Adaptive Agents in NAS-Wide Simulations: A Case-study of COTP and SWIM

Guillermo Calderón-Meza (Ph.D. Candidate)
Center for Air Transportation Systems Research
George Mason University
Fairfax, Virginia, 22030, USA
Email: gcaldero@gmu.edu

Lance Sherry (Ph.D.)
Center for Air Transportation Systems Research
George Mason University
Fairfax, Virginia, 22030, USA
Email: lsherry@gmu.edu

***Abstract*—To support the development and evaluation of NextGen/SESAR concepts-of-operations and technologies, NAS-wide simulations and analysis methodologies are required to evaluate the feasibility and estimate the NAS-wide benefits. Of particular importance is the evaluation of unintended consequences that have historically been a roadblock to innovation in the NAS.**

This paper describes an analysis of adaptive airline behavior for flightplan route selection in the presence of Collaborative Trajectory Options Program (CTOP) and System Wide Information Management (SWIM). The results of analysis of simulation of 60,000 flight per day for 80 days shows that: (i) flightplan route selection reaches a system-wide equilibrium, (ii) the equilibrium state yields system-wide performance benefits, (iii) the equilibrium is achieved in 18 days, (iv) inaccurate and delayed information have no impact on system-wide performance but require additional days to achieve equilibrium, and (v) global (i.e. all airline) information does not improve individual airline route selection. The implications of these results on NextGen planning are discussed.

Keywords-NextGen, evaluation, conflicts, FACET, distance flown, delays.

I. Introduction

Concerns over unintended consequences and gaming are a significant issue for modernization initiatives and have historically been a roadblock for innovation and productivity improvement in the NAS. To support the development and evaluation of NextGen/SESAR concepts-of-operations and technologies, analysis methodologies and simulation infrastructure are required to evaluate the feasibility and estimate the NAS-wide benefits. State-of-the-art NAS-wide Simulations, capable of simulating 60,000 flights per day, have limited decision-making capability. This decision-making capability is static, not adaptive.

The real-world NAS, however, is a highly adaptive environment in which autonomous agents (e.g. airlines, air traffic control) are continuously adapting their decision-making strategies to minimize costs of operation. Further, analysis of an inventory of "gaming" scenarios in the NAS identified "adaptation" by agents (e.g. airlines) as the underlying mechanism for taking advantage of opportunities to increase productivity in the NAS.

The *Collaborative Trajectory Options Program* (CTOP) is one the NextGen initiatives The goal of CTOP is to allow airlines to submit alternate preferred flightplan routes that can be used in the event their preferred route affected by capacity reduction in the airspace. CTOP includes a centralized and collaborative algorithm with which the FAA assigns alternate flightplan routes to flights based on lists of route preferences (called *Trajectory Options Set*, TOS) provided by the airlines. System Wide Information Management (SWIM) is a communications and protocol for sharing real-time information about the status of all agents and resources in the NAS.

Airline Operation Centers (AOCs) will use SWIM to determine their preferred route sequence for COPT. Exactly what information will be available through SWIM has not been defined in detail, but in theory, SWIM could provide AOCs with own-ship information as well as aggregated, de-identified information about the performance of other airline's flights as well as airspace capacity and throughput.

This paper describes a case study of the effects of SWIM on airline route choice for COPT. The AOCs choose their preferred routes based on shortest distance criteria as well as historic performance of alternate routes (Figure 1).Each AOCs ranks the routes based on historic multi-objective performance and a Reinforcement Learning technique.

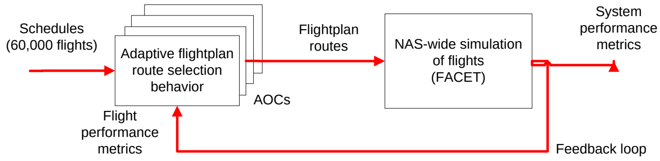

Schedules
(60,000 flights)
Adaptive flightplan route selection behavior
AOCs
Flightplan routes
NAS-wide simulation of flights (FACET)
System performance metrics
Flight performance metrics
Feedback loop

Figure 1: Integration of adaptive airline behavior in a NAS-wide simulator

The architecture and algorithms to implement adaptive behavior in NAS-wide simulators is described. An existing NAS-wide simulator, NASA's Future ATM Concepts Evaluation Tool (FACET), was used as the basis of the simulations, and was extended with the adaptation and route selection algorithms. Due to the stochastic nature of the adaptation process, the study also required quantitative methods to analyze the effects of adaptation on the performance and predictability of the NAS.

The results of the study indicate that the learning process of the AOCs takes 17 days stabilize when the information provided by SWIM is accurate, real-time, and global (not just from the airline itself). The number of days needed to reach stability in the learning process increases when the information is local (own airline), is delayed, or inaccurate. The magnitude of the increase can be as much as 11 days. In most of the performance metrics using local/global, accurate/noisy, or real-time/delayed information from SWIM showed marginal effects on performance, which is an indication of the robustness of the NAS.

This paper is organized as follows: Section 2 describes the methodology and the design of the experiment, the simulation used for the experiment, and the configuration and parameters used in the experiment, Section 3 describes the results of the experiment, and Section 4 provides conclusions, implications of these results, and future work.

II. METHOD AND DESIGN OF THE EXPERIMENT

This section describes the methodology, the design of the experiment, and the simulation environment used.

The idea of the case study is to demonstrate the integration of *adaptive airline behavior* in a NAS-wide simulation (see **Error! Reference source not found.**). In this study, the adaptive airline behavior is limited to the behavior of *dispatchers* working for the AOCs. It is assumed that each airline has a single dispatcher that selects the routes for all the flights of the airline in one day. Real dispatchers use criteria to select the routes. This study models the behavior of the AOC (dispatchers) with a probabilistic selection algorithm. The algorithm has a parameter that controls the probability of selecting the "best" route at the moment, and the probability of selecting any other alternate route randomly. This algorithm is known as *ε-greedy* [1]. The adaptation of the behavior is modeled by a *feedback loop* containing historic performance data for the individual flights, and a *Reinforcement Learning* algorithm [2] that ranks the routes based on the historic performance and "learns" the best route. The behavior of the AOCs is *non-cooperative* and selfish since AOCs do not coordinate their decision making process with other AOCs and they always try to minimize their own cost. The flight performance data used by the AOCs to rank the routes are the set of metrics composed by: fuel burn, departure delay, arrival delay, airborne conflicts, and number of congested sectors crossed[1].

The experiment was conducted using the *Future ATM Concept Evaluation Tool* (FACET) [3] complemented with multi-agent simulation tool, MASON [4], a database, and a Java application (see Figure 2). FACET has been used in previous studies [5][6][7] to evaluate new *Traffic Flow Management* (TFM) concepts in the NAS. FACET offers options like a Java API[2], and batch processing of input data. Without random inputs (e.g. weather) or changes in parameters, a simulation in FACET is deterministic. The results will be the same regardless of the number of executions. MASON is used to implement the flights and AOCs as autonomous agents. The database contains data for alternate routes between O/D pairs (i.e. no route generator was used in this study), and for performance of individual flights, airlines, and the whole NAS. The Java application (called Main Application in Figure 2) controls the other modules of the simulator, i.e. FACET, database, and MASON, and implements the adaptive AOC route selection behavior.

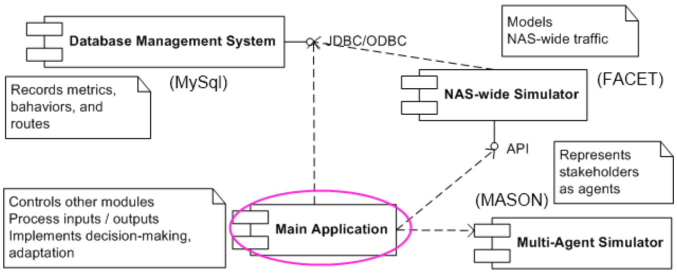


Figure 2: Software architecture of the NAS-wide simulator with adaptive agent behavior

A. The flight route selection algorithm

The main input to FACET is the flight schedule, flight tracks and cruise flight-levels. FACET accepts several formats for these input files known as ASDI and TRX. To achieve the goals of this experiment, a TRX input file was generated based on historical data from the Airline On Time Performance Data provided by Bureau of Transportation Statistics (BTS). The procedure for generating the TRX file is described below.

First, the sample TRX files that come with FACET were parsed and the O/D pairs and corresponding flight plans were extracted and exported to a database.

Second, the BTS Airline On-Time Performance (AOTP) data was queried to obtain a single day of domestic operations. The query extracted the O/D pair, the coordinates for the airports (taken from a proprietary table), the scheduled departure and arrival times, the flight and tail numbers, and the aircraft type (taken from a proprietary table related to On Time by tail number). The results of this query are sorted, ascending, by scheduled departure time.

For each record returned by the query the great circle distance of the O/D pair, the expected flight time (that is the difference of the scheduled departure and arrival times both converted to GMT), the required ground speed (and integer number of knots), the heading (an integer number computed from the coordinates of the airports assuming 0 degrees for

[1] A sector is congested if the number of flights in the sector in that moment is greater than the Monitor Alert Parameter (MAP) for the sector.

[2] API: Application Program Interface.

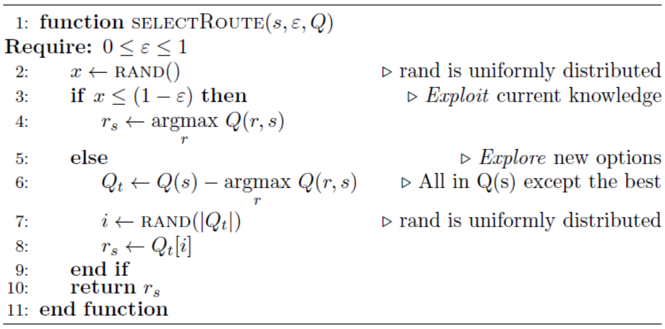
North heading, and 90 degrees for West heading), and the flight level (a uniformly distributed random integer number from 200 to 450), and the flight plan (taken randomly from available plans for the O/D pair). The coordinates of the airports are converted into integer numbers with the format [+|-]DMS where D stands for degrees (two or three digits), M stands for minutes (two digits), and S stands for seconds (two digits). FACET requires western longitudes to be negative.

Third, for each group of records with the same GMT scheduled departure time one "TRACKTIME" record is written to a text file. The value of the TRACKTIME record is the GMT scheduled departure date/time converted into the number of seconds from January 1, 1970 GMT. After this TRACKTIME record, the individual "TRACK" records for the flights are written using the data computed in the second step. The process repeats until there are no more records from the query. An input file generated this way does not track the flights through the *National Airspace System* (NAS). It only describes every flight with a single record. So this file can be used for simulation purposes only, not for playback in FACET.

B. *The implementation of adaptation*

The first part of the adaptive behavior of the AOCs is the route selection algorithm (see Figure 3)[1]. In this study, the route selection is done by the dispatcher at the AOC of the airline. In a CTOP, the selection will be performed by the FAA using the TOSs provided by the airlines.

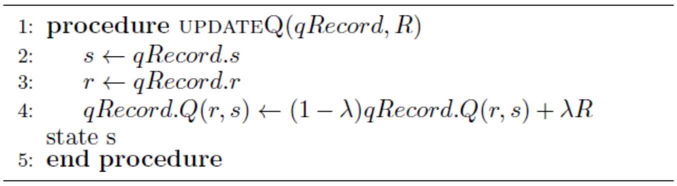
For the route selection algorithm, the knowledge of the dispatcher is modeled by a function (i.e. implemented by a matrix called, *Q-matrix*) that maps a *state s*, and a *route r* to a real number *Q* between 0 and 1 [2]. The state is a 3-tuple containing the origin and destination airports, and the scheduled departure time. The *Q-value* indicates the *value* (1 = best value, 0 = no value) of selecting route *r* in state *s*. From the airline perspective, this Q-matrix is a TOS for a CTOP. The parameter ε is a real number between 0 and 1. A value of 1 turns the algorithm into a purely random selection. A value of 0 turns it into a purely greedy algorithm that selects always the best route known to the moment of selection.



```
function SELECTROUTE($s, \varepsilon, Q$)
Require: $0 \le \varepsilon \le 1$
    $x \leftarrow$ RAND()                          ▷ rand is uniformly distributed
    if $x \le (1-\varepsilon)$ then                ▷ Exploit current knowledge
        $r_s \leftarrow \operatorname{argmax}_r Q(r,s)$
    else                                           ▷ Explore new options
        $Q_t \leftarrow Q(s) - \operatorname{argmax}_r Q(r,s)$   ▷ All in Q(s) except the best
        $i \leftarrow$ RAND($|Q_t|$)               ▷ rand is uniformly distributed
        $r_s \leftarrow Q_t[i]$
    end if
    return $r_s$
end function
```

Figure 3: Route selection algorithm (ε-greedy)

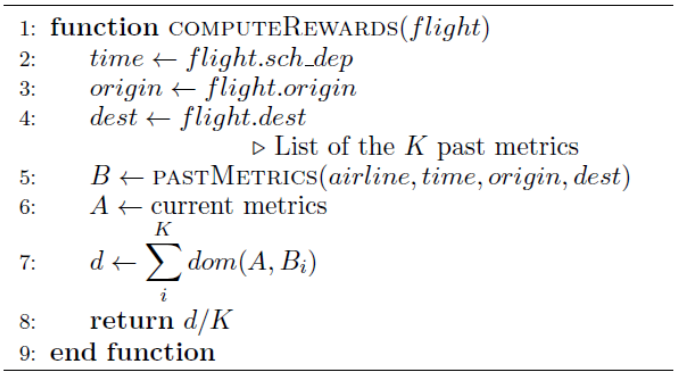
The second part is the actual adaptation of the Q-matrix, which is implemented by a Reinforcement Learning algorithm [2] (see Figure 4). The algorithm receives a *qRecord* (i.e., an element of the Q-matrix) and a *reward-value* that is a real number between 0 and 1. The parameter *λ*, *learning rate*, determines the importance given to the previous values of the qRecord with respect to the newly obtained reward. A value of 1 will turn make the algorithm follow the rewards obtained (i.e. oscillations are possible, and there will be no memory of previous results). A value of 0 will keep the algorithm from learning.



```
procedure UPDATEQ($qRecord, R$)
    $s \leftarrow qRecord.s$
    $r \leftarrow qRecord.r$
    $qRecord.Q(r,s) \leftarrow (1-\lambda)qRecord.Q(r,s) + \lambda R$
state s
end procedure
```

Figure 4: Adaptation of the Q-value using Reinforcement Learning algorithm (Q-learning)

The computation of the reward *R* is done for each flight at the end of a simulated day of operations using the algorithm in Figure 5. A *flight* contains the scheduled departure time (*sch_dep*), the *origin*, and the destination (*dest*) airports. A flight also contains a vector with its performance metrics, which are computed during the simulation. The list of vector with *K* performance metrics vectors for past flights with the same O/D pair and scheduled departure time is retrieved from the database.



```
function COMPUTEREWARDS($flight$)
    $time \leftarrow flight.sch\_dep$
    $origin \leftarrow flight.origin$
    $dest \leftarrow flight.dest$
                                  ▷ List of the $K$ past metrics
    $B \leftarrow$ PASTMETRICS($airline, time, origin, dest$)
    $A \leftarrow$ current metrics
    $d \leftarrow \sum_i^K dom(A, B_i)$
    return $d/K$
end function
```

Figure 5: Computation of reward for a flight

The performance vectors *A*, and *B*, are compared using the *dom* operator, a modified *domination*[8] operator defined in Eq. 1. The operator counts the number of times a member of a performance vector is greater than or equal to the corresponding member of the other vector.

$$\begin{array}{lrcr} Let & A & = & \langle a_1, a_2, \ldots, a_n \rangle \\ & B & = & \langle b_1, b_2, \ldots, b_n \rangle \\ then & dom(A,B) & = & |\{a_k | \forall k \text{ such that } a_k \ge b_k\}| \end{array} \quad (1)$$

The computation of rewards and the updating of the Q-values are done for all the flights in the simulated day that meet the following requirements:

1. The flight starts in a US airport

2. The flight is not a loop (i.e. its origin and destination airports are different)
3. The flight appears in the simulation for the first time when it is still at the origin airport (i.e. less than 10 miles from the coordinates of the airport)

The performance metrics used to rank flightplan route selections are:

1. Fuel burn (in kilograms)
2. Departure delay (in minutes)
3. Arrival delay (in minutes)
4. Distance (in nautical miles)
5. Number of airborne conflicts
6. Percentage of congested sectors crossed during the flight

The first four metrics are always available to the AOCs. The input data for the simulations does not include the scheduled arrival time for the flights, and some flights are not scheduled, hence the scheduled arrival time does not exist. Therefore, the arrival times are estimated during the simulation using the scheduled departure time, the distance of the flightplan, and the filed speed of the flight. The last two metrics are only available when SWIM provides them as part of the global information (see next section for details). An airborne conflict is signaled for a pair of aircraft if the distance separating them is less than 5 nm horizontally, or less than 1,000 ft vertically. A flight crosses a congested sector if the number of aircraft in the sector at a moment (time step of the simulation) is greater than the *Monitor Alert Parameter* (MAP) value defined for the sector. Each flight counts the number of congested sectors it crossed and divides it by the total number of sectors it crossed to obtain the last metric. The reason for choosing these metrics is that the dispatchers at the AOCs consider them (i.e. or equivalent metrics) to decide which route to select. The goal of the dispatchers is to optimize operations performance by reducing the value of all these metrics.

The performance of the NAS is measured with the following set of metrics:

1. Total fuel burn (in kilograms)
2. Total number of airborne conflicts
3. Total departure delay (in minutes)
4. Total arrival delay (in minutes)

Two more metrics were used to measure NAS performance, but the results showed that flightplan route selection does not have a significant effect on them. For brevity, these two metrics are not included in this paper.

Each flight and airline is modeled as an agent in the experiments. Flights are *explicit passive* agents. They are represented by Java objects in the simulation, but they do not actively make decisions. They exist to simplify the measurements of performance for the flights. Airlines (i.e. AOCs) are *active* agents since they select routes before the flights takeoff. In total, a simulation in these experiments contains about 69,000 agents for one day of operations.

C. Design of Experiment

The goal of this study is to evaluate the effect of degraded operation of SWIM in the performance of the NAS in the presence of adaptive AOC flightplan route selection.

TABLE I. DESIGN OF EXPERIMENTS

Experiment	**Independent variable**		
	Data availability	***Data latency***	***Data accuracy***
1	Global	Real-time	Accurate
2	Local	Real-time	Accurate
3	Global	Delayed	Accurate
4	Local	Delayed	Accurate
5	Global	Real-time	Noisy
6	Local	Real-time	Noisy

This paper presents and compares the multi-objective performance of the NAS when the AOCs compute the TOSs using the information provided by SWIM. The experiments explore the differences in performance when the operation of SWIM is degraded. Table I shows that the degradation is represented by three independent variables: the *availability* of data, the *latency* of the communication, and the *accuracy* of the communicated data. In the process of adaptation, the AOCs could have historic performance data of their own operations only (i.e. *local* data), or data from all airlines (i.e. *global* data). The communication of data could be *real-time* (available at the end of the day) or *delayed* one day. The data could be *accurate* or corrupted with a 30% noise (i.e. *noisy*).

To achieve the goal of the study with a small number of experiments and executions, 5 comparisons between experiments are done. The first comparison is between experiments #1 and #2 (see Table I) to determine the effect on NAS performance of the availability of global information. Two comparisons are done to determine the effect of delays in the communication of data. Experiments #3 and #1, and #4 and #2 are compared for this purpose. The last two comparisons are done to determine the effect of inaccuracies in the data. Experiments #5 and #1, and #6 and #2 are compared for this purpose. The comparisons are done by testing hypotheses for the mean and the variance of the system-wide performance metrics. The hypotheses are as follows:

$$\begin{array}{lcc} H_0^1 & : & \bar{x}_{global} = \bar{x}_{local} \\ H_0^2 & : & s_{global} = s_{local} \\ H_0^3 & : & \bar{x}_{delayed} = \bar{x}_{real-time} \\ H_0^4 & : & s_{delayed} = s_{real-time} \\ H_0^5 & : & \bar{x}_{noisy} = \bar{x}_{accurate} \\ H_0^6 & : & s_{noisy} = s_{accurate} \end{array} \quad (2)$$

Identify applicable sponsor/s here. *(sponsors)*

In this study, an experiment is a sequence of simulations of 24 hours of NAS operations. The input for a simulation is a schedule of the flights for the NAS complemented with information for the aircraft types, target speeds and altitudes and the coordinates of the flights when they first appear in the simulation. The same input file is used for each simulation assuming that the schedules are similar between operation days. The first simulated day of an experiment the AOCs select flightplan routes randomly among the list of alternatives. Every successive simulated day the AOCs acquire more knowledge and gradually commit more to selecting the best route known at the moment of the selection. The gradual change is achieved by reducing the value of ε from 1 down to 0.2. When ε reaches 0.2 at the 17th simulated day, the ACOs select the best route 80% of the times, and randomly 20% (see Figure 3). The set of these first days is known as the *exploration* period. The set of days after ε reaches 0.2 is known as the *exploitation* period.

The file used in this experiment contains 67,753 flights including domestic, international, general aviation, and private flights that occurred from August 17 2006 at 08:00:00 UTC to August 18 2006 at 07:59:00 UTC.

The λ parameter is set to 0.2 for all the experiments. The selection of this value is arbitrary, but it is customary in the Reinforcement Learning field to use low values. The ε parameter starts at 1.0 to select routes randomly. At the end of every simulated day, the value of ε is decreased a fixed amount until the value reaches 0.2 at the seventeenth simulated day. From that moment on ε = 0.2. This strategy is a special case of *ε –greedy*[1] strategy called *linear annealing*.

D. *Metodology of analysis*

The analysis of results starts by measuring the effect of the independent variables in the time needed by the learning process to reach steady state. This information helps determining if the simulation is applicable to real-world situations in which a rapid response is required, e.g. to compute the TOSs for a CTOP.

The second step is the comparison of the NAS performance at the first simulated day (i.e. when AOCs select routes randomly) to the performance at the day in which the learning process reaches steady state. The goal of this comparison is to determine the effect of adaptation in the flightplan route selection.

Figure 6 illustrates the third step of the analysis, which determines the effects of the independent variables on the each system-wide performance metric measured during the exploitation period of the simulations. A set of values for a metric obtained during the exploitation period is a *data sample*. The values in this set are considered *Independent and Identically Distributed* (IID). This statistical analysis was performed using Minitab 16. The 2-sided 2-samples unpaired t-test provided by Minitab performs well with normal and non-normal samples, even with non-equal standard deviations, if the sample size is large enough. The 2-sided 2-samples test for standard deviations is non-parametric and compares the standard deviations directly; the test performs well if the samples sizes are large enough. All samples in the experiments are large (i.e. more than 20 data points) for the tests to be accurate regardless of the distributions of the samples.

The *significance level*, α, is 0.5 for all the hypothesis tests. Therefore, the *null hypotheses* are rejected when the *p-value* of the test is smaller than α.

No external disturbances are included during the simulations, i.e. there are no restrictions due to weather, congestion, push-back delays, or other stochastic events. The only source of stochasticity in the simulations is the interaction between the learning processes of the AOCs.

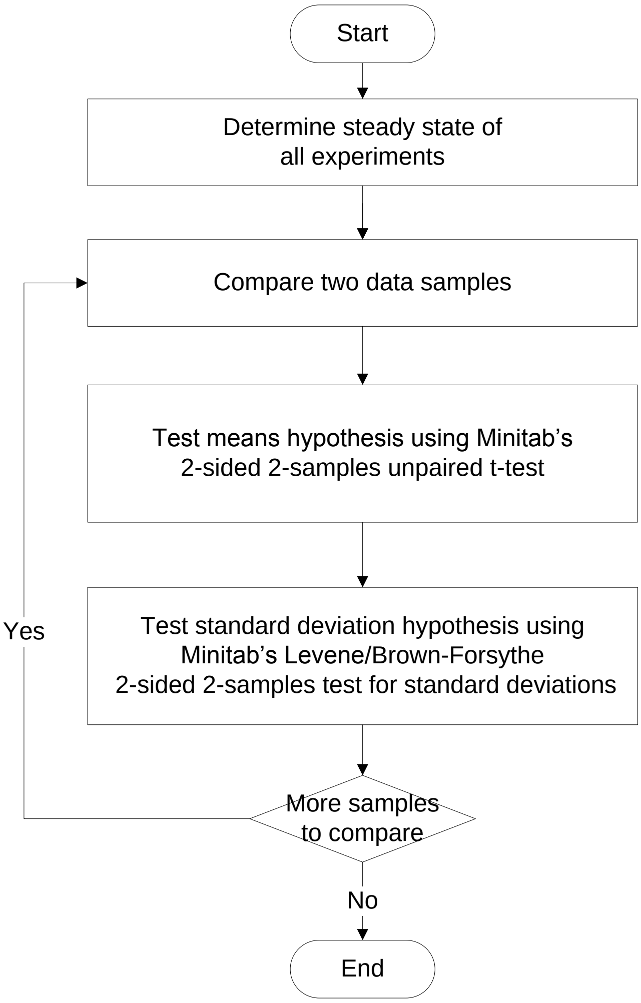

Start
Determine steady state of all experiments
Compare two data samples
Test means hypothesis using Minitab's 2-sided 2-samples unpaired t-test
Test standard deviation hypothesis using Minitab's Levene/Brown-Forsythe 2-sided 2-samples test for standard deviations
More samples to compare
Yes
No
End

Figure 6: Flow chart for the statistical analysis of simulation outcomes

The arrival rates of the 35 busiest airports in the US (OEP[3]-35 airports) are set to the VFR[4] values for the entire day (see Table II). The arrival and departure rates of the other airports

[3] OEP: Operational Evolution Partnership Plan.

[4] VFR: Visual Flying Rule.

are set to their defaults (i.e. infinite). These limitations are the only source of departure delays for the flights[5].

E. *FACET settings used in the experiments*

FACET takes its input from batch files loaded Java API. The simulation outputs are stored in a database (see Figure 2).

TABLE II. 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
KHNL	40	KSEA	36
KIAH	72	KSFO	60
KIAD	64	KSLC	44
KJFK	44	KSTL	52
KLAS	52	KTPA	28
KLAX	84		

FACET was configured to detect airborne conflicts through the whole NAS with the following parameters: the surveillance zone is 120 nm, the *horizontal separation* is 5 nm, and the *vertical separation* is 1,000 ft.

III. RESULTS

This section presents the results of the six experiments and the comparisons described in the methodology section.

A. *Steady state*

An experiment simulates a number of days before the learning process of the AOCs reaches a steady state. This study introduces two rules to empirically determine when a simulation reaches steady state.

1. When 60% or more of the total Q-records has a value greather than 0, and
2. When 10% or less Q-values change 10% or less at the end of a simulated day.

Both rules must be met simultaneously for the steady state to be considered reached (see Figure 7). Table III presents the number of simulated days needed for the experiments to reach the steady state. In general, having only local information requires more time to reach stability when the experiments are compared to experiment having global information. The effect of latency in the data is observable in longer times to reach stability when compared to real-time data. However, experiments having inaccurate information take longer to reach steady state when compared to similar experiments having accurate data regardless of whether the data are real-time or delayed.

TABLE III. NUMBER OF SIMULATED DAYS NEEDED BY THE EXPERIMENTS TO REACH STEADY STATE

Experiment	**Simulated days to reach steady state**
1	17
2	18
3	21
4	26
5	23
6	32

All the experiments start with AOCs selecting routes randomly. While knowledge is acquired, AOCs gradually reduce the probability of selecting routes randomly and increase the probability of selecting the best route known so far. This gradual change ends when the ε parameter becomes constant (i.e. 0.2) and the learning process reaches steady state.

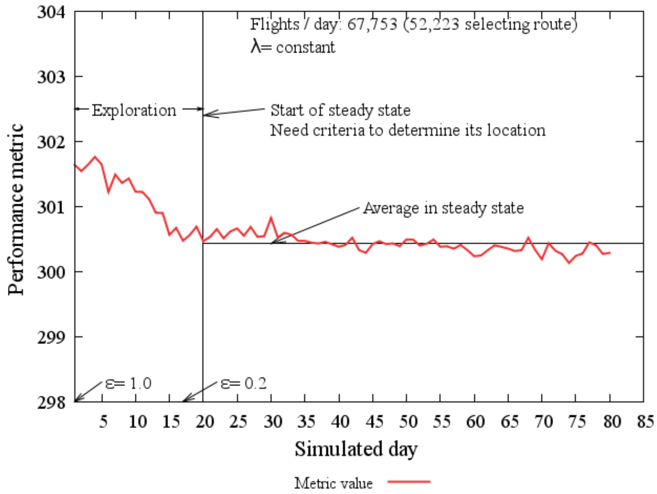


Figure 7: Typical behavior of a system-wide performance metric during the exploration period, and exploitation period (start of steady state)

[5] The departure delays are generated by the Ground Delay Programs (GDP) functionality of FACET.

B. *Effect of the inclusion of learning*

It was observed that the system-wide metrics, in general, reduce their values during the *exploration period*. The reductions are due to the inclusion of learning in the route selection process (see Table IV). The reductions are negligible delays are negligible since they are in the order of seconds per flight. In average, each flight in the simulation in involved in about 2 conflicts (the distribution is not symmetric). Therefore, a reduction of 2,000 conflicts is equivalent to reducing the number of flights in a day by 1,000, which is a significant reduction (≈ 1%). The reductions in fuel are equivalent to millions of dollars saved for the entire NAS, but the measuring the differences flight-by-flight in the real world would not be practical.

TABLE IV. IMPROVEMENTS IN THE SYSTEM-WIDE METRICS DUE TO THE INCLUSION OF LEARNING

Metric	Absolute reduction	Per flight reduction
Fuel burn	-85,097 to 752,935 kg/day	-1.7 to 14.8 kg/day
Airborne conflicts	2,041 to 5,160	N/A
Departure delay	3,436 to 7,171 min/day	3 to 6.4 sec/day
Arrival delay	10,508 to 20,308 min/day	9.3 to 18.0 sec/day

C. *Hypotheses testing for global vs. local data*

The hypotheses testing step of the analysis is divided into the tests for means and the tests for the variances (see Figure 6). For the interpretation of results, the hypotheses are rejected when $p < 0.05$. A rejection of the hypotheses means that there is a significant difference between the experiments. Negative differences indicate decreases when the factor is present. Positive differences indicate increases. The differences are only meaningful when they are statistically significant.

TABLE V. HYPOTHESIS TESTING FOR THE MEANS OF METRICS FROM EXPERIMENTS #1 (GLOBAL DATA) AND #2 (LOCAL DATA)

Metric	P-value	Difference (%)	C.I.
Fuel burn	0.000	174,607 (0.10)	148,745 / 200,469
Conflicts	0.000	-1,713 (-2.80)	-1,808 / -1,617
Departure delay	0.483	340 (0.40)	-615 / 1,295
Arrival delay	0.000	1,347 (0.20)	726 / 1,967

The only significant difference in Table V is in the number of conflicts that show a reduction equivalent to reducing 850 flights per day[6]. The changes in fuel and delays are negligible. The reason for a reduction in conflicts is that global information makes conflicts data available to the decision making process of the AOCs whereas conflicts data are not available in the local data. Therefore, AOCs optimize for conflicts when they use global information, but they do not when using local data only.

TABLE VI. HYPOTHESIS TESTING FOR THE STANDARD DEVIATIONS OF METRICS FROM EXPERIMENTS #1 (GLOBAL DATA) AND #2 (LOCAL DATA)

Metric	P-value	Percentage difference
Fuel burn	0.465	22.6
Conflicts	0.000	90.8
Departure delay	0.065	-15.1
Arrival delay	0.020	-21.0

Table VI shows the significant increase in the standard deviation of the conflicts, and the smaller decrease in the arrival delay when global data are available. The changes in fuel burn and departure delay are not significant.

D. *Hypotheses testing for global delayed vs. global real-time data*

This section studies the effect of introducing 24 of delay to the global data available to the AOCs.

TABLE VII. HYPOTHESIS TESTING FOR THE MEANS OF METRICS FROM EXPERIMENTS #3 (DELAYED DATA) AND #1 (REAL-TIME DATA)

Metric	P-value	Difference (%)	C.I.
Fuel burn	0.218	-17,271 (0.00)	-44,866 / 10,323
Conflicts	0.023	127 (0.20)	18 / 236
Departure delay	0.000	3,012 (2.00)	2,293 / 3,731
Arrival delay	0.000	1,807 (0.30)	1,280 / 2,334

The only significant difference shown in Table VII is in the arrival delay, an increase in the delay caused by the delay in the global data available to the AOCs. The changes in the means are, economically, negligible even for those that are statistically significant. The type of adaptation used in these experiments, improves the robustness of the decision-making process so that it is not sensitive to a delay of 24 hours in the data.

TABLE VIII. HYPOTHESIS TESTING FOR THE STANDARD DEVIATIONS OF METRICS FROM EXPERIMENTS #3 (DELAYED DATA) AND #1 (REAL-TIME DATA)

Metric	P-value	Percentage difference
Fuel burn	0.745	-7.7
Conflicts	0.064	-20.3
Departure delay	0.024	-41.9
Arrival delay	0.780	-8.2

[6] The average number of conflicts per flight with the current input file is close to 2 per flight.

Table VIII shows that all standard deviations decrease when delayed data are available. The standard deviation of the departure delay changes significantly with the introduction of delays in the global data. All the other standard deviations do not change significantly. The presence of delays in the global data does not affect the standard deviations of the metrics.

E. Hypotheses testing for local delayed vs. local real-time data

This section studies the effect of introducing 24 of delay to the local data available to the AOCs.

TABLE IX. HYPOTHESIS TESTING FOR THE MEANS OF METRICS FROM EXPERIMENTS #4 (DELAYED DATA) AND #2 (REAL-TIME DATA)

Metric	P-value	Difference (%)	C.I.
Fuel burn	0.445	-9,553 (0.00)	-34,256 / 15,150
Conflicts	0.000	409 (0.70)	337 / 480
Departure delay	0.000	1,569 (0.50)	768 / 2,370
Arrival delay	0.019	711 (0.10)	119 / 1,304

When the information is local (see Table IX), the presence of delays in the data significantly increases the conflicts and the delays, but leaves the fuel burn unchanged. In this case also, the changes in the means are, economically, negligible even if they are statistically significant. The type of adaptation used in these experiments, improves the robustness of the decision-making process so that it is not sensitive to a delay of 24 hours in the data.

TABLE X. HYPOTHESIS TESTING FOR THE STANDARD DEVIATIONS OF METRICS FROM EXPERIMENTS #4 (DELAYED DATA) AND #2 (REAL-TIME DATA)

Metric	P-value	Percentage difference
Fuel burn	0.596	4.8
Conflicts	0.180	17.1
Departure delay	0.000	-59.1
Arrival delay	0.002	-35.9

Table X shows significant decreases in the delays with the introduction of delays in the local data. The standard deviations of fuel and conflicts do not change significantly.

F. Hypotheses testing for global noisy vs. global accurate data

This section studies the effect of introducing corruption in the global data available to the AOCs.

TABLE XI. HYPOTHESIS TESTING FOR THE MEANS OF METRICS FROM EXPERIMENTS #5 (NOISY DATA) AND #1 (ACCURATE DATA)

Metric	P-value	Difference (%)	C.I.
Fuel burn	0.000	410,362 (0.10)	32,146 / 438,578
Conflicts	0.000	-2,247 (3.70)	-2,462 / -2,032
Departure delay	0.049	708 (0.00)	3 / 1,413
Arrival delay	0.000	-5,761 (1.00)	-6,452 / -5,071

When the information is global (see Table XI), the presence of inaccuracies in the data significantly decreases the conflicts and the departure delays, but marginally increases fuel burn. The improvement in delays is negligible since it represents saving of seconds per flight. The reduction of conflicts is counter intuitive. The hypothesis to explain the reductions considers the way inaccuracies are introduced. The added noise is proportional to the actual values of the metrics. Hence, metrics with greater values are more affected than metrics with smaller values. For single flights, the number of conflicts averages 1.8, the departure delays is about 3 minutes, and the arrival delays are about 15 minutes. The average fuel burn for flights is the order of tens of kilograms. Therefore, fuel burn will be affected more than the conflicts and the delays because its value is higher.

TABLE XII. HYPOTHESIS TESTING FOR THE STANDARD DEVIATIONS OF METRICS FROM EXPERIMENTS #5 (NOISY DATA) AND #1 (ACCURATE DATA)

Metric	P-value	Percentage difference
Fuel burn	0.958	-5.1
Conflicts	0.000	122.3
Departure delay	0.015	-47.2
Arrival delay	0.001	42.6

As Table XII shows, the presence of inaccuracies increases the standard deviations of the conflicts and the arrival delay, but decreases the standard deviation of the departure delay. The variance of the fuel burn does not change significantly.

G. Hypotheses testing for local noisy vs. local accurate data

This section studies the effect of introducing corruption in the local data available to the AOCs.

TABLE XIII. HYPOTHESIS TESTING FOR THE MEANS OF METRICS FROM EXPERIMENTS #6 (NOISY DATA) AND #2 (ACCURATE DATA)

Metric	P-value	Difference (%)	C.I.
Fuel burn	0.000	506,895 (0.20)	477,413 / 536,376
Conflicts	0.000	-583 (-0.90)	-664 / -501
Departure delay	0.545	-354 (-0.20)	-1,509 / 801
Arrival delay	0.000	-7,275 (-1.30)	-8,115 / -6,434

When the information is local (see Table XIII), the presence of inaccuracies in the data significantly increases the fuel burn and decreases the arrival delays. Conflicts and departure delays decrease negligibly since their decreases represent 290 less flights per day and savings of seconds per flight. The reduction of arrival delays is counter intuitive. The same hypothesis used to explain the reduction in the metrics with global information can be used in this case.

TABLE XIV. HYPOTHESIS TESTING FOR THE STANDARD DEVIATIONS OF METRICS FROM EXPERIMENTS #6 (NOISY DATA) AND #2 (ACCURATE DATA)

Metric	P-value	Percentage difference
Fuel burn	0.047	30.7
Conflicts	0.050	33.7
Departure delay	0.853	7.6
Arrival delay	0.511	23.0

The presence of inaccuracies increases the standard deviations of the fuel burn and conflicts (see Table XIV). The variances of the delays do not change significantly.

IV. CONCLUSIONS

This study presents the results of comparing six experiments each consisting of repeated simulations (80 times) of 24 hours of operations in the NAS (67,753 flights). The simulations extend FACET functionality by including adaptive AOC behavior in the flightplan route selection process. The model of adaptation is a form of *Reinforcement Learning*, known as *Q-learning*, based on flight performance metrics obtained at the end of each simulated day of operations. The model for the flightplan selection process is a type of *ε-greedy strategy* that resembles a *linear annealing*, ε starts at 1.0 and decreases a constant value each simulated day until it reaches a constant value and remains at it for the rest of the experiment. AOCs select routes from a database of alternatives. Both models reasonably abstract the behavior of a *dispatcher* working at an AOC. Weather is excluded to focus the study on the inclusion of adaptation in the behavior of the AOCs and on the effect of degraded functionality of SWIM (e.g. delays, noise).

The results show performance improvements of the NAS improves with the inclusion of learning in the behavior of the AOCs, e.g. reduction of 14.8 kg of fuel burned by each flight, and a reduction of conflicts equivalent to taking 2,500 flights out of the NAS.

The learning process of the AOCs reaches a steady state after some days of exploration. The day in which the steady state is reached depends on the independent variables and it goes from 17 to 32 simulated days. The system performance metrics also reach a steady state after the exploration period ends.

The availability of global data reduces the number of conflicts (3%), but increases the standard deviation (90%), i.e. reduction of predictability. Fuel and arrival delays marginally increase with global data because the AOCs trade fuel for conflicts when the global information provides them with conflicts data.

The presence of latency in the communication of data results in marginal increases in the number of conflicts and the delays (2% at most). The standard deviations remain similar to when no delays are present in the data, except for the delays in which reductions of 41% and 60% are observed.

The presence of inaccuracies in the data results in reductions in some metrics and increases in other metrics. The decrease in conflicts is 3.7% at mist, and the decrease in arrival delays is 1.3% at most. Fuel burn increases less than 0.5%. The changes in standard deviations are mixed too. One hypothetical explanation for the observed reductions is that noise is penalizing more severely the metrics with higher absolute values, since noise is proportional to the actual measured value. Therefore, fuel burn is being negatively affected, but conflicts are not.

Future Work

More sensitivity analyses are needed to gain knowledge about the effect of several parameters on the results. For instance, the value of the learning rate (λ) and strategy to change ε should be explored. The sensitivity to different values of the independent variables like days of latency, or magnitude and distribution of noise must be determined.

The mixed results obtained in the simulations (increases combined with decreases) suggest that the computation of rewards used in the learning algorithm could be modified to become total or partial orders in the mathematical sense. The current operator used to compare performance vectors is an approximation of domination, but it is does not preserve transitivity as it is required by any operator used in optimizations. Alternatives like Data Envelopment Analysis (DEA)[9] should be explored. The rewards should allow actions (flightplan route selections) to quickly become dominant in a simulation if rapid adaptation is needed.

Weather effects, represented by winds and reductions in the capacity of airports and sectors, should be included in the simulations to make them more realistic.

ACKNOWLEDGMENT

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, and Kurt Neitzke, all NASA employees, have provided suggestions and comments throughout the research process.

George Hunter and Huina Gao from Sensis Corporation have also provided suggestions and comments through the whole research process.

From George Mason University the authors would like to acknowledge the help from Dr. Thomas Speller Jr., Dr. Kenneth De Jong, Dr. Robert Axtell, Dr. George Donohue, Dr. John Shortle, Dr. Rajesh Ganesan, Maricel Medina-Mora, John Ferguson, and Keith Sullivan. They have all contributed to the improvement of the research in their particular areas of expertise. From Metron Aviation, the authors acknowledge the help and contribution of Jason Burke, Bob Hoffman, Terry Thompson, and Norm Fujisaki. From FAA, the authors acknowledge the help and contribution of Joe Post and Tony Diana. Finally, thanks to the Ministerio de Ciencia y Tecnología (Minister of Science and Technology) of Costa Rica.

References

[1] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, Mass: MIT Press, 1998.

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

[3] K. Bilimoria, B. Sridhar, G. B. Chatterji, K. Sheth, and S. Grabbe, “FACET: Future ATM Concepts Evaluation Tool,” *Air Traffic Control Quarterly*, vol. 9, no. 1, pp. 1-20, 2001.

[4] S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan, “MASON: A New Multi-Agent Simulation Toolkit,” in *Proceedings of the 2004 SwarmFest Workshop*, Michigan, USA, 2004.

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

[6] B. Sridhar, G. B. Chatterji, S. Grabbe, and K. Sheth, “Integration of Traffic Flow Management Decisions,” *American Institute of Aeronautics and Astronautics*, 2002.

[7] R. Jakobovits, P. Kopardekar, J. Burke, and R. Hoffman, “Algorithms for Managing Sector Congestion Using the Airspace Restriction Planner,” 2007.

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

[9] M. J. Farrell, “The Measurement of Productive Efficiency,” *Journal of the Royal Statistical Society. Series A (General)*, vol. 120, no. 3, p. 253–290, 1957.