guichard, JP Nicolaon, and V. Duong. Operational concepts for Supersector. In *Proceedings of the fifth USA/Europe Air Traffic Management R&D Seminar*, 2003.

[9] J.H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[10] J.H. Holland. Adaptive algorithms for discovering and using general patterns in growing knowledge bases. *International Journal of Policy Analysis and Information Systems*, 4(3):pp. 245–268, 1980.

[11] J.H. Holland. Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In *Machine learning, and artificial intelligence approach.* Morgan Kaufmann, 1986.

[12] J.H. Holland and J.H. Reitman. Cognitive systems based in adaptive algorithms. In Waterman Hayes-Roth, editor, *Pattern-directed Inference Systems.* Academic Press., 1978.

[13] Faten Kharbat, Mohammed Odeh, and Larry Bull. New approach for extracting knowledge from XCS learning classifier system. Technical report, University of the West of England, 2006.

[14] Jimmy Krozel. Free flight research issues and literature search. Technical Report NAS2-98005, NASA Ames Research Center, 2000.

[15] Sandip Sen and Mahendra Sekaran. Multiagent coordination with Learning Classifier Systems. In *Proceedings of the IJCAI Workshop on Adaption and Learning in Multi-Agent Systems*, pages 218 – 233. Springer Verlag, 1996.

[16] Stewart W. Wilson. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149 – 175, 1995.

[17] Stewart W. Wilson. State of XCS classifier system research. Technical report, Prediction Dynamics, March 18 1999.