

# A Simulator for Modeling Aircraft Surface Operations at Airports

Zachary Wood \*

*Univ. of Calif., Santa Cruz, Moffett Field, CA, 94035, U.S.A.*

Matthew Kistler †

*Univ. of Calif., Santa Cruz, Moffett Field, CA, 94035, U.S.A.*

Sivakumar Rathinam ‡

*Texas A & M, College Station, TX, 77840, U.S.A.*

Yoon Jung §

*NASA Ames Research Center, Moffett Field, CA, 94035, U.S.A*

Rapid prototyping of integrated taxiway and runway schedulers at airports has become highly desirable for Safe and Efficient Surface Operations (SESO) to support the Next Generation Air Transportation System (NextGen).<sup>1</sup> Finding optimal solutions for taxiway and runway scheduling are generally computationally difficult and therefore it is essential to develop fast algorithms that can produce high quality solutions for the scheduling problems. Equally important is to develop a fast-time simulation that can be integrated with these algorithms and can handle high traffic demand in the presence of uncertainties. This paper describes the development of Surface Operations Simulator and Scheduler (SOS<sup>2</sup>) that can facilitate the integration of several scheduling algorithms with the surface simulation of the aircraft. The SOS<sup>2</sup> architecture is used to implement First-Come-First-Served (FCFS) heuristics and a comparison to the average taxi times of actual data from Dallas Fort Worth International (DFW) airport with SOS<sup>2</sup> output is made. The integration of a dynamic programming departure scheduler with FCFS taxiway heuristics demonstrates the modular architecture of SOS<sup>2</sup> and a comparison to the output of pure FCFS heuristics demonstrates the potential system-wide savings in taxi delay over FCFS scheduling.

## I. Introduction

Under the NextGen<sup>1</sup> vision, a simulation tool to integrate various surface traffic planning algorithms is required in support of the SESO milestones. This simulation tool requires a modular architecture for scheduling algorithms and fast-time capabilities. There are existing airport simulators<sup>2,3</sup> that have been available to the air transportation community for some time. These simulators typically provide a high level of detail and were designed for analyzing airport capacities under current-day operations. To simulate current-day operations, a non-modular architecture<sup>3</sup> is often times adequate, although some simulators provide rule-based systems<sup>2</sup> to compensate for potential changes in traffic controller operations. A rule-based system usually uses localized heuristics and does not have the capability of supporting optimization algorithms. An example of such an optimization algorithm would be formulating the taxi scheduling problem

\*Development Engineer, Univ. of Calif., Santa Cruz, Mail Stop 210-6, Moffett Field, CA 94035. Zachary.P.Wood@nasa.gov.

†Aerospace Engineer, Univ. of Calif., Santa Cruz, Mail Stop 210-6, Moffett Field, CA 94035. Matthew.S.Kistler@nasa.gov.

‡Assistant Professor, Department of Mechanical Engineering, Texas A & M University, College Station, TX 77840. SRathinam@tamu.edu.

§Aerospace Engineer, NASA Ames Research Center, Mail Stop 210-6, Moffett Field, CA 94035. Yoon.C.Jung@nasa.gov. AIAA member.

at an airport as a Mixed Integer Linear Program (MILP) and solving the MILP using a generic branch-and-bound solver like CPLEX<sup>6</sup> or GLPK.<sup>7</sup> Another example of an optimization algorithm would be finding the departure sequence for a given runway that optimizes delay<sup>12</sup> discussed in Section IV.

With the development of surface management technologies envisioned by NextGen, SESO requires the capability to rapidly prototype various optimization routines to analyze potential benefits and shortcomings of applying the solutions produced. Rathinam et al<sup>4</sup> showed significant savings to taxi times while maintaining safety through a MILP formulation of a node-link model of the east side of DFW airport. While encompassing all aspects of surface operations into a single optimization model produces the best theoretical solution, the execution-time to solve such models can be prohibitive in application. Dividing the airport into smaller regions, or sub-regions, and decoupling runway management from the taxiway can allow optimization models to execute in fast-time. With such computational gains, application of various optimization models to the sub-regions of the airport could become feasible. Although fast-time capabilities can be achieved when applying optimization models to these sub-regions, the schedules produced still need to be integrated into an efficient solution for the entire airport.

This paper describes the development of a fast-time simulation tool called the SOS<sup>2</sup>, a simulation platform with a plug-in architecture that can facilitate the integration of several scheduling algorithms with the surface simulation of the aircraft. In Section II the SOS<sup>2</sup> architecture is described. Using the SOS<sup>2</sup> architecture, implementation of localized schedulers into two different airport schedulers that will be focused on in later sections. The first scheduler described in Section III is Baseline-FCFS, which has pure FCFS heuristics and will be used as a benchmark for comparison of the schedulers currently under development in SESO; a comparison of average taxi times is made between Baseline-FCFS and actual data for calibration of the aircraft dynamics module implemented in SOS<sup>2</sup>. The second scheduler described in Section IV is Enhanced-FCFS, which has pure FCFS heuristics implemented for the taxiway and a dynamic programming module to optimize delay at the departure runways; a comparison is made between Enhanced-FCFS and Baseline-FCFS to demonstrate the modular framework of SOS<sup>2</sup> and improvements upon pure FCFS scheduling heuristics.

## II. SOS<sup>2</sup> Architecture

SOS<sup>2</sup> is a tool for the integration of optimization algorithms with aircraft simulation implemented in C++. The input requirements include a node-link model of the airport, aircraft model information, departure separation times, and runway configuration. Given a list of the aircraft to be scheduled, SOS<sup>2</sup> outputs time-based clearances for each node in aircraft routes. To calculate the time-based clearances, SOS<sup>2</sup> accepts additional input from scheduling algorithms. These algorithms could be local heuristics, such as FCFS described in Sections III and IV, or optimization algorithms such as those described in Sections III.A and IV.A. An alpha version of an OpenGL<sup>10</sup> Graphical User Interface (GUI) used for development and demonstration purposes is shown in Fig. 1 of a SOS<sup>2</sup> simulation at DFW airport; SOS<sup>2</sup> is independent of the GUI. The node-link model used is overlaid on a satellite image from GoogleEarth.<sup>9</sup> The GUI uses color to display some aircraft information with unimpeded aircraft within 75% of the nominal speed in green, unimpeded aircraft less than 75% of the nominal speed in yellow, braking aircraft in orange, and non-moving aircraft in red. The size of the rectangle representing an aircraft corresponds to the length and wingspan of the aircraft.

To ensure safety, a polygon is created around each node to represent an intersection on the taxiway. The dimensions of this intersection is derived from the maximum expected dimension of any aircraft scheduled in the airport, which is either the maximum expected wingspan or the maximum expected length of the aircraft, and the angle between adjacent links. By creating such a polygon to represent an intersection, any aircraft stopping short of this polygon can be assured that there is no chance of collision when another aircraft passes through the intersection regardless of aircraft model. Another safety mechanism of SOS<sup>2</sup> is a dynamic separation distance when an aircraft is taxiing in trail. This dynamic separation distance is derived from the desired minimum spacing of stationary aircraft on the taxiway, the stopping distance of the aircraft in trail, and an additional safety factor for pilot reaction time; the values for the minimum spacing of stationary aircraft and the pilot reaction time are variable inputs to SOS<sup>2</sup>.

A plug-in interface for aircraft dynamics provides many options for modeling aircraft surface movements from simple models using instantaneous acceleration to very detailed models based upon throttle position. For the purposes of this paper an intermediate model has been adopted. Deceleration or braking is modeled using a negative acceleration that remains constant over a single time-frame; a maximum expected deceleration is given for runway exits and for aircraft on the taxiway. Thrust is modeled using a positive acceleration



Figure 1. A screen shot of the SOS<sup>2</sup> GUI showing the movements of the aircraft at DFW airport overlaid on a satellite image from Google Earth Mapping Service.

that remains constant over a single-time frame; a maximum expected acceleration for aircraft on the surface is given. Separate functions for aircraft braking and aircraft thrust are provided to SOS<sup>2</sup>. These functions for modeling aircraft dynamics provide a means to generate detailed speed profiles for each aircraft. For aircraft braking, the stopping distance is provided and the braking function calculates the necessary deceleration over the next time-frame. For aircraft acceleration there are two options. If no required time of arrival (RTA) at a node is given, the thrust function returns the maximum expected acceleration. Otherwise if an RTA at a node is given, the thrust function calculates the necessary acceleration to meet that RTA provided it does not exceed the maximum expected acceleration nor the maximum taxi speed of the aircraft.

To create a generic framework for scheduling algorithms, SOS<sup>2</sup> hosts a scheduling interface to allow schedulers to input their solutions into SOS<sup>2</sup> *control points*. On the taxiway, each node is considered a single control point. Input into each control point is a sequence of aircraft passing through the control point ordered on the required time of arrival (RTA) at the control point. Although the aircraft RTAs at the control points is not required by SOS<sup>2</sup>, inputting the RTAs with the sequence allows SOS<sup>2</sup> to attempt meeting the RTA using the plug-in aircraft dynamics. If the RTAs are omitted, then SOS<sup>2</sup> attempts to move the aircraft at nominal speed using the thrust function described above.

A runway manager has been implemented in SOS<sup>2</sup> to manage departure and arrival runways. An input requirement is a matrix of departure separation times due to wake vortices as well as separation times between departure and crossings aircraft to ensure safety. Since separation times must be maintained and aircraft can cross runways in parallel, the *departure nodes* representing departure points and the *crossing nodes* representing the divide between runway and taxiway must be linked in control. This link in control for a runway causes the sequence input to SOS<sup>2</sup> for these runway nodes to take the form of a single control point that can allow multiple crossings simultaneously. Management of arrival runways is handled slightly differently since ground controllers do not have control over arrival runways. Arrival aircraft are accepted as input to SOS<sup>2</sup> with a time window, a specific time, and the arrival runway. During the time window, SOS<sup>2</sup> has all crossing aircraft hold short of the arrival runway. At the specific time within the time window, the arrival aircraft is considered to have arrived at the taxi exit and is controlled by SOS<sup>2</sup>. For both arrival and departure runways, SOS<sup>2</sup> ensures enough open pavement is available on the taxiway on the opposite side of the crossing to ensure crossing aircraft do not stop on a runway.

To integrate aircraft dynamics, scheduling, runway management, and safety parameters, SOS<sup>2</sup> uses a *look-ahead distance* in front of each aircraft. This look-ahead distance is based upon the maximum stopping distance of the aircraft derived from the maximum airport movement area speed. If a node appears in an aircraft's look-ahead distance, SOS<sup>2</sup> checks whether the aircraft is the next scheduled aircraft through that node. If an aircraft is maintaining proper separation distances and is the next scheduled aircraft through the nodes within the look-ahead distance, then SOS<sup>2</sup> attempts to drive the aircraft to meet the RTA or at the nominal taxiway speed based upon the model of the aircraft and provided aircraft dynamics. If the aircraft is in danger of violating a separation distance or is not the next scheduled aircraft at a node within the look-ahead distance, then SOS<sup>2</sup> applies braking acceleration according to the aircraft dynamics. These calculations are time driven and calculated in a timeframe-to-timeframe fashion similar to Euler's method.<sup>11</sup>

### III. Baseline-FCFS

A node-link model with 283 nodes of DFW airport in south-flow configuration using all runways and static routes based on the standard taxi routes was input to SOS<sup>2</sup>. Brinton et al<sup>5</sup> showed that almost 90% of aircraft on major taxi routes at DFW airport are handled using FCFS, but only 50% of aircraft are given spot clearances using FCFS. Because only 50% of the clearances at the spots are managed using FCFS, the times of the spot clearances in the actual data was input to SOS<sup>2</sup> as the planned aircraft schedule instead of inputting the times aircraft arrive at the spot. By using the times of spot clearances as input, nearly 100% of spot clearances for departure aircraft in the actual data are aligned with spot clearances in the SOS<sup>2</sup> simulations. For the purposes of these simulations, the controllers at DFW are assumed to use FCFS to sequence aircraft. The Surface Operations Data Analysis and Adaptation (SODAA)<sup>8</sup> tool is used to compile the actual data from DFW airport. Baseline-FCFS described below in Section III.A is used to provide a comparison of the SOS<sup>2</sup> simulations using FCFS heuristics with actual data.

Approximate values for the weight classes including small, large, heavy, and B757 were input to SOS<sup>2</sup> as the aircraft models. An acceleration model is input with a constant time and speed values to represent the spool-up time of an aircraft below a speed threshold, where the spool-up time is the amount of time an aircraft waits before acceleration to be applied if it attempts to accelerate and the speed threshold is the point an aircraft's speed has slowed to the point rolling friction takes over and it now must wait for a spool-up time whenever attempting to accelerate. If an aircraft attempts to accelerate, a constant acceleration is applied to the aircraft for the next timeframe if the spool-up time is satisfied or the aircraft is not below the speed threshold, otherwise the aircraft must wait for the spool-up time to be satisfied before the acceleration is applied. The surface data at DFW airport is obtained from SODAA for May 24th and June 11th, 2008. These days were chosen due to clear weather, unrestricted visibility, and low delays. Clear weather was chosen since SOS<sup>2</sup> currently does not use velocity profiles for bad weather days. The same spot times and arrival times from SODAA data were input to SOS<sup>2</sup> as push-back times and runway exit times respectively. Any metering constraints imposed upon departure fixes during the scenarios are not considered since that information was not available in the actual data at the time the simulations were conducted.

#### III.A. Scheduling Algorithms

A queue balancing heuristic operates prior to running SOS<sup>2</sup>. Given the aircraft input to SOS<sup>2</sup> for a specified runway, the departure queues are assigned using round-robin scheduling in the order of aircraft spot times. Aircraft on the taxiway are assumed to already have a queue assigned from a previous iteration of the queue balancing heuristic.

FCFS on the taxiway is modeled locally at each taxiway node. A list is generated of all aircraft within the look-ahead distance of a node that is not taxiing in trail of another aircraft to the node. If the aircraft already exist in the sequence at the node, then FCFS removes those aircraft from the generated list. The aircraft remaining in the list are then sorted by the estimated time of arrival at the node and appended to the sequence already at the node.

FCFS is applied to the departure runways with a similar procedure. All aircraft within the look-ahead distance of a departure node or crossing node are put in a list. If the aircraft are already in the departure runway sequence, then the aircraft are removed from the list. The minimum wait-time of the aircraft in the list is computed where the wait-time is the time which an aircraft requests to use the runway. If the aircraft with the minimum wait-time of the list is a departure aircraft, the aircraft will be appended to the departure runway sequence and removed from the list. If the aircraft with minimum wait-time is a crossing aircraft, the aircraft will be appended with other parallel crossing aircraft to the departure runway sequence and they will be removed from the list. This procedure is repeated until the list is empty.

#### III.B. Results

A comparison is made between the actual data derived from SODAA and the Baseline-FCFS output. In Figs. 2-5 a notable difference can be seen in average taxi times with arrival aircraft having significantly less taxi times than Baseline-FCFS and departure aircraft having significantly more taxi times than Baseline-FCFS. For the May 24th scenario there is a 10.2% difference on average arrival taxi times and 16.9% difference on average departure taxi times; for the June 11th scenario there is a 10.7% difference on average arrival taxi times and 18.6% difference on average departure taxi times. Some of the difference in arrival taxi times can

be attributed to some arrival aircraft landing on departure runways 17R and 18L which have shorter taxi routes than arrival runways 17C and 18R, while all aircraft in Baseline-FCFS landed on 17C and 18R. Since metering constraints were not considered in these simulations, some of the difference in departure taxi times could be contributed to aircraft waiting in the departure queues to meet a miles-in-trail restriction that was not imposed in the Baseline-FCFS simulation.

Although the usage of the center runways and metering constraints in the actual data likely contribute to the differences in average taxi times, it does not seem to contribute to all of the difference. A detailed analysis of the data revealed a significant number of arrival aircraft had negative delays and the majority of aircraft with negative delays were arrivals; negative delays indicates an aircraft travelling faster than the average unimpeded speed. The average unimpeded speed was derived from SODAA and determined to be 15 knots for all aircraft on the taxiway and 30 knots for arrival aircraft on a runway exit. The Baseline-FCFS simulation used the average unimpeded speed on the taxiway for both arrival and departure aircraft. These negative delays based upon the average unimpeded speed of aircraft implies that arrival aircraft are likely travelling at a higher average speed than departure aircraft. Indeed, SODAA data for the large weight class, which constitute the majority of the traffic and DFW airport, shows arrival aircraft travel on average at 18 knots on the taxiway and departure aircraft at 16 knots on the taxiway.

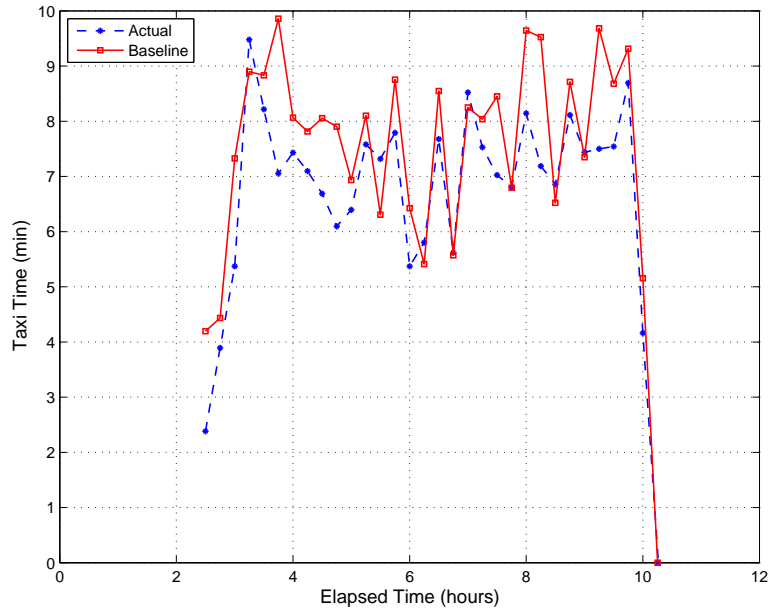


Figure 2. Comparison of average taxi times every 15 minutes for arrivals from the SOS<sup>2</sup> Baseline-FCFS simulation with the actual data at DFW airport for May 24th.

## IV. Enhanced-FCFS

To demonstrate the ability to quickly integrate various schedulers within SOS<sup>2</sup>, Enhanced-FCFS was developed using the algorithms described below in Section IV.A. The same node-link model, dynamics module, aircraft routes, and scenarios of May 24th and June 11th are used as in Section III. A comparison of the results between Enhanced-FCFS and Baseline-FCFS is made below in Section IV.B. Since the input is identical to both simulations, any gains in delay of Enhanced-FCFS over Baseline-FCFS can be interpreted as potentially realizable savings in the absence of departure fixes and if departure and arrival aircraft had the same average speed.

### IV.A. Scheduling Algorithms

Enhanced-FCFS uses the FCFS heuristic on the taxiway and the queue balancing heuristic described in Section III.A. These heuristics are integrated with a dynamic programming module for scheduling departures as well as a crossing heuristic to insert crossings into departure sequences. For departure runway scheduling, the dynamic programming algorithm described by Rathinam et al<sup>12</sup> is implemented for the departure scheduling

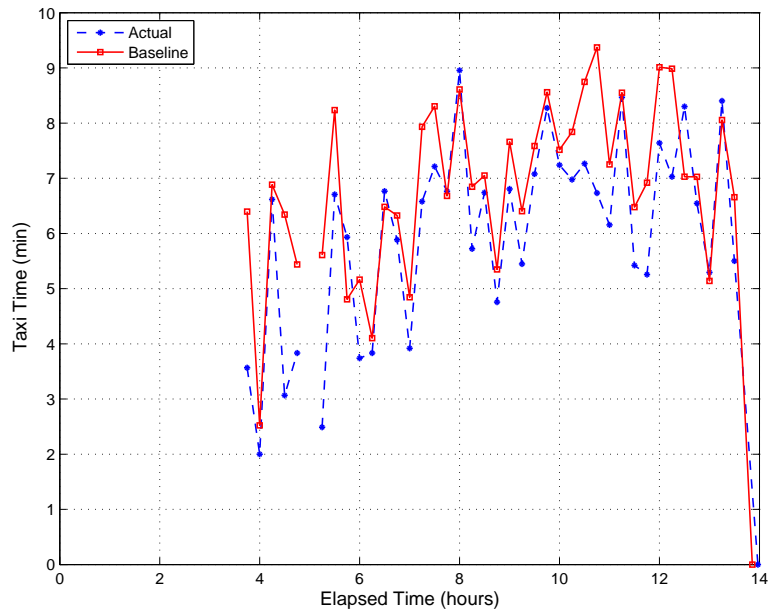


Figure 3. Comparison of average taxi times every 15 minutes for arrivals from the SOS<sup>2</sup> Baseline-FCFS simulation with the actual data at DFW airport for June 11th.

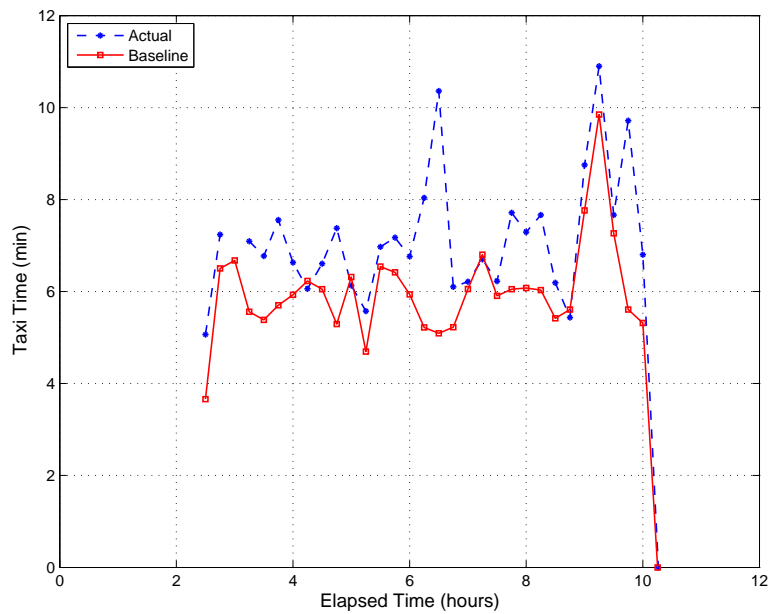


Figure 4. Comparison of average taxi times every 15 minutes for departures from the SOS<sup>2</sup> Baseline-FCFS simulation with the actual data at DFW airport for May 24th.

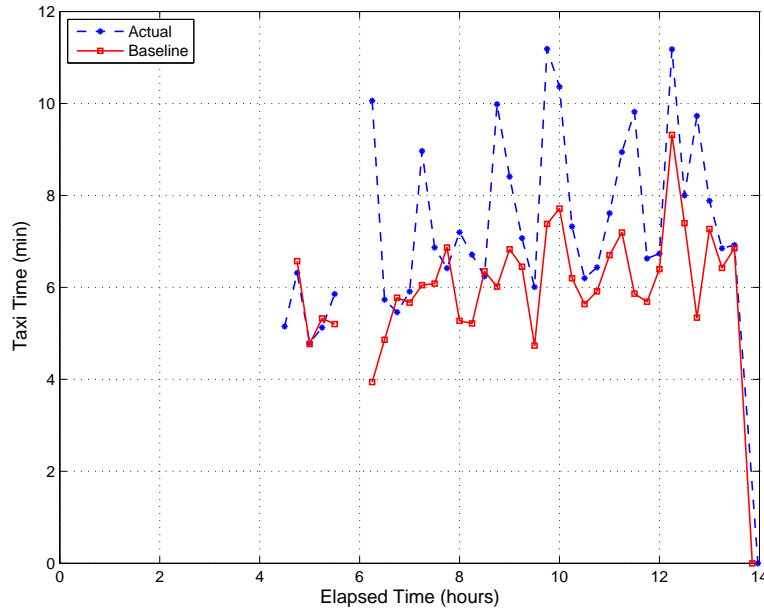


Figure 5. Comparison of average taxi times every 15 minutes for departures from the SOS<sup>2</sup> Baseline-FCFS simulation with the actual data at DFW airport for June 11th.

problem with chain-like queues. The unimpeded time to the departure runway for aircraft in the departure queue and on the taxiway are input to the DP module as the release times. Assignment to the departure queue is provided from the queue balancing heuristic.

Since the DP module currently does not have the capability to schedule crossings, a crossing heuristic has been implemented to improve delay. For a specified departure runway, if all departure aircraft estimated times of arrival are more than the estimated time an aircraft takes to cross the runway, then a set of parallel crossings are inserted at the beginning of the runway control point sequence. However, if there are departure aircraft that are waiting or will arrive at the departure runway within the estimated crossing time then decisions branch based upon a few possible states. If the number of aircraft in any crossing queue is three or more, which is close to the capacity of the crossing queues at DFW, then a set of parallel crossings is inserted at the beginning of the sequence for the runway control point. If the count of aircraft in the individual crossing queues are not in danger of violating the capacity of the queues, then the total number of aircraft in the departure queue and the total number of aircraft in the crossings queues are counted. If the number of aircraft in the crossing queues exceeds the number of aircraft in the departure queue, then a set of parallel crossings is inserted at the beginning of the sequence for the runway control point. Otherwise, if no crossing queue is in jeopardy of violating capacity and the total number of aircraft in the crossing queues is not more than the number of aircraft in the departure queue, then a set of parallel crossings is inserted into the runway control point sequence at an estimated time when the total number of aircraft in the crossing queues exceeds the number of aircraft in the departure queue.

## IV.B. Results

In Fig. 6 there is a large spike in the average taxi delays for arrivals around the ninth hour for Enhanced-FCFS. This spike in the data is a result of the behavior of the crossing heuristic implemented in Enhanced-FCFS. Although this departure push increased arrival delays around the ninth hour of the scenario, a noticeable improvement over Baseline-FCFS can be seen in Fig. 8 in the delay for departure aircraft around the ninth hour. The resulting decrease in delay of departure aircraft over multiple data points surrounding the ninth hour in Fig. 8 while only increasing arrival delays over two data points near the ninth hour in Fig. 6 indicates the crossing heuristic has some ability to balance when a departure push is necessary since arrival delay was temporarily increased to provide a better over system delay over that time period.

Examining the average delay shown in Figs. 6-9, Enhanced-FCFS follows a similar profile to Baseline-FCFS, but does not appear to improve significantly upon the delay from Baseline-FCFS although Enhanced-FCFS consistently has a lower delay than Baseline-FCFS with the exception of a few data points. However,

when calculating the saving in delay over the simulation, which was a time window of 7.7 hours for May 24th and 10 hours for June 11th, in both scenarios Enhanced-FCFS improved substantially over Baseline-FCFS on the total delay. For the May 24th scenario Enhanced-FCFS has a 3.37% improvement in total taxi times and a 33.1% improvement in total delay over Baseline-FCFS; for the June 11th scenario Enhanced-FCFS has a 2.34% improvement in total taxi times and a 20.0% improvement in total delay over Baseline-FCFS. The improvements were most notable in the delay of departure aircraft. For the May 24th and June 11th scenarios respectively, Enhanced-FCFS has a 27.7% improvement and 13.2% improvement for arrivals and a 42.0% improvement and 30.6% improvement in delay. The savings in delay and taxi times when using Enhanced-FCFS amounted to 13.8 seconds per aircraft for May 24th and 14.0 seconds per aircraft for June 11th, which is a total system savings of 9475 seconds and 6678 seconds respectively.

Although these savings may seem very large, Baseline-FCFS uses pure FCFS behavior for the departure runways, most notably arrival crossings are not queued for departure pushes. When the departure runway is under constant pressure from arrivals and departures, pure FCFS behavior alternates 1-to-1 between giving a departure clearance for a departure aircraft and a crossing clearance for a set of arriving aircraft crossing the runway in parallel. Implementing a heuristic similar to the crossing heuristic implemented in Enhanced-FCFS or a heuristic similar to what traffic controllers use for current day operations could improve the Baseline-FCFS model and prevent overestimating savings as may be the case in the comparison between Enhanced-FCFS and Baseline-FCFS. Although such a heuristic implemented in Baseline-FCFS would reduce the improvement Enhanced-FCFS achieved over Baseline-FCFS, a significant savings in delay is still expected over with Enhanced-FCFS over Baseline-FCFS.

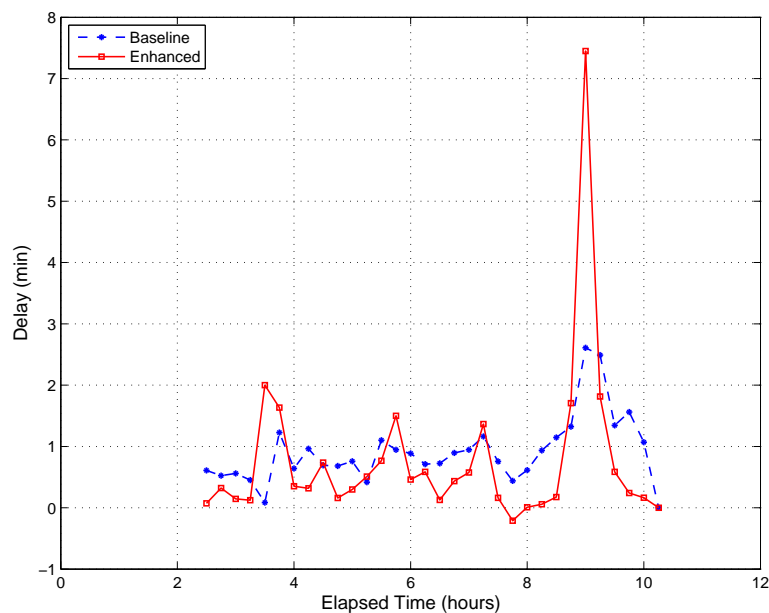


Figure 6. Comparison of average taxi delay every 15 minutes for arrivals from the SOS<sup>2</sup> Enhanced-FCFS and Baseline-FCFS for the May 24th scenario.

## V. Conclusions

Enhanced-FCFS produced a significantly larger than expected savings over Baseline-FCFS, which may be due to the pure FCFS behavior used for scheduling arrivals crossing the departure runway. If the changes to better align the Baseline-FCFS data with actual data suggested in Section III.B are made and there is no significant difference in average delay or taxi times between Baseline-FCFS and actual data, then it can be assumed that the savings seen in delay with Enhanced-FCFS over Baseline-FCFS can be construed as potential savings to current day operation at DFW airport. If there is significantly a better average delay with the actual data over Baseline-FCFS once the changes suggested in Section III.B are made, then the case is likely that FCFS used as a crossing heuristic is not adequate to provide a model for current day operations. It is not certain whether a Baseline-FCFS implementation with FCFS as a crossings heuristic is



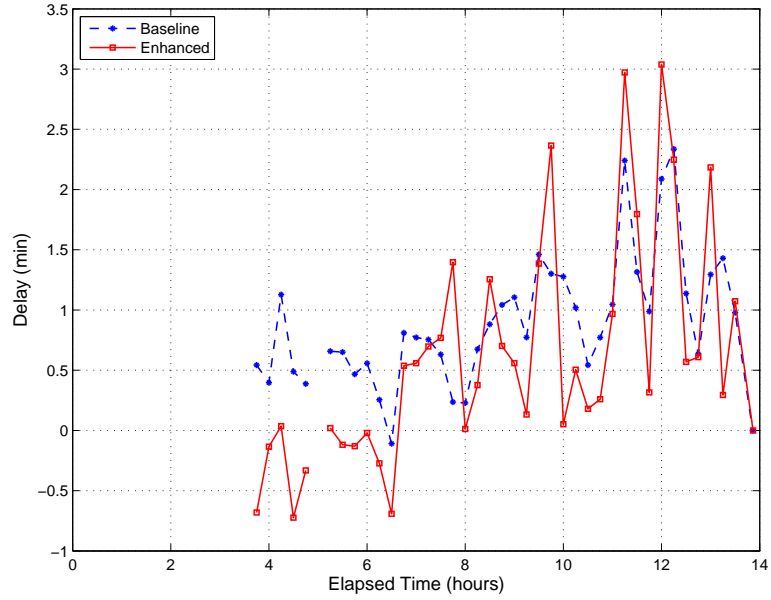


Figure 7. Comparison of average taxi delay every 15 minutes for arrivals from the SOS<sup>2</sup> Enhanced-FCFS and Baseline-FCFS for the June 11th scenario.

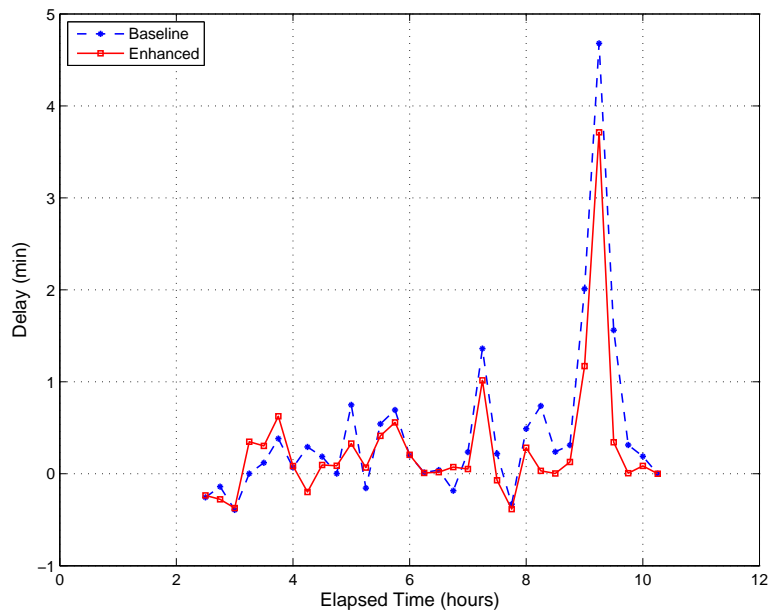


Figure 8. Comparison of average taxi delay every 15 minutes for departures from the SOS<sup>2</sup> Enhanced-FCFS and Baseline-FCFS for the May 24th scenario.

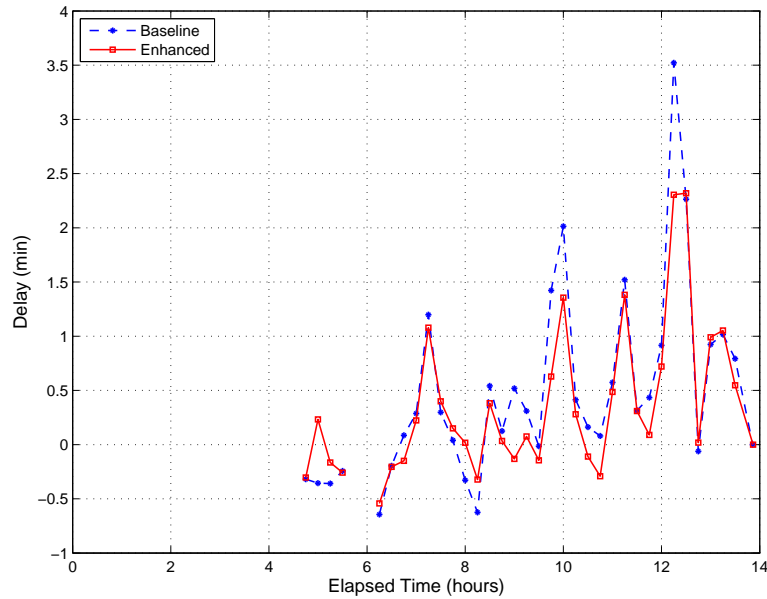


Figure 9. Comparison of average taxi delay every 15 minutes for departures from the SOS<sup>2</sup> Enhanced-FCFS and Baseline-FCFS for the June 11th scenario.

a valid benchmark of current day operations since the dynamic programming algorithm used in Enhanced-FCFS was able to produce savings in delay up to 30.8% over FCFS during Monte-Carlo simulations<sup>12</sup> for a 3-queue problem, which is the same model for queue structure used for DFW airport, but was only able to produce a savings in delay of 6.7% averaged over all the simulations. Further refinement and testing of SOS<sup>2</sup> and Baseline-FCFS will help remove this uncertainty.

The current dynamics module implemented in SOS<sup>2</sup> does not encompass the differing behavior between arrival and departure aircraft. However, the differences between Baseline-FCFS and the actual data for arrival aircraft as well as the differences between Baseline-FCFS and departure aircraft were consistent. These consistent differences implies implementing a new aircraft dynamics module, gathering actual data on metering for departure fixes, and allowing arrivals to use the departure runways during times of low activity at DFW airport will further refine Baseline-FCFS and should give a better alignment of Baseline-FCFS output with the actual data.

Additional features are being implemented into SOS<sup>2</sup> for more extensive capabilities such as departure fix metering constraints, uncertainty, and additional airport configurations. Each of the main components of the simulation, such as departure scheduling and taxi scheduling, are being integrated with various optimization algorithms to test how various combinations of schedulers perform under varying conditions. Scenarios and metrics are currently being developed for these simulations. For initial testing, simulations will use a DFW airport model starting at a 1x traffic density and incrementally increase to a 2x traffic density, where 1x and 2x are 1 times and 2 times the current average traffic density at DFW airport respectively. Scenarios will include SOS<sup>2</sup> simulations using Baseline-FCFS compared to DFW airport data gathered from SODAA, similar to that described in Section III.

Preliminary testing of Baseline-FCFS provided an initial comparison of the SOS<sup>2</sup> simulation with actual data and indicate the need for further refinement of the aircraft dynamics module to encompass the difference in speeds between arrival and departure aircraft. The integration of a dynamic programming module with FCFS and arrival crossing heuristics demonstrate the ability to use SOS<sup>2</sup> for rapid prototyping and testing combinations of surface schedulers. Additional algorithms will be implemented in varying combinations including generalized Dynamic Programming (DP) for departure scheduling, a MILP formulation solved with CPLEX for taxi scheduling, a MILP formulation solved with CPLEX for departure scheduling and queue balancing, taxi scheduling heuristics, and departure scheduling heuristics. The varying combinations of these algorithms will be compared against the Baseline-FCFS simulations as well as actual data from DFW airport through SODAA where applicable.

## References

- <sup>1</sup>NASA, "*NASA & The Next Generation Air Transportation System (NextGen)*", NASA, 2008.
- <sup>2</sup>ATAC, "*Experts in Aviation Analysis*," [[www.atac.com/Products\\_Research-a.html](http://www.atac.com/Products_Research-a.html). Accessed 1/28/09.]
- <sup>3</sup>The Preston Group, "*TAAM Plus, Fast-Time Gate-to-Gate Simulation, Reference Manual*," The Preston Group Pty Ltd., Richmond, Victoria 3121, Australia, 1999.
- <sup>4</sup>Rathinam S., Montoya J., and Jung Y., "*An Optimization Model for Reducing Aircraft Taxi Times at the Dallas Fort Worth International Airport*," 26th International Congress of the Aeronautical Sciences, Anchorage, Alaska, September 2008.
- <sup>5</sup>Brinton M., Wood B., and Engelland S., "*Microscopic Analysis of Airport Surface Sequencing*," 26th International Congress of the Aeronautical Sciences, Anchorage, Alaska, September 2008.
- <sup>6</sup>ILOG, Inc., "*ILOG CPLEX 10.0 - Users Manual*," 2006.
- <sup>7</sup>Free Software Foundation, Inc., "*GLPK - GNU Project - Free Software Foundation (FSF)*," [<http://www.gnu.org/software/glpk/#introduction>. Accessed 2/2/09.]
- <sup>8</sup>SODAA, "*Surface Operations Data Analysis and Adaptation, Software Package, v.1.8*," Mosaic ATM, Leesburg, VA 2008.
- <sup>9</sup>Google, "*Google Earth Mapping Service*," [Accessed May 2008.]
- <sup>10</sup>SGI, "*OpenGL Reference Manual*," [<http://www.opengl.org>. Accessed 2/3/09.]
- <sup>11</sup>Ascher, Uri M., Petzold, Linda Ruth. "*Computer methods for ordinary differential equations and differential-algebraic equations*," 1998. SIAM.
- <sup>12</sup>Rathinam S., Wood Z., Sridhar B., Jung Y., "*A Generalized Dynamic Programming Approach for a Departure Scheduling Problem*," American Institute of Aeronautics and Astronautics - GNC, Chicago, Illinois, August 2009.