The Effects of Airline Adaptive Route Selection on NAS-wide Performance

Guillermo Calderón-Meza, PhD. Candidate[1] and Lance Sherry, PhD.[2]
George Mason University, Fairfax, VA, 22030

NAS-wide simulations are used to evaluate concepts of operations, technologies, and their cost/benefits. When modeling stakeholder behavior, users of state-of-the-art NAS-wide simulators must encode decision-making in fixed sets of rules before executing the simulations. Adaptability and evolution in stakeholder decision-making is a next step to accommodate progressively realistic modeling of the NAS. This paper introduces adaptable behavior for airlines into a NAS-wide simulation tool. Adaptability is achieved by the application of Reinforcement Learning and the concept of domination to compare performance of multi-valued functions. The paper studies the effects on the NAS of adaptable route selection behavior of airlines in the presence of local and system-wide historic performance information. The results show the effectiveness of adaptive route selection in improving NAS-wide performance (even) when all agents seek use of the same resources simultaneously. It is also effective when the airlines have only information about their own past and current performance and do not have access to system-wide information. This effectiveness results in benefits for multiple stakeholders. The implications of these results are discussed.

I. Introduction

NAS-wide simulations are used to evaluate concepts of operations, technologies, and their cost/benefits. When modeling stakeholder behavior, users of state-of-the-art NAS-wide simulators must encode decision-making in fixed sets of rules before executing the simulations. Adaptability in stakeholder behavior is a next step to accommodate progressively more realistic modeling of NAS.

This paper introduces adaptable behavior for airlines into the NAS-wide simulation tool *Future ATC Concepts Evaluation Tool* (FACET) [1]. Adaptability in the decision making for pre-departure route selection is achieved by the application of Reinforcement Learning and the concept of domination to compare performance of multivalued functions (Figure 1). This paper describes the results of an experiment to evaluate the impact of accessibility to system-wide information. In treatment 1, the airlines route selection decision-making has access to each other's past performance. In treatment 2, the airlines have access to only their own past performance.

The results show: (i) the degree of stochastic behavior introduced into the NAS by pre-departure route selection alone, (ii) both treatments result in a steady-state equilibrium, (iii) the route structure of some airlines results in less variability and greater benefits, (iv) system-wide access to information results in greater benefits to all stakeholders (including airlines, Air Traffic Control, and passengers) than access to only local airline data.

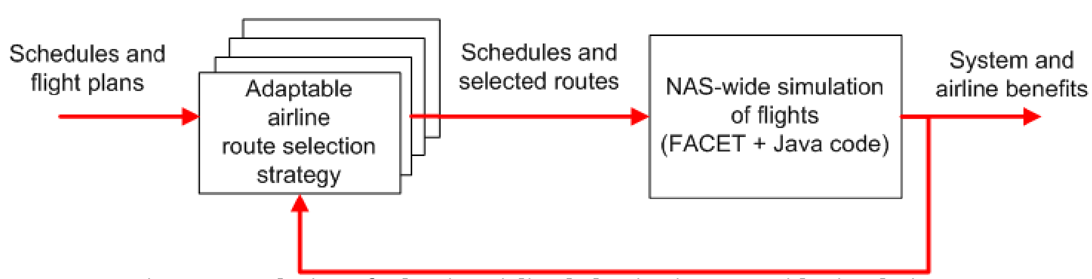

Figure 1: Inclusion of adaptive airline behavior in NAS-wide simulations

[1] PhD. candidate, Center for Air Transportation Systems Research, Fairfax, VA.
[2] Director, Center for Air Transportation Systems Research, Fairfax, VA.

II. NAS-wide simulation tools

Through time, the research community has proposed, designed, and implemented several NAS-wide simulation tools. The following paragraphs briefly describe some of those tools and their potential to include adaptive behavior for the stakeholders. References to all these tools are available from the publications section of the website of the Center for Air Transportation System Research at George Mason University (http://catsr.ite.gmu.edu).

The *National Airspace System Performance Analysis Capability* (NASPAC) by FAA, is a *fast-time*[3] discrete event NAS-wide simulation. The system consists of a set of tools to pre-process data, simulate, and post-process simulation outputs. The simulation core is based on *Queuing Theory*. It has been used for about two decades to model the NAS and obtain estimates of delays and traffic flows at a NAS-wide level. With the current version it is not easy (or even possible) to interact with each flight during the simulation. This “closed” operation mode makes the inclusion of adaptive behavior into the system hard to implement.

The *LMINET* by LMI, is a queuing network NAS-wide model. TRACON and en-route sectors are represented by queues, these queues are linked together to model the NAS.

The *Systemwide Modeler* by CAASD and FAA, is a discrete event fast-time NAS-wide simulator. The simulator includes information for NAS resources and for constraints in their use. The flight plans are executed and modified according to the constraints.

Future ATC Concepts Evaluation Tool (FACET) by NASA, as explained in Bilimoria et al. [1], is a compact, stand-alone, free, NAS-wide simulation tool. The tool is easy to use and its footprint in the computing system is small. It provides a Java *Application Program Interface* (API) that makes the control of the simulation possible from external Java code. The tool includes databases for most of the current administrative and navigation rules of current use in the NAS, i.e., sector and center attributes, navigation aids, airports, flight plans, capacities, conflict detection. The tool is physics-based and neutral in terms of procedures, i.e., no control algorithm is implemented in the model. The tool allows the individual control of many parameters of the NAS and of each individual flight. Since there is access and control to each individual flight, this tool also offers the potential to combine it with a multi-agent simulation framework and include adaptive behavior to the flights. The system does not include any behavior except that flights follow the physical rules at all times.

The *Probabilistic NAS Platform* (PNP), based on ProbTFM [2], is in development at Sensis Corporation, is a modular NAS-wide simulator. At the center of the system is a server with a model of NAS. This model is probabilistic in the sense that it considers the uncertainties inherent to the NAS, i.e., weather, delays, capacity forecasts, speed errors. Custom-made clients can be implemented and connected to the server to use its modeling capabilities. These connectable clients offer the potential of incorporating adaptive behaviors in to the NAS-wide simulations.

The *Airspace Concept Evaluation System* (ACES) by NASA, as explained by Sweet et al. [3], is a flexible, distributed, multi-layered NAS-wide simulation framework. Its design is intended to support a wide range of different models for the NAS to cover any aspect of the system. Its layers include infrastructure for control, coordination, communications, and data analysis. It is also multi-agent which gives the potential to include different behaviors and to observe their interactions.

The *Collaborative Human-in-the-Loop Laboratory* (Chill/Sim-C) by ISA Software, is a system-wide agent-based modeling platform. It can operate based on a model or with human-in-the-loop for research that include humans (e.g., gaming, human factors). This tool includes the System Wide Information Management (SWIM) functionalities. The agents in this tool can be macroscopic or microscopic, which allows the implementation of different levels of fidelity and granularity in the simulation. The possibility of defining agents allows the implementation of adaptable behaviors.

Reorganized ATC Mathematical Simulator (RAMS) by ISA Software (http://www.ramsplus.com/), is a gate-to-gate fast-time ATC/ATM simulation package. Since RAMS is turn-key solution to air transportation modeling it does not allow for the user to expand functionality by including external models that can affect the simulation.

III. Game Theory basics

Airlines make decisions on a daily basis in the competitive market of the air transportation. The current decision-making process can be regarded as game in which the players have only partial information of their environment and complete information about their own performance. The goal of airlines is to maximize their utility, so they are rational agents [4]. Game Theory is a suitable tool to analyze situations with several players (agents, stakeholders)

[3] A fast-time simulation is a simulation in which the ratio of the simulation time to the time of the real process is less than one.

making decisions simultaneously. Russell and Norvig [4] provide a substantial summary of the Game Theory as applied to decision-making. The concepts described there will be summarized in the following paragraphs, since they are the basis for analyzing the results of the simulations in this paper.

Three components define a game in Game Theory:

- *Player*: the entities making decision. Players are also called *agents*.
- *Action*: what a player can choose to do in a particular situation.
- *Payoff matrix*: with the utility to each player for each combination of actions for all players.

The result of the game is its *outcome*. An outcome is *Pareto optimal* if there is no action that can improve all players' utilities without making at least one utility worse. And it is *Pareto dominated* by another outcome if all players would prefer the other outcome.

The behavior of an agent is called a *strategy*, i.e., a mapping between situations (states) and actions. The strategy is also called *policy*. If an agent always takes the same decision when it is in a particular state, it follows a *pure strategy*. If probabilities are involved in the processes of choosing an action, the agent follows a *mixed strategy*. If an agent is rational, it adopts a *rational strategy*, i.e., it will maximize its utility regardless of the performance of other agents in the game. The set of strategies used by all the players of a game is a *strategy profile*.

A strategy can *strongly dominate* in a game if its corresponding outcomes are better than the outcomes of any other strategy adopted by other players in the same game. A strategy *weakly dominates* in a game if it is better that at least one strategy and no worse than any other. A *dominant strategy* dominates all others. Rational agents adopt dominant strategies. If the strategy profile consists of dominant strategies only, the profile is called *dominant strategy equilibrium*. In an equilibrium no player will benefit from individually changing its strategy, i.e., it is an attractive *local optimum* in the space of strategies. According to John Nash every game has at least one equilibrium (i.e., a Nash equilibrium), but not every game has a dominant strategy equilibrium (or even a dominant strategy). In general, there could be multiple equilibrium points for a game, and their outcomes could be Pareto optimal or not. A *solution* of a game is strategy profile that results in a Pareto-optimal Nash equilibrium if it exists. When there are multiple equilibria, the agent must either "guess"' or communicate with other agents to choose the strategies (i.e. the game of this type is called *coordinated game*)[4].

The theory complicates when the games are repeated, i.e., when players act several times in the same game. In this paper, the decisions of airlines are the choices of routes. These decisions are examples of simultaneous single move games. Therefore, the cases for repeated games are not considered here.

IV. Reinforcement Learning

Reinforcement Learning (RL) is an AI[4] technique that enables agents to learn, without supervision, how to behave successfully in an environment [4]. In this context, *unsupervised learning* means learning without a teacher to provide examples of good and bad actions. The idea in RL is to use *feedback* from the environment to guide the knowledge acquisition process. The feedback is in the form of *rewards* or *reinforcement*. Usually, agents start the learning process with no prior knowledge of the environment or the rules of the process it is learning.

There are several types of RL depending on which information is available to the designer and what type of agent is used. The type of RL called *Q-learning* is suitable for the simulation in this research since it does not need a model of the environment (i.e., Q-learning is a *model-free* method). Q-learning measures the performance of each action and learns the *values* of the actions. Later it compares the values of the actions and chooses actions based on those values. The drawback is that Q-learning agents cannot predict the outcomes because they do not have a model of the environment. This is a limitation to the agent's ability to learn [4].

The Q-learning agents learn the values of actions by progressively building a *Q-function*, which is an action-value function. Q-learning is an example of *active learning* for agents must learn what to do (i.e., the policy is not fixed, but changes through time). This type of learning requires *exploration* of the action-value space to acquire knowledge. This is especially true in the beginning of the learning process. The *exploitation* of the knowledge can come only after there is some cumulated experience (i.e., the Q-function becomes more a "total function").

The Q-learning process is simplified for situations in which the action is not a part of a sequence of actions, but it is the only action to choose. In this research airlines choose an action (i.e., a route between two airports) at a particular point in time. This decision is not linked to the past or future route choices of the airline, i.e., route selection is modeled as a single time game. In this simplified situation the *update equation* for the Q-function is as shown in Eq. (1). Where s is the state (i.e., time, origin and destination airport) in which the decision is made, a is the action selected, R is the reward obtained at state s due to the action a, and λ is a *learning rate* parameter that

[4] AI: Artificial Intelligence.

determines the importance of the historic value of Q compared to the reward, R, obtained. When $\lambda = 1$ the agent "forgets" its history and learns only the immediate reward. When $\lambda = 0$ the agent does not learn, but only remembers its past. This parameter must set to balance the trade-off between history and present and make useful learning agents.

$$Q(s,a) \leftarrow (1-\lambda)Q(s,a) + \lambda R \quad (1)$$

The actual selection of the route uses the relation between *utility* and the Q-functions considering the *exploration function* [4] as shown in Eq (2). This variation of Q-learning is called ***ε-greedy Q-learning*** [5]. The function rand() returns a random number from a uniform distribution. The ε parameter controls the level of exploration and exploitation of the algorithm. If ε = 0 the algorithm only exploits the knowledge. If ε = 1 the algorithm only explores the action-state space by choosing actions randomly among the alternates.

$$a = \begin{cases} maxarg_a\, Q(s,a) & if rand(\,) < (1-\varepsilon) \\ randomly\ pick\ an\ a\ in\ Q(s,a) & if\ rand(\,) > \varepsilon \end{cases} \quad (2)$$

The computation of the rewards depends on the problem being studied. In this study, the reward is ranking of the performance obtained by the action. The details of how the performance of each action and the ranking are computed will be explained later. In general, the performance of an airline (or the NAS) is a multi-variable quantity. So, the comparison of performances must consider the multiple objectives. There are several techniques to measure and compare multi-variable efficiencies as summarized by Fonseca and Fleming[6]. Two examples are the *Data Envelopment Analysis* (DEA)[7], and *domination* [8][9]. DEA requires the computation of the Pareto surface of the system to successfully compare two efficiencies, i.e., the comparison relies on the relative performance with respect to the maximum possible. DEA is a type of *target vector* technique [6]. Domination was proposed to compute the Pareto surface of multi-objective (mainly genetic) algorithms. Domination is easy to implement and it does not require normalization of the metrics, or the computation of optimal values. The multi-variable performance values are compared directly. If the performance of the agent A is the vector of values $<a_1, a_2, \ldots, a_n>$ and the performance of the agent B is the vector $<b_1, b_2, \ldots, b_n>$, domination is computed as shown in Eq. (3). Domination is a *goal attainment* technique [6].

$$\begin{aligned} A\ dom\ B \quad iff \quad & \forall a_i \in A \land b_i \in B \mid a_i \geq b_i \\ & and \\ & \exists a_i > b_i \end{aligned} \quad (3)$$

If the performance of A is compared to the performance of all Bs, the count of the *true* results is a ranking of A in the universe of agent performances.

The type of domination described in Eq. (3) is binary, i.e., either A dominates B or it does not. There is no idea of how close is A to dominate B. This is a problem when domination is used in learning methods, because a binary relation does not help the progressive learning. At least in the first executions, no performance will dominate the others and the rank will be 0 for all the performances. Because of this, Eq. (4) proposes a modified (partial) domination that is used for this paper. The maximum value of *pdom* is 1 when all the values of the performance of A are equal to or greater than the corresponding values of B, but there are other *n* values smaller than one for As that are better than Bs in some of the values only. N is the size of the performance vectors. This is also a goal attainment technique [6].

$$\begin{aligned} A\ pdom\ B &= \frac{|C|}{n} \\ C &= \{a_i \mid a_i \geq b_i\} \end{aligned} \quad (4)$$

The rank of A relative to the historic records is obtained with Eq. (5), where *m* is the number of historic records.

$$R_A = \frac{1}{m}\sum_i^m A\ pdom\ B_i \quad (5)$$

The R_A value is used in Eq. (1) for the route *a*. The *B*'s are the performances obtained in previous executions for same state (*s*), but for other routes.

V. Methodology

The goal of this research is to include adaptive airline behavior in NAS-wide simulations (see Figure 1). The simulation developed for this research includes a NAS-wide traffic simulator and realistic demand for 24hrs of operations in the NAS. It also includes Java code to pre-process data, record and summarize outputs, adapt the airlines behavior, and to control the NAS-wide traffic simulator. The NAS-wide traffic simulator used in this case is

NASA's *Future ATM Concepts Evaluation Tool* (FACET) because of its Java API[5], its ability to connect to a database, its cumulated years of development, its record of successful applications, and availability (see Figure 2).

The input file is obtained from a tracking data of actual flights. The tracking data is filtered to contain only one record per flight. The record describes the time of first appearance during the simulation, the flight number, the aircraft type, the original coordinates of the flight (the flight could first appear when it is already flying or when it is still on the ground), the target (maximum) speed, altitude, and the flight plan. The input file is in the TRX format required by FACET, and it contains some other data for the flight not relevant for these simulations. This input file is pre-processed before it is fed into FACET. The preprocessing introduces the route selection that changes the flight plan, but not the O/D pair. It also introduces the push-back delays (if any). These changes make each execution of the simulation different from any other. This has the same effect as simulating a different day of the NAS but with equal demand level.

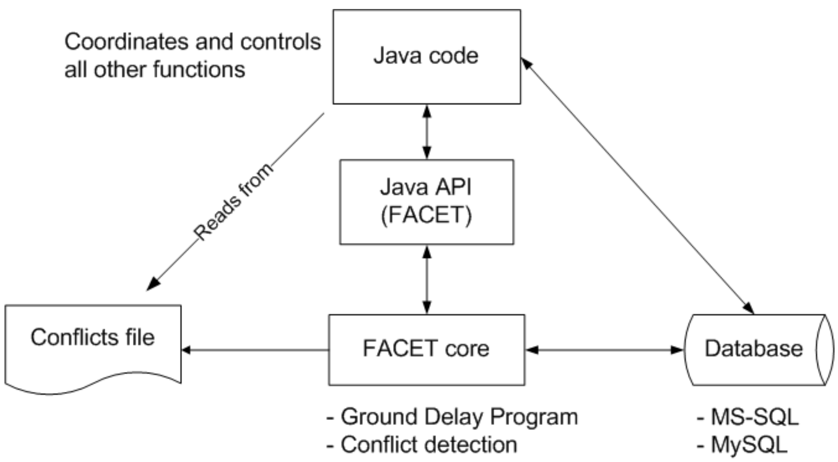

Figure 2: Implementation of the simulation

The outputs of the simulation are recorded in a database. The only exception is the conflicts. FACET does not provide, through its API, a way to count the conflicts during the simulation. It only writes the conflicts to a file. The summarization of this file is carried out by a parser after the simulation is finished. The values for conflicts are included in the airline and the system performance data.

The adaptation of the airlines route selection behavior is based on the historic information recorded in the database (from precious executions) and is implemented using reinforcement learning as described before. In this case, the state (s) in Eq. (1) is the combination of time of the day (just before the flight takes off) and O/D pair. The action (a) is a route selection for that state. The actual selection of the route is done using the algorithm described in Eq. (2). The alternative routes were obtained from sample files before the experiments and stored in the database. The direct Great Circle Distance route is always in set of alternate routes for any O/D pair.

The computation of the rewards (R) from Eq. (1) follows the algorithm in Eq. (5). The adaptation is done for all US airlines at the end of each execution of the simulation. The non-US airlines are grouped together in a single fictitious airline to save memory, database space, and reduce the execution time of the simulation.

A. Design of experiment

One of the goals of the research that supports this study is to evaluate the adaptive behavior in the presence of the new technologies and procedures proposed by NextGen[10][11]. In particular, the presence of the System Wide Information Management (SWIM) that will provide airlines, and other agents in the Air Transportation System, with more, more timely, and more accurate information.

The experiment presented in this paper is a stepping stone for more detailed studies about the effects of including adaptive airline behavior in a NAS-wide simulation when the NAS includes SWIM. The experiment has one *independent variable*: *availability of information* the airline agents can access. The variable can take two values: airlines have access to *system-wide and their own information*, or they have only access to *their own information*. This variable represents two cases. One in which SWIM is not present or its functionality is degraded, i.e., airlines only have their own information. And another case in which SWIM is present, i.e., airlines also have access to the system-wide information. In this paper, the information provided by SWIM is considered real-time and accurate. The NAS is considered deterministic, i.e., weather effects are absent, there is no push-back delays for the flights, and no errors in their speeds and altitudes.

The *dependent variables* are performance data for different agents. The performance of each flight is measured as follows:

[5] API: Application Program Interface.

- *Fuel burned* by the flight during the simulation.
- *Departure delay* if the flight takes off during the simulation and its destination is an OEP-35 airport. These departure delays are the result of adjustments made by FACET when the function of GDPs is activated with the arrival capacities of the OEP-35 airports set to VFR conditions (see Table 1).
- *Arrival delay* if the flight lands during the simulation (i.e., airborne delay is not considered). This delay is an approximation since the scheduled arrival time is not available in the input file. So the total distance to fly is computed from the initial coordinates of the flight, the flight plan, and the filed speed (all of them in the input file).

Table 1: Default VFR Airport Arrival Rates (AAR) for the OEP-35 airports used in the simulation

Airport name (ICAO)	Airport Arrival Rate (Moves per hour)	Airport name (ICAO)	Airport Arrival Rate (Moves per hour)
KATL	80	KLGA	40
KBOS	60	KMCO	52
KBWI	40	KMDW	32
KCLE	40	KMEM	80
KCLT	60	KMIA	68
KCVG	72	KMSP	52
KDCA	44	KORD	80
KDEN	120	KPDX	36
KDFW	120	KPHL	52
KDTW	60	KPHX	72
KEWR	40	KPIT	80
KFLL	44	KSAN	28
PHNL	40	KSEA	36
KAID	64	KIAH	72
KSFO	60	KJFK	44
KSLC	44	KLAS	52
KSTL	52	KLAX	84
KTPA	28		

The performance data of an airline are the cumulated performance values of its flights, i.e., there are values for *fuel burn*, *arrival delay*, and *departure delay*.

The performance of the NAS (system) includes the cumulated performances of the airlines, i.e., *total fuel burn*, *total departure delay*, and *total arrival delay*. The system performance also includes the *total number of airborne conflicts* and two more metrics. The first of these metrics is the *percentage of simulated time in which at least one sector was at or above its MAP*[6] (this is labeled *% time OL*[7] in the charts). The other metric is the *percentage of time in which at least one OEP-35 airport was overscheduled* in terms of arrivals per unit of time (this is labeled *% over sch*[8] in the charts).

An airline selects a route using Eq (2). So it selects either the route that has brought the most benefit in the past or, with lower probability, it takes any of the available routes to explore new options. This is the *route selection step* of the simulation. It is performed before the flights take off.

At the end of each execution of the simulation each airline retrieves historic performance data for its flights and joins these records with the performance data of the execution that has just ended. The performance record is a vector of 12 components. Each component is the value of one of the metrics explained above (3 from the flight itself, 3 from the airline, and 6 from the system). If the airline does not have access to the system information the 6 metrics for system are given a value of 0 in the vector. Using Eqs. (5) and (4), the airline ranks the performance vectors for each flight with respect to the historic performance. The airline uses Eq. (1) and the ranking to update its Q-functions. This is the *adaptation step* of the simulation; it is a continuous learning process.

Since there is only one independent variable with two values, the experiment in this paper has two *treatments*. Treatment 1 is when airline have *system-wide* and *airline* information. Treatment 2 is when airlines have *airline* information only. Each treatment is a sequence of 40 *executions* of the simulation. This sequence models 40 days of operations in the NAS, in which airlines learn and use the acquired knowledge to improve their performance.

[6] MAP: *Monitor Alert Threshold*. It is the maximum instantaneous aircraft count that can be safely handled by a sector controller.
[7] OL stands for *Overload*.
[8] Over sch stands for *over schedule*.

The first 20 executions of each treatment are used to *explore* the space of airline behaviors and get sensible values for the Q-functions. For these executions, λ = 0.5 and ε = 0.95 (see Eqs. (1) and (2)). This represents the situation in which an airline has no experience and needs to test its options. After these 20 executions the value of λ remains the same, but ε = 0.2 to better exploit the knowledge. This represents the situation in which the airline already has some knowledge and seeks improvement of its performance based on that knowledge.

The demand in the input file corresponds to the operations starting on August 17, 2006 from 8:00 GMT to 7:59 GMT (next day). This demand is reduced to a third of the total for about 20,000 flights in the NAS including international and domestic. A total of 1,000 airlines are represented in this input sample including big corporations and very small local airlines both national and international. But only about 60 domestic airlines are actually recorded for individual adaptation.

The parameters for conflict detection are follows: the surveillance zone is 120 nm; the look-ahead time is 0; the horizontal separation is 6 nm; the vertical separation below f1290 is 1000 ft; the vertical separation above f1290 is 1000 ft.

The arrival capacities for the OEP-35 airports are shown in Table 1 in arrivals per hour.

VI. Results

This section presents the results of executing the simulations for 40 days of NAS operations under treatments 1 and 2. The metrics for the system performance and the performance of selected airlines are shown in the form of charts. The values of all metrics are scaled to a [0,1] range ease their charting. The scaling is done as shown in Eq. (6). A scaled value of 1 indicates that the metric got the highest (worse) value observed across all executions for the treatment. A scaled value of 0 indicated that the metric got the lowest (best) value observed. The charts include the absolute minima and maxima to ease the "rescaling" process.

$$scaledValue = \frac{value - min}{max - min} \quad (6)$$

B. Results for treatment 1

Figure 1 shows the scaled values of the system metrics for 40 executions under treatment 1. The first 20 executions show high values of the metrics with high variations. The airlines behaviors were still being learned and airlines were selecting routes randomly 95% of the time (ε=0.95). After the 20 executions, all metrics show a trend

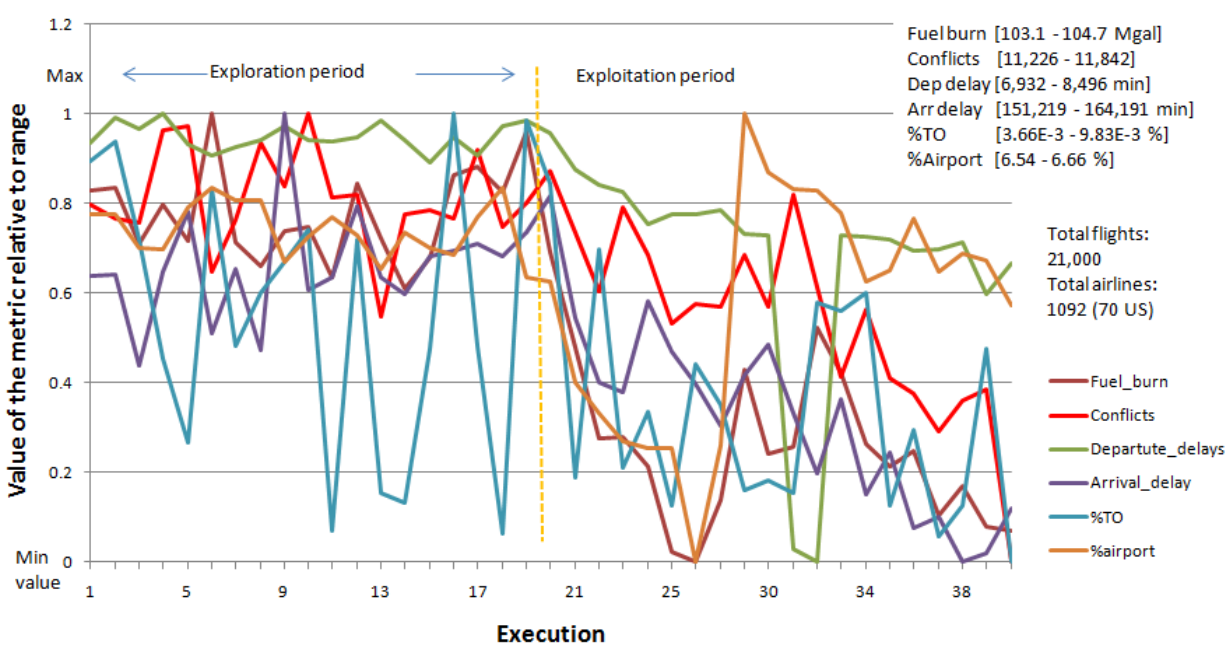

Figure 3: Scaled values of the system metrics for 40 executions under treatment 1

towards 0 (i.e., the lowest value of the range), but still with variations. The trend toward 0 is signal of the

Table 2: Airlines with more than 200 flights a day in the simulation

ICAO code	Name	Number of flights
AAL	American Airlines Inc.	1,130
SWA	Southwest Airlines Co.	1,029
COA	Continental Air Lines Inc.	739
DAL	Delta Air Lines Inc.	734
SKW	Skywest Airlines Inc.	503
UAL	United Air Lines Inc.	453
NWA	Northwest Airlines Inc.	428
ASH	Mesa Airlines Inc.	314
USA	US Airways Inc.	288
ASQ	Atlantic Southeast Airlines	277
FDX	Federal Express Corporation	241
FLG	Pinnacle Airlines Inc.	239
TRS	AirTran Airways Corporation	227
AWE	America West Airlines Inc.	205

effectiveness of the learning process to improve the system performance. The departure delay and the percentage of time with overscheduled destination airports (*% over sch* in the chart) show a slower reduction.

The big variations in the metrics from one execution to the next one could result from the interactions of the individual learning processes of the airlines. These processes are assumed to be independent, but they are really related and affect each other. The absolute values of the metrics could not be as big as suggested by the chart, since the chart shows values scaled relative to the range of variation. In fact, the percentage of time with overscheduled airports changes in total only 0.12% (or about 2 minutes), so in absolute terms the congestion at the destination airports remains equal with this treatment. That type of congestion is mostly compensated by the GDPs, implemented by FACET, at the expense of departure delays. Also, airlines might select different routes to avoid en-route congestion and get less affected by GDPs, these changes will be observed in fuel burn and arrival delays.

1. Results for airlines

For brevity, only the 14 US airlines with more than 200 flights per day are considered in the following analysis (see Table 2). The airlines in the table include the subsidiaries when the subsidiary flies only for the airline. For instance, American Eagle Airlines Inc. (EGF) flies only for American Airlines Inc. (AAL), so all its flights are counted as AAL flights. The same is true for Comair Inc. (COM) and Delta Airlines Inc. (DAL), and for Expressjet Airlines Inc. (BTA) and Continental Air Lines Inc. (COA). Other airlines like Skywest Airlines Inc. (SKW), Pinnacle Airlines Inc. (FLG), and Mesa Airlines Inc. (ASH) that serve several airlines or that split airlines are considered separately. From the 14 airlines, 3 are selected to show results in this paper. Southwest Airlines (SWA) was selected because it does not have a hub, but it flies point-to-point in the NAS (see Figure 4). United Airlines (UAL) was selected because it is has a hub in Chicago and it is more an “East-West” airline. Delta Airlines (DAL) was selected because it has a hub in Atlanta, and it is more a “North-South” airline. It is of interest in this paper to study these airlines’ learning process.

For Southwest Airlines (SWA), in Figure 4, the first 20 executions show high values for the metrics. The next 20 executions show the fuel-burn, arrival and departure delays reducing until they all converge to the lowest value observed in the last execution.

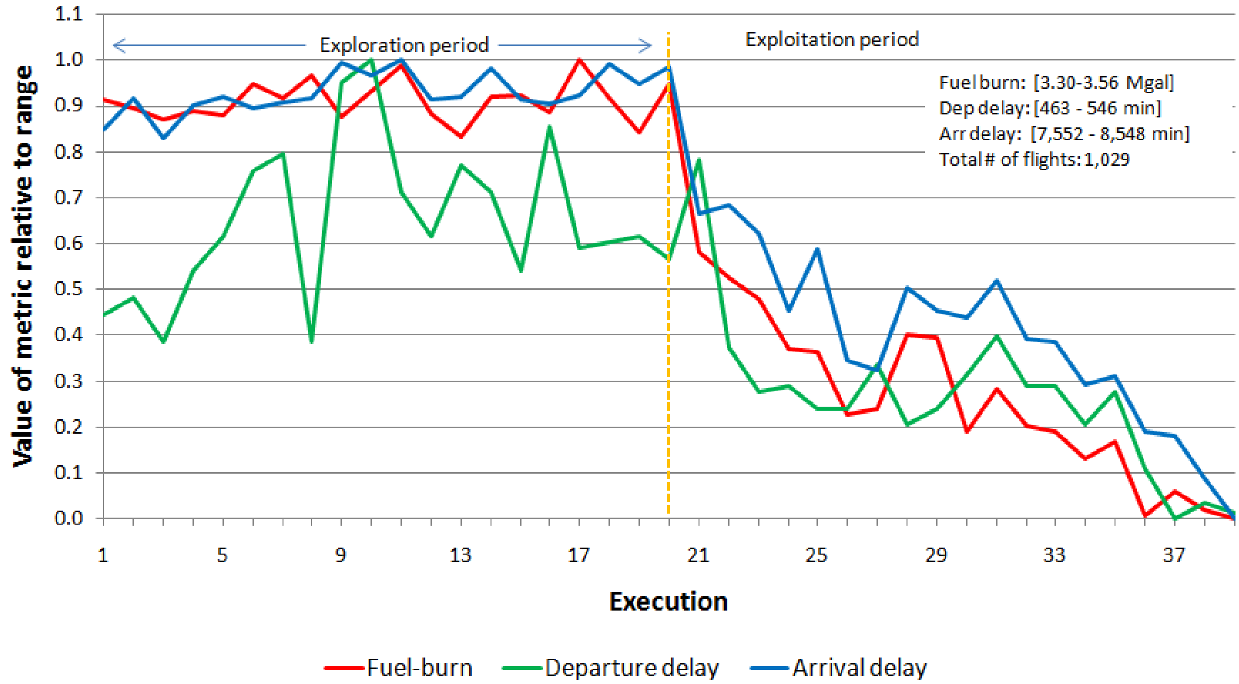

Figure 4: Scaled values of the metrics for Southwest Airlines (SWA) under treatment 1

For Delta Airline (DAL), in Figure 5, the first 20 executions show high values of the metrics with great variations for the delays (about 50% of their ranges). The next 20 executions show the fuel-burn, departure and

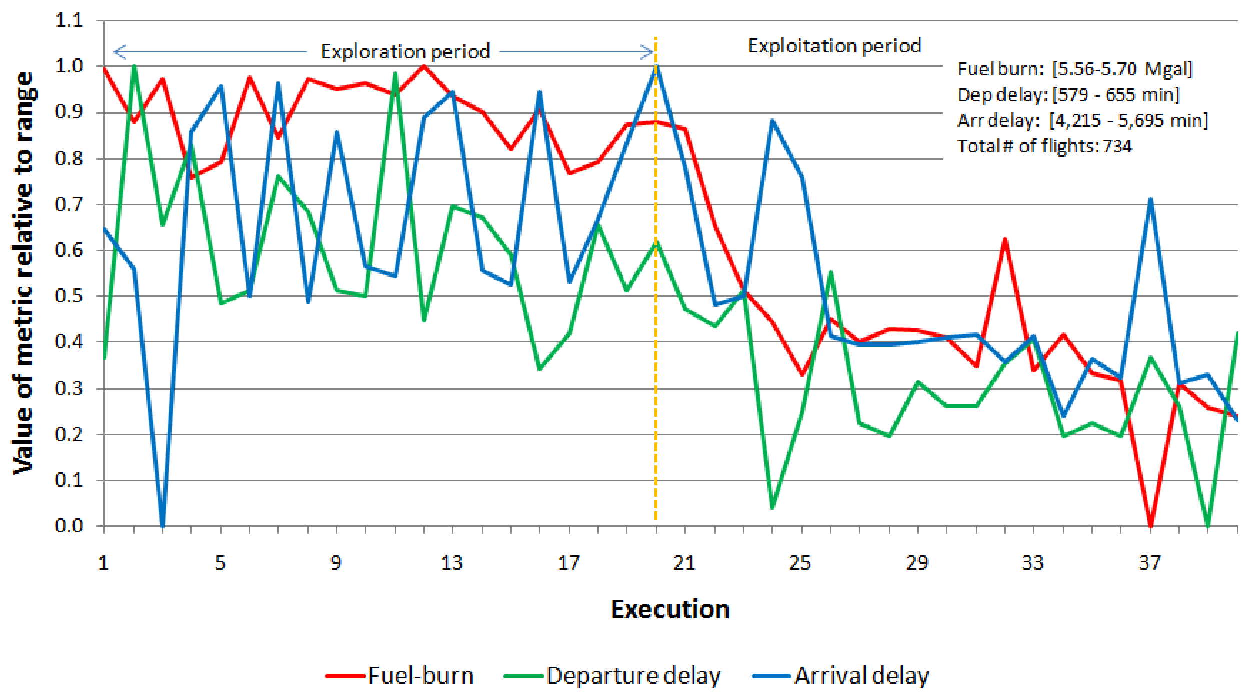

Figure 5: Scaled values of the metrics for Delta Airlines (DAL) under treatment 1

arrival delays reducing progressively.

For United Airlines (UAL), in Figure 6, the first 20 executions high values for the metrics and high variation (almost 100% of the range). The next 20 executions fuel-burn and departure delay reducing their values. The arrival delay reduces, but its variation is higher in the beginning.

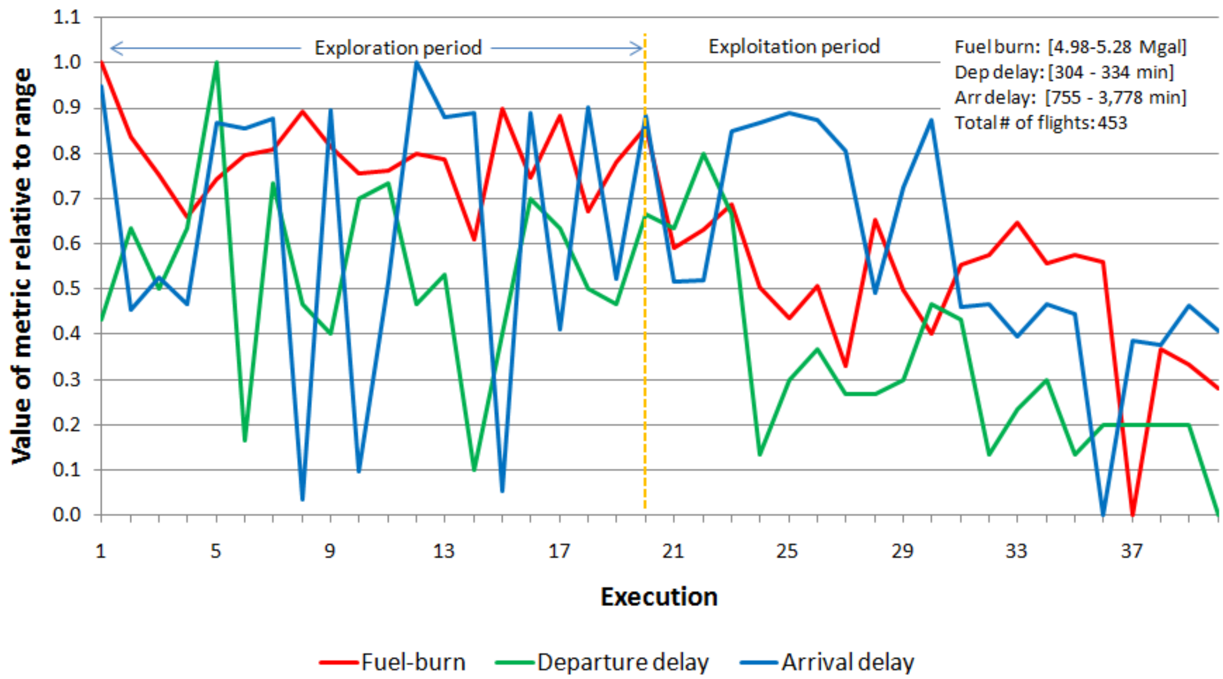

Figure 6: Scaled values of the metrics for United Airlines (UAL) under treatment 1

C. Results for treatment 2

Figure 7 shows the scaled metrics of the system for 40 executions under treatment 2, i.e., the airlines only have access to their own performance history, not the system's. In the first 20 executions, the system metrics show relatively high values with high variance. After execution 20 the metrics rapidly reduce their values showing that *the learning process is effective even when the information available to the airlines is more limited*. *The reduction of the metric values is faster in this treatment as it was in treatment 1*. The departure delay does not reduce as fast as the other metrics. Though there are some low values at executions 24 and 26, the value delay remains high in this treatment. This delay behaved similarly in treatment 1. The percentage of simulation time with at least one over scheduled destination airport (*% over sch* in the charts) reduced its value under treatment 2, but it did not under treatment 1.

The absolute values of the system metrics are similar between treatments. The minima and maxima of all the metrics for this treatment are of the same order of magnitude than the value for treatment 1, and the numerical values are also similar. *The system is not optimized by simply providing more information to the airlines*.

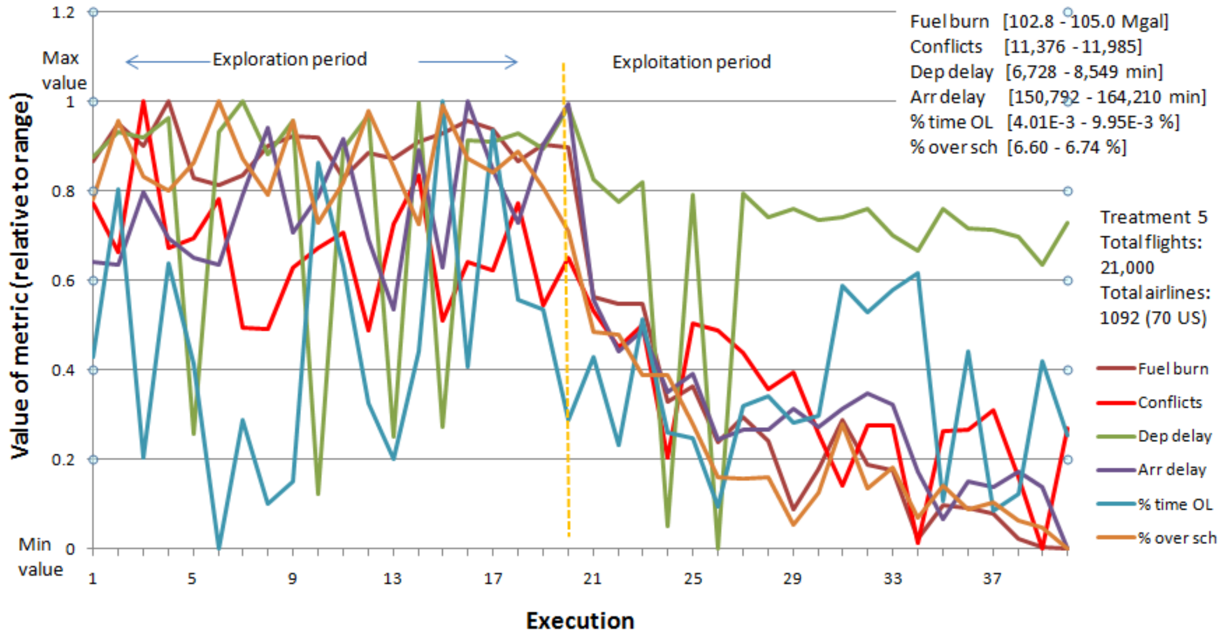

Figure 7: Scaled values of the system metrics for 40 executions under treatment 2

2. Results for the airlines

For Southwest Airlines (SWA), the metrics reduce faster in treatment 2 (see Figure 8) as they did in treatment 1 (see Figure 4). The metrics reach their minimum values between executions 30 and 33, whereas under treatment 1 they did between executions 36 and 40. The variation of the departure delay from execution to execution is high for

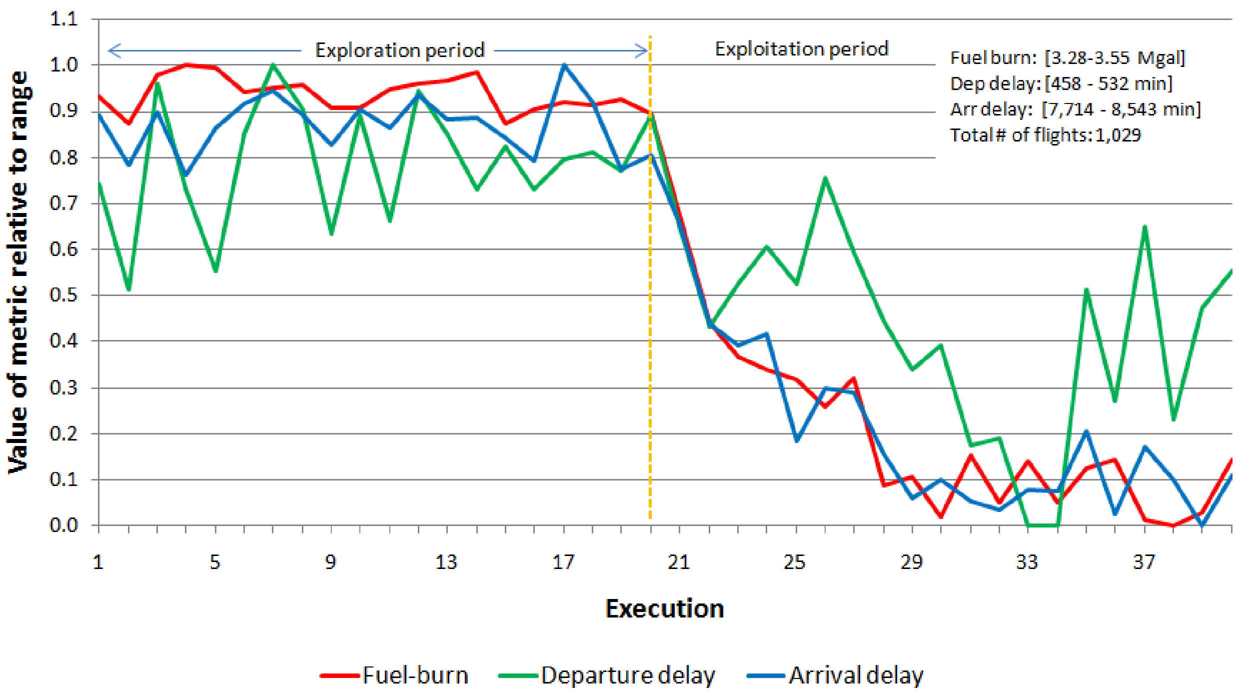

Figure 8: Scaled values of the metrics for Southwest Airlines (SWA) under treatment 2

this airline. That was not observed under treatment 1.

For Delta Airlines (DAL) the reduction is similar under both treatments (see Figure 9 and Figure 5). But the variation is higher under treatment 1.

For United Airlines (UAL) the reduction of values is also faster under treatment 2 (see Figure 10). But the arrival delay does not reduce as much. It actually tends to slowly reduce its average, but with high variations around it. Fuel and departure delay reduce clearly with small variations. Under treatment 1, the variation in these two metrics was

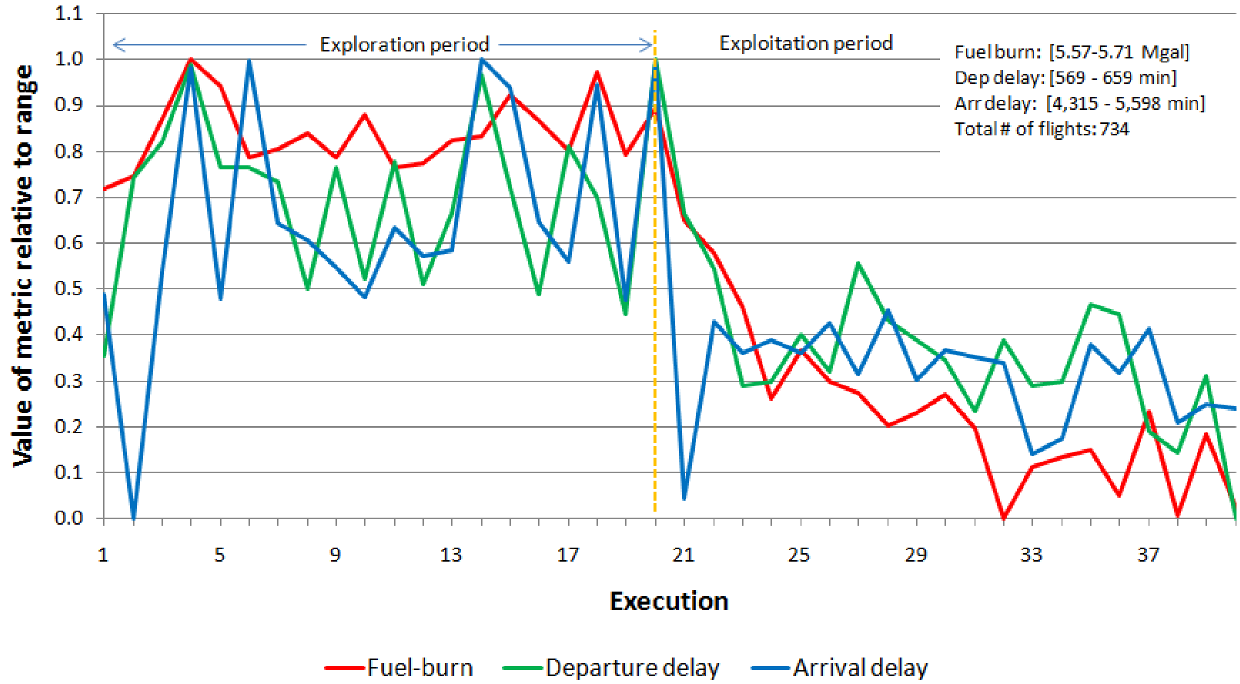

Figure 9: Scaled values of the metrics for Delta Airlines (DAL) under treatment 2

higher.

In terms of absolute values of the metrics, the ranges overlap in all the cases. The width of the ranges changes from treatment to treatment. But, treatment 2 tends to result in wider ranges of variation.

In all the cases, the introduction of adaptation bring benefits to the airlines in terms of reduction of fuel-burn, and at least one of the two types of delays (either departure of arrival). The total reduction in fuel-burn is in the range on 1.6 to 2.2 millions of gallons for the whole system. Treatment 2 results in the wider range of improvement (up to 2.2 million gallons). In the absence of weather effects (e.g., no wind) saving of fuel can only come from selecting shorter paths to fly as shown in previous studies [12][13]. Studying the routes selected by the airlines shows that the shortest path, i.e., the Great Circle Distance path between origin and destination, is one of the most frequently selected routes. The departure delay improves in the range of about 1,564 to 1,821 minutes being treatment 2 the one resulting in the widest improvement range (1,821 min). The arrival delay improves in the range of 12,972 to 13,418 minutes. Again treatment 2 results in the widest range of improvement.

Air Traffic Controllers will also benefit with both treatments since the total number of conflicts reduces. The improvement range is about 615 in both cases. With the low demand used in this paper, the metric for percentage of time with at least one overloaded sector has a very low absolute value. These values are also similar in magnitude between treatments. The benefit for the air traffic controllers in this case is not significant.

Finally, the percentage of time with at least one destination airport overscheduled is in the order of 6.6% for both treatments. Route selection is of little help for this metric under this level of demand. Previous studies with similar levels of demand [12][13] showed that airlines selecting shorter paths reduce their delays and the total number of conflicts in the system as well as their fuel-burn, but not the congestion at destination airports.

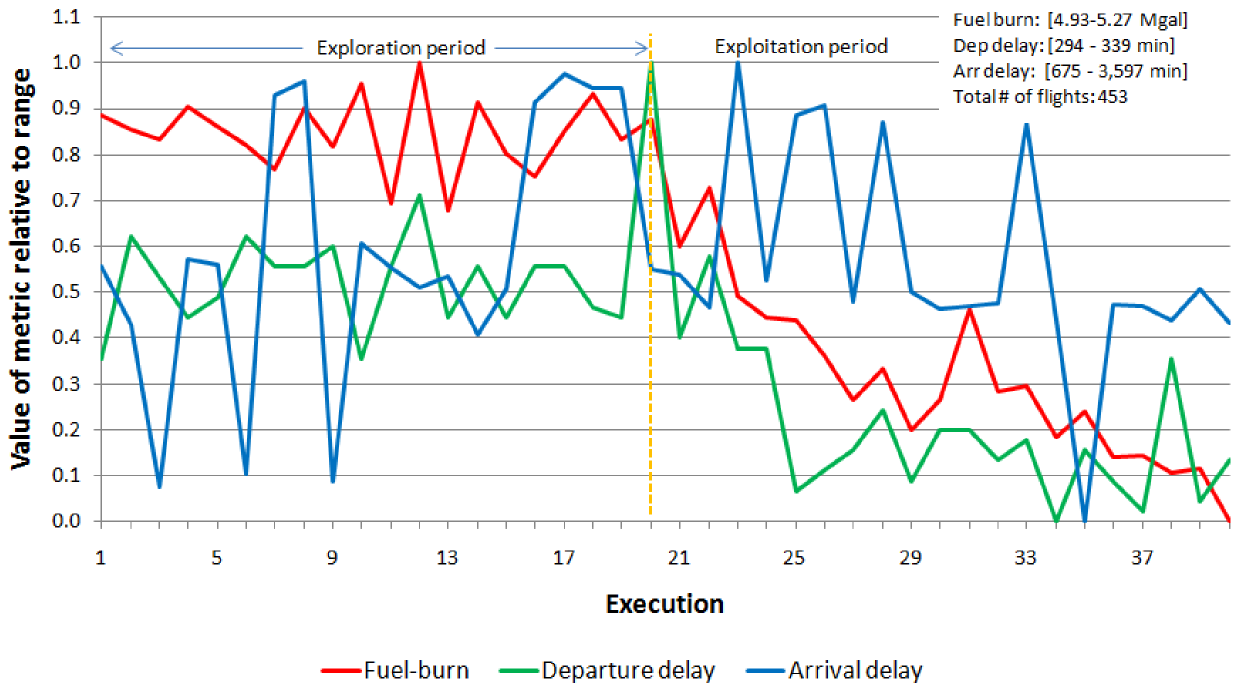

Figure 10: Scaled values of the metrics for United Airlines (UAL) under treatment 2

VII. Summary and future work

The experiment described in this paper suggests that including adaptive airline route selection behavior in NAS-wide simulations is effective. Airlines and system show decreasing values for the metrics (in this case smaller values are better) after some knowledge has been acquired by the airlines. A random choice of routes (as done in the first 20 executions of each treatment) does not bring any sustained improvement in the metrics. But it helps the learning process since after exploiting the knowledge acquired during this stage, the airlines and the system improve performance.

Despite the interaction between strategies that could arise in adaptive multi-agent systems, this experiment converged to better values of the metrics for two different treatments and tens of executions. The effects of the interaction are partially reflected in the variance of the results from one execution to the other: an airline that has been improving for several executions suddenly decreases its performance to improve again in the following executions.

Though all airlines benefit from the learning process, there are some differences worth commenting on. Under treatment 1, Southwest obtains reductions in all the metrics with low variations. Delta obtains mild reductions in all the metrics and the variation tends to be higher. United obtains also mild reductions but its variation is the highest in all metrics. Arrival delay even appears to be at the same level with high variations around it. Same is true for fuel-burn. Under treatment 2, Southwest obtains significant reductions in fuel-burn and arrival delay, but the reduction is departure delay is mild and the variation is high. Delta reduces all its metrics with lower variation. United obtains good reductions in fuel-burn and departure delay, but almost constant arrival delay and high variations. This is an indication that route structure plays a significant role in efficiency. Attempts to modernize need to take this into account. In summary, all the airlines converge faster to the lowest values of the metrics when they only have access to their own performance data than when they have access to the system performance data also.

In absolute terms, having only local information helps the airlines achieve wider ranges of improvement. The total reduction in fuel-burn is between 1.6 to 2.2 millions of gallons for the whole system. The departure delay improves between 1,564 to 1,821 minutes. And the arrival delay improves between 12,972 to 13,418 minutes. These are all benefits for the airlines since they imply cost reductions. For Air Traffic Controllers the reduction in the number of conflicts is about 615 conflicts. The other metrics do not show wide variations ranges perhaps due to the low demand used in this paper.

The design of the experiment will be extended to include three more independent variables with two values each for a total of 16 treatments (including the two treatment presented in this paper):

- *Stochasticity* of the NAS: exponentially distributed push-back delays (average is 25 min) and normally distributed flight speed errors (maximum value is 10 knots, i.e., $6\sigma = 10$)
- *Latency* of information: the performance data could be available immediately after they are computed at the end of the execution or some executions later.
- *Accuracy* of information: the airline can receive the data without errors or with some normally distributed noise.

The 14 extra treatments created after the extension must be analyzed the same way treatments 1 and 2 were in this paper. The goal of the extension will be to study the effects on the learning process of degraded SWIM operation (latency, inaccuracies) and of stochasticity in the NAS. Furthermore, the demand must be increased to its true level of about 60,000 flights instead of the 20,000 used in this experiment. The extension and the executions with increased demand are already being carried out in parallel using several computers.

Acknowledgments

The authors would like to acknowledge the contributions and help from the following persons and institutions.

The research this study is part of is funded by NASA (NRA NNA07CN32A). Furthermore, Natalia Alexandrov, Kapil Sheth, María Consiglio, Brian Baxley, Kurt Neitzke, and Shon Grabbe (NASA), have provided suggestions and comments throughout the whole research process.

From Sensis Corporation, George Hunter and Huina Gao, Matt Blake. From George Mason University, Dr. Thomas Speller Jr., Dr. Kenneth De Jong, Dr. Robert Axtell, Dr. George Donohue, Dr. John Shortle, Dr. Rajesh Ganesan, Dr. Paulo Costa, John Ferguson, and Keith Sullivan. From Metron Aviation, Jason Burke, Terry Thompson, Bob Hoffman, Bengi Manley, Arash Youseffi, Richard Xie, Jerome Lard, Michael Brennan, Giles O'Keefe, Bruno Miller, Alan Bloodgood and Norm Fujisaki. From FAA, Steve McMahon, Joe Post, Kimberly Noonan, Dan Murphy, Ralph Iovinelli and Tony Diana. From Southwest Airlines, Rick Dalton. From GRA Inc., Frank Berardino. From Reason Fundation, Robert Poole. From OpusTek, Kevin Niewoehner. From Daniel Webster College, Joakim Karlsson. From General Dynamics, George Emilio, DavidTuttle. From ONERA, Bruno Lamiscarre. From Oliver Wyman, Eric Ford, Robert Hazel. From NASA Yuri Gawdiak.

They have contributed to the improvement of the research.

Finally, thanks to the Ministerio de Ciencia y Tecnología (Minister of Science and Technology) of Costa Rica.

References

[1] K. Bilimoria, B. Sridhar, G.B. Chatterji, K. Sheth, and S. Grabbe, "FACET: Future ATM Concepts Evaluation Tool," *Air Traffic Control Quarterly*, vol. 9, 2001, pp. 1-20.

[2] K. Ramamoorthy, B. Boisvert, and G. Hunter, "A Real-Time Probabilistic Traffic Flow Management Evaluation Tool," *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, Oct. 2006, pp. 1-13.

[3] D.N. Sweet, V. Manikonda, J.S. Aronson, K. Roth, and M. Blake, "Fast-time simulation system for analysis of advanced air transportation concepts," *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, 2002, pp. 5–8.

[4] S. Russell and P. Norvig, *Artificial Intelligence. A Modern Approach*, USA: Prentice Hall, 2003.

[5] A. Agogino and K. Tumer, "Regulating Air Traffic Flow with Coupled Agents," *Proceedings of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, P. Padgham and Parsons, Eds., Estoril, Portugal: 2008, pp. 535-542.

[6] C.M. Fonseca and P.J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, vol. 3, 1995, pp. 1-16.

[7] M.J. Farrell, "The Measurement of Productive Efficiency," *Journal of the Royal Statistical Society. Series A (General)*, vol. 120, 1957, pp. 253–290.

[8] D.E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Reading, Massachusetts, USA: Addison-Wesley Pub. Co., 1989.

[9] K. Deb, "Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, vol. 7, 1999, pp. 205-230.

[10] JPDO, *Concept of Operations for the Next Generation Air Transportation System, Version 2.0*, Washington DC, USA: Joint Planning and Development Office, 2007.

[11] NIIO, *FAA's NextGen Implementation Plan*, Washington DC, USA: NextGen Integration and Implementation Office, 2009.

[12] G. Calderon-Meza and L. Sherry, "Analysis of Stakeholder Benefits of NextGen Trajectory-Based Operations," *Integrated Communications, Navigation and Surveillance Conference, 2010. ICNS '10*, 2010.

[13] G. Calderon-Meza and L. Sherry, "Establishing an Upper-Bound for the Benefits of NextGen Trajectory-Based Operations," *Electronic Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT 2010)*, 2010.