

# Benefits Assessment of a Surface Traffic Management Concept at a Capacity-Constrained Airport

Katy J. Griffin,<sup>1</sup> Aditya Saraf,<sup>2</sup> and Peter Yu<sup>3</sup>  
*Saab Sensis Corporation, Campbell, California, 95008*

Steven R. Stroiney<sup>4</sup> and Benjamin S. Levy<sup>5</sup>  
*Saab Sensis Corporation, E. Syracuse, New York, 13057*

Gustaf Solveling<sup>6</sup> and John-Paul Clarke<sup>7</sup>  
*Georgia Institute of Technology, Atlanta, GA, 30332*

and

Robert D. Windhorst<sup>8</sup>  
*NASA Ames Research Center, Moffett Field, California, 94035*

Inefficient surface traffic management may lead to congested taxiways, long departure queues, and excess delay in the air transportation system. To address this problem, NASA researchers have developed optimization algorithms and a concept of operations for an airport surface traffic management tool called the Spot and Runway Departure Advisor (SARDA). Past SARDA research efforts have been focused on the Dallas/Fort Worth International airport. This paper describes the development of SARDA-like schedulers for managing the traffic at an operationally dissimilar airport—Charlotte Douglas International airport, and presents the results of a fast-time simulation-based benefits assessment. Fast-time simulations were conducted to test the benefits of optimized scheduling over a baseline model of current-day operations. In the fast-time simulations, it was observed that optimization schedulers reduced movement area delays by up to 3.1 minutes per departure on average, as compared to the baseline simulation. The movement area delay savings translated to shorter movement area taxi-out times and an average reduction in fuel burn and emissions of approximately 24% per departure. The overall trend observed in the total delay (gate delay + ramp delay + movement area delay) comparison indicated the optimization schedulers were not able to reduce total delay, and runway throughput comparisons suggested the optimization schedulers had little to no effect on throughput.

## I. Introduction

AIRPORTS are key capacity bottlenecks within the national air transportation network. As traffic demand grows, delays experienced by major airports are expected to increase significantly if there are no major increases in airport capacity. Beyond the Federal Aviation Administration's (FAA) planned capacity increases,<sup>1</sup> sustaining the increased demand requires significant improvements in surface traffic management efficiency. Reducing the environmental impact of airport surface traffic is an additional challenge.

<sup>1</sup> Research Engineer, Advanced Development – STC, 1700 Dell Avenue, AIAA Member.

<sup>2</sup> Senior Research Engineer, Advanced Development – STC, 1700 Dell Avenue, AIAA Senior Member.

<sup>3</sup> Research Engineer, Advanced Development – STC, 1700 Dell Avenue, AIAA Member.

<sup>4</sup> Research Engineer, Advanced Development – NTC, 85 Collamer Crossings.

<sup>5</sup> Operations Research Group Manager, Advanced Development – NTC, 85 Collamer Crossings, AIAA Member.

<sup>6</sup> Graduate Research Asst., School of Industrial and Systems Engineering, 765 Ferst Drive, AIAA Student Member.

<sup>7</sup> Associate Professor, School of Aerospace Engineering, 270 Ferst Drive, AIAA Associate Fellow.

<sup>8</sup> Aerospace Engineer, Aerospace High Density Operations Branch, MS210-15, AIAA Senior Member.

NASA researchers are developing surface optimization algorithms and a concept of operations for an airport surface management tool called the Spot and Runway Departure Advisor (SARDA).<sup>2</sup> SARDA integrates two decoupled schedulers—the Spot Release Planner (SRP) and Runway Scheduler (RS).<sup>2-5</sup> Until recently, NASA’s SARDA algorithm designs and experiments have been focused solely on Dallas/Fort Worth International airport (DFW). Some aspects of the SARDA schedulers are designed to take advantage of certain optimization opportunities or degrees of freedom that are specific to DFW airport. For example, runway scheduling at DFW is made easier by large areas of pavement or “pads” near the runway threshold, which are used to build up to three separate departure queues. Thus, departure sequencing decisions can be delayed until the very end of taxi-out.

Research is currently underway to assess whether or not SARDA can be effective at other capacity-constrained airports where dissimilar airport geometries and operational characteristics exist. The benefits assessment described in this paper is focused on Charlotte/Douglas International airport (CLT). CLT has the potential to benefit from improved surface management since it experiences significant departure delays. The runway geometry at CLT is different than the geometry at DFW, and CLT features various types of dependent runway operations (e.g., mixed-use). Lastly, CLT is dominated by a single carrier (US Airways), which runs a single ramp tower for the entire passenger terminal.

For the CLT assessment, fast-time simulations were conducted to quantify the potential benefits (e.g., increased taxi efficiency, reduced environmental impacts) of surface optimization schedulers. NASA’s Surface Operations Simulator and Scheduler (SOSS) simulation platform was used to conduct the fast-time simulations, and an airport model of CLT surface traffic operations was developed. Terminal-gates, ramp and movement area taxiways, spot-locations (for transition between the non-movement and movement areas), and runways were modeled using a node-link graph. ASDE-X data were analyzed to identify and build realistic movement area taxi routes. Site visits to the CLT air traffic control tower and US Airways ramp tower were conducted to gather additional details on operational constraints, and this information was used to enhance the airport model. Optimization schedulers were developed and integrated with SOSS for the fast-time simulation-based benefits assessment. The schedulers were adapted from the original SARDA design to be compatible with the operational characteristics of CLT.

The remainder of this paper is organized as follows. Section II describes the SARDA concept and provides details on the original SRP and RS schedulers designed for DFW. Section III describes how the SARDA schedulers were adapted for CLT. Section IV provides details of the fast-time simulation experiment setup and modeling approach that was used to assess the performance of SARDA at CLT. Section V highlights the results of the performance evaluation. Finally, a discussion of future work and conclusions are included in Sections VI and VII.

## II. SARDA Concept

This section provides an overview of the SARDA concept and describes how the SRP and RS scheduling algorithms were implemented in previous DFW-focused experiments. These descriptions establish a foundation for subsequent discussions in Section III regarding the scheduler adaptations that were necessary for CLT, and in general, for airports other than DFW.

### A. Overview

SARDA is a decision support tool (DST) that is intended to help the airport tower controllers and airline ramp controllers better manage airport surface traffic.<sup>2</sup> SARDA divides the airport surface traffic management problem into two sub-problems—(i) determining the optimum spot release times to minimize taxi delays in the movement area, and (ii) determining the optimum runway departure and crossing sequence and times for maximizing runway throughput. SARDA integrates the solutions to these two sub-problems and generates controller advisories. The first sub-problem described above is handled by the SRP, and the second is handled by the RS. The SRP and RS schedulers solve the optimization problem over a finite look-ahead time called the planning horizon. Both schedulers utilize deterministic predictions of aircraft arrival times at key locations on the airport surface (e.g., runway entrance).

The DFW SARDA schedulers described in the following sections are based on the 2010 SARDA concept. Since 2010, the SARDA concept has evolved and different variations of the SRP and RS schedulers have been explored. For example, the schedulers used in a recent human-in-the-loop (HITL) simulation, conducted at NASA Ames Research Center in May 2012, optimally scheduled gate push back in addition to spot release and runway sequencing.\*

\* [http://www.aviationsystemsdivision.arc.nasa.gov/news/highlights/af\\_highlights\\_20120615.shtml#hilit2](http://www.aviationsystemsdivision.arc.nasa.gov/news/highlights/af_highlights_20120615.shtml#hilit2)

## B. DFW SARDA Spot Release Planner

The SARDA SRP is based on a departure aircraft metering concept,<sup>5</sup> which introduces a two-stage algorithm for metering aircraft at gates or spots. The DFW SRP used in the 2010 SARDA experiments generated an optimal spot release schedule for departure aircraft to maximize runway usage while minimizing taxi delay in the movement area.<sup>2</sup> The DFW SRP computations were independent of the DFW RS computations. Since the SRP acted with no knowledge of the optimal runway sequence desired by the RS, it had to estimate an optimized runway sequence. Consequently, the DFW SRP used a two-stage process for computing the optimal schedule. The first stage involved solving a mathematical optimization problem, which computed an optimal departure schedule at the runway for a set of departing flights, with the objective of maximizing runway throughput. The second stage involved computing the optimal time to release aircraft from their assigned spots, in order to minimize taxi delay in the movement area, and meet the optimal departure schedule from the first stage. The DFW SRP computed spot release times by simply subtracting the estimated unimpeded taxi time from the scheduled take-off time. This simplified spot release time computation was sufficient for DFW experiments because on the east side of DFW airport, where the SARDA research experiments were focused, the taxi route structure is relatively simple and consists of three standard departure taxi routes (K-EF, K-EG, L-EH) with almost equal taxiway lengths. However, for other airports where the taxiway/ramp geometry is more complicated, a scheduler can exercise more control over the runway arrival sequence by enabling the control of release times at critical merge nodes within the ramp area or the movement area.

Furthermore, the DFW SRP used in the 2010 SARDA experiments exercised control over spot release times only. It did not control release times at upstream locations, such as gates. A scheduler that controls gate release times can further enhance fuel efficiency on the airport surface by shifting taxi and runway queue delay to the gate, where the aircraft can wait with their engines turned off. In contrast to the SRP implementation described in Ref. 5, more recent work described in Ref. 6 explores a scheduler that does issue gate release times in order to minimize taxi delay.

## C. DFW SARDA Runway Scheduler

The DFW RS is motivated by operations specific to DFW airport. In contrast to most other airports, DFW has multi-lane taxiways and wide queuing areas adjacent to the departure runways. The queuing area enables local departure runway controllers to change the sequence of operations after aircraft have passed the spot and up until the point when aircraft enter the runway.

The main purpose of the DFW RS presented in previous SARDA research experiments is to determine a sequence of operations that optimizes a delay or throughput objective.<sup>2,4, 7-9</sup> Three different objectives have been evaluated (i) minimize system delay, (ii) maximize throughput, and (iii) minimize maximum delay.<sup>4</sup> The DFW RS assigned each aircraft to one of the queues in the runway queuing area, as well as provided optimum sequencing and timing decisions for departures and crossings on the runway system. The mathematical optimization problem presented in Ref. 4 was formulated as a Mixed Integer Linear Programming (MILP) problem. The MILP problem considered the set of all departure flights taking off from runway 17R at DFW, and the set of all arrival flights crossing runway 17R. The decision variables in the MILP problem were reported as follows:

- 1) Binary variable  $y_{f,p}$  takes on a value of 1 if flight  $f$  is assigned to position  $p$ , zero otherwise. Position in this context refers to the position of the flight within the runway usage sequence.
- 2) Binary variable  $x_{f,q}$  takes on a value 1 if flight  $f$  is assigned to queue  $q$ , zero otherwise. This variable is necessary in the DFW RS formulation because there are three queue lanes in the queuing area adjacent to runway 17R.
- 3) Variable  $t_p$  is the runway usage time of an aircraft assigned to position  $p$ . Position in this context refers to the position of the flight within the runway usage sequence.

Key input parameters in the MILP problem included the earliest runway usage time, allowable runway usage time window, and minimum separation requirements (e.g., wake vortex separation, miles-in-trail restrictions). Lastly, the core constraints of the MILP problem included a constraint to enforce minimum separation requirements between each pair of aircraft, a constraint to enforce a first-in-first-out (FIFO) structure in each departure queue and at each crossing location, a constraint to ensure that each aircraft occupies a single position in the runway usage sequence, and a constraint to ensure that each departure aircraft occupies a single departure queue.

There are a few potential drawbacks with applying the above formulation to airports other than DFW. The most critical one is that it is difficult to model a general precedence constraint, enforcing that flight  $a$  must use the runway before flight  $b$ . Such a constraint is necessary when  $a$  is physically ahead of  $b$  on a single-lane taxiway, and both are headed to the same departure runway. In addition, the above formulation does not generalize easily to more complicated configurations, e.g., when a departing flight needs to cross its departure runway prior to departing.

### III. CLT Adaptation of SARDA Schedulers

Given the differences of airport geometry, operational constraints, and operating procedures between DFW and CLT, the following scheduler modifications were implemented:

- 1) Modified the scheduler architecture and order of execution to enable better synchronization between the RS and SRP schedules.
- 2) Upgraded the RS to do the following:
  - a. Handle mixed-use, intersecting, and virtually intersecting departure and arrival runway operations.
  - b. Enable adherence to only physically limiting sequence constraints (blocking constraints) rather than departure queue assignment-based sequence constraints. This was necessary because at CLT there is not enough space at the departure end of the runway to form multiple parallel departure queues.
  - c. Incorporate a freeze horizon parameter.
- 3) Developed a new SRP algorithm which has two main features:
  - a. Control sequencing at critical merge nodes on the taxiway surface (including ramp area) for upstream enforcement of RS-prescribed departure sequences.
  - b. Address uncertainty by providing earlier-than-required spot-release, critical intersection node-release, and gate-release times to maintain a pre-specified departure queue depth.

#### A. Modified Scheduler Architecture

In the 2010 SARDA experiments, the RS and SRP acted independently without the explicit knowledge of the optimum sequence/timing decisions taken by the other. In the CLT adaptation, the RS and SRP schedulers were combined into an integrated scheduler and were executed in series, as illustrated in Fig. 1.

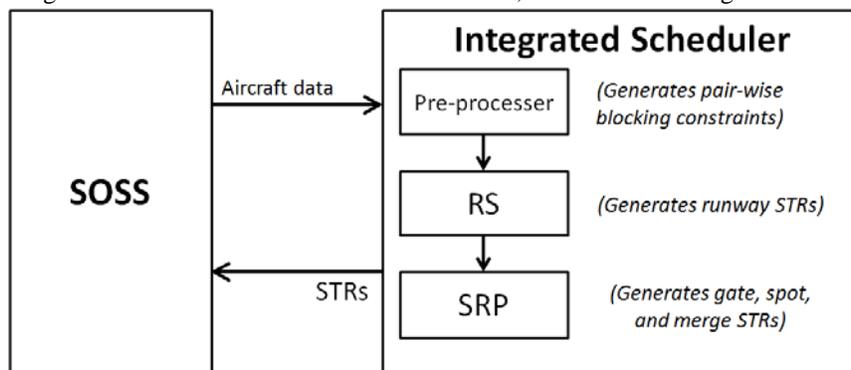


Figure 1. System Design.

Within the integrated scheduler, the RS acts as the “master” scheduler and executes before the SRP. It determines the runway sequence for departure and crossing operations by optimizing the sequence to minimize system-wide delay. The Pre-processor computes blocking constraints based on the physical location of flights on the airport surface, and the RS adheres to these constraints. The SRP acts as the “enabler.” It accepts the RS runway sequence as input, and schedules times of release (STRs) for upstream control-points (e.g., gates, spots, and merge nodes) to ensure that flights arrive at the runway in the right sequence. The SOSS simulation platform translates the STRs issued by the scheduler into a runway usage sequence, and enforces that sequence at the runway. SOSS also holds flights at gates, spots, and merge nodes as necessary to adhere to STRs issued by the scheduler for those locations.

As shown in Fig. 1, SOSS sends aircraft data to the scheduler, and the scheduler sends STRs back to SOSS. The aircraft data sent to the scheduler includes the following information: departure pushback time, arrival landing time, estimated arrival times (ETAs) to the first and last route position (i.e., gate and runway ETAs), routes, etc. In this experiment, the integrated scheduler did not use the ETAs provided by SOSS. Instead, it used its own internal ETA prediction component to estimate arrival times to every position in the taxi route.

#### B. CLT Adaptation of SARDA Runway Scheduler

Adapting the RS for CLT required a different formulation of the runway scheduling problem than what had been done originally for DFW. To discuss the adaptations in detail, it is first necessary to present the CLT RS MILP formulation.

*Parameters:*

- $F$  is the set of all flights.  
 $F'$  is the set of fixed flights that were scheduled before the flights in  $F$ .  
 $a_f$  is the earliest runway usage time for flight  $f$ .  
 $\Delta_f$  is the time window for flight  $f$ .  
 $d_{i,j}$  is the minimum separation between flight  $j \in F$  and  $i \in F$ , if flight  $j$  follows flight  $i$ .  
 $d_{f',f}$  is the minimum separation between flight  $f' \in F'$  and  $f \in F$ , if flight  $f'$  is followed by flight  $f$ .  
 $t'_{f'}$  is the fixed time of runway usage for flight  $f'$ .

*Decision Variables:*

- $t_f$  is the time of runway usage for flight  $f$ .  
 $t^L$  is the time of latest arrival/departure time.  
 $z_{i,j}$  is a binary variable which takes on a value of 1 if flight  $i \in F$  is scheduled before flight  $j \in F$ , zero otherwise.

*Formulation:*

$$\text{Minimize System Delay: minimize } \sum_{f \in F} t_f - a_f \quad (1)$$

such that

$$t^L \geq t_f \quad \forall f \in F \quad (2)$$

$$t_j \geq t_i + d_{i,j} - Mz_{j,i} \quad \forall i, j \in F, i \neq j \quad (3)$$

$$z_{i,j} + z_{j,i} = 1 \quad \forall i, j \in F, j > i \quad (4)$$

$$t_f \geq a_f \quad \forall f \in F \quad (5)$$

$$t_f \leq a_f + \Delta_f \quad \forall f \in F \quad (6)$$

$$t_f \geq t'_{f'} + d_{f',f} \quad \forall f' \in F', f \in F \quad (7)$$

$$z_{i,j} \in \{0,1\} \quad \forall i, j \in F \times F, i \neq j \quad (8)$$

$$t^L \geq 0 \quad (9)$$

The set of flights in  $F'$  are fixed flights that have been scheduled in the past (i.e., they have already departed, arrived, or crossed the runway) or that are “frozen.” The “frozen” flights are those that fall within a user-defined freeze horizon. For further details on the freeze horizon parameter, see Section III, subsection B3.

The pair-wise separation requirements  $d_{i,j}$  and  $d_{f',f}$  represent wake vortex and miles-in-trail separation requirements. These separation requirements depend on the airport configuration and aviation authority regulations, and do not change during the course of the simulation. Therefore, these rules are included as input parameters to the model. For further details on separation requirements, see Section IV, subsection B4 and B5.

Equation (1) is the system delay objective function. Equation (2) ensures the runway usage time of the latest flight is greater than or equal to the runway usage time of all other flights. Equation (3) enforces wake vortex and miles-in-trail separation requirements, and physical blocking constraints. Physical blocking constraints may arise because of various reasons that are discussed in the next section. Equation (4) ensures that blocking constraints will not lead to circular dependencies between flight pairs. Equations (5) and (6) ensure the runway usage time for a flight falls within the allowable time window. Equation (7) ensures the runway usage time for  $f$  does not violate separation requirements between  $f$  and every other flight that precedes it. Lastly, Eq. (8) and (9) are provided to ensure the solution is feasible. The constraint structure of this formulation is simpler than the DFW RS used in the 2010 SARDA experiments, and requires fewer constraints. The downside of this formulation is that “big-M” type constraints must be introduced (see Eq. 3) to ensure the blocking constraints are not violated. This may result in a weak LP-relaxation of the model, and in turn, may lead to long execution times for large instances.

Using this formulation, time windows for aircraft and various types of operations on single or multiple dependent runways can be modeled, including the case where aircraft cross the departure runway prior to departing. Some of the advantages of this formulation are as follows:

- 1) Equation structure supports modeling more complex airport configurations.
- 2) Physical blocking constraints can easily be included.
- 3) Freeze horizon parameter can be incorporated.
- 4) If desired, a general cost function for aircraft delay can be incorporated.

There are some potential disadvantages of this formulation. The most critical disadvantage is a potentially weak LP-relaxation that limits the ability to solve large instances. On the other hand, this disadvantage can be alleviated by using a rolling horizon implementation, which has the potential to significantly reduce the problem size and computation time.

### 1. RS upgrades for handling multiple types of runway interactions

The DFW RS formulation, which handled a single departure runway, had to be extended to handle different types of runway interactions, such as departures, crossings, and arrivals on the same or on coupled runways. To simplify the description of the solution for handling multiple interaction types, consider runway 18C at CLT shown in Fig. 2. This runway is used for both arrival (*A*) and departure (*D*) operations. In addition, arrivals on 18R cross (*C*) 18C on their way to the terminal. Also, arrivals on Runway 23 must be considered, due to the interaction (*I*) between the two runways. The four types of operations are denoted in Fig. 2 as *A*, *D*, *C*, and *I*.



**Figure 2. CLT 18C Runway Interaction Types.**

The RS adapted for CLT includes all operation types in the model. If  $f$  is an arrival operation (*A* or *I*),  $a_f$  is fixed. Departures and crossings can be scheduled within the time interval defined in Eq. (6).

Prior to solving the optimization problem, there are a number of steps performed by the CLT RS to initialize the set of flights and accurately calculate the input parameters. The steps are performed for each runway (or set of dependent runways) at the airport. These steps are summarized below.

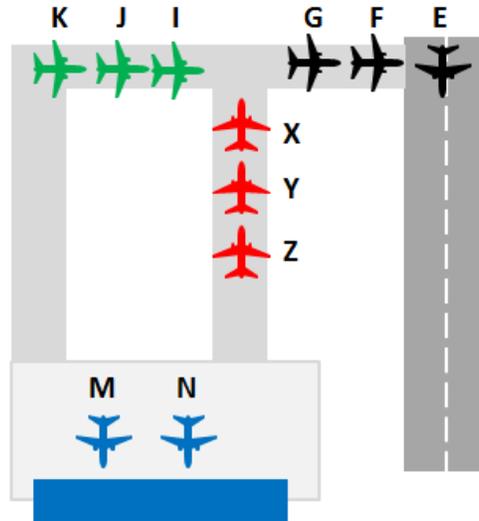
- 1) Among all active flights, find the ones that are operating on 18C or are interacting with operations on 18C, and categorize them as *A*, *D*, *C*, and *I*, respectively. Let  $F$  be the set of all flights included for optimization on 18C.
- 2) Calculate  $a_f$  for each  $f$  in  $F$ .
- 3) Set the initial schedule based on earlier optimized flights and the current state of the runway. The initial schedule dictates when the first  $f$  in  $F$  can be scheduled.
- 4) Find blocking constraints computed by the pre-processor (if any) and include these in the model.
- 5) Solve the optimization model.
- 6) Update the scheduled release times for the departure and crossing nodes relevant to operations on 18C.

In this experiment, a rolling planning horizon paradigm is used, thus the runway scheduling problem is continuously re-optimized or re-solved during successive iterations. Therefore, flights can be included in the model more than once.

### 2. RS upgrades for enabling adherence to physical blocking constraints

There were three departure queues used in the 2010 DFW SARDA experiments. Since CLT has only one departure queue per departure runway, the FIFO constraint present in the original DFW RS formulation had to be eliminated. Otherwise, the RS would have always enforced the FIFO sequence based on earliest runway arrival time estimates for all departures, allowing no opportunity to choose a different, optimized sequence.

The RS implementation at CLT needed to consider physical limitations that would prevent position shifts within the sequence—i.e., blocking constraints. For example, if flight  $F$  is taxiing to the departure runway, a following flight  $G$  cannot be scheduled before  $F$ , if  $F$  is physically ahead of  $G$ . This example is illustrated in Fig. 3.



**Figure 3. Example of a Blocking Constraint between Two Aircraft on the Same Route.**

For the CLT adaptation, a RS pre-processor algorithm was implemented to dynamically define pair-wise blocking constraints. For example in Fig. 3, the pre-processor would determine that F must follow E, G must follow F, I must follow G, X must follow G, etc. The pair-wise blocking constraints defined by the pre-processor are RS input parameters. These constraints ensure the RS will produce a feasible solution. The pre-processor is not fully general. In particular, it assumes only a single route option for each flight. Returning to Fig. 3, the flights at the gate (M and N) could proceed by either of two routes, joining aircraft I, J, and K (green group), or aircraft X, Y, and Z (red group). By considering the multiple route options, the constraints imposed on M and N are not simple pair-wise constraints, but rather are route-dependent. The pre-processor may be extended to this more general case; but in the current implementation, only a single route option is considered for each flight.

### 3. RS upgrades for incorporating a freeze horizon parameter

The RS implementation at CLT included a user-defined freeze horizon parameter. A freeze horizon of zero allows the scheduler to update scheduled release times up until the last possible moment – e.g., right before takeoff. A non-zero freeze horizon prevents the scheduler from making these last minute changes. If the scheduler was deployed in a fielded DST and the freeze horizon was non-zero, the controller would see fewer scheduler advisory updates as the aircraft got closer to the runway.

In the CLT implementation, the RS acts as the “master” scheduler and executes before the SRP (see Fig. 1). Since the SRP takes the RS scheduled runway release times as input, once the runway times are fixed (i.e., “frozen”) the SRP-prescribed gate, spot, and merge node release times also become fixed. If the freeze horizon is set to zero the gate release time will never be fixed, which means the scheduler could potentially delay gate pushback at each successive optimization cycle. Therefore, setting the freeze horizon parameter to a non-zero value prevents any particular aircraft from being overly delayed in subsequent optimization cycles.

### C. CLT Adaptation of SARDA Spot Release Planner

In the CLT adaptation, the SRP was designed to run in series with the RS. This design concept was implemented so the SRP could generate gate, spot, and merge point release times that supported the optimal runway sequence set by the RS. The CLT SRP issues STRs at gate, spot, and critical merge points in order to (i) set up the desired runway sequence, (ii) manage the flow of aircraft into the movement area, (iii) minimize ramp congestion and engines-on taxi time by delaying flights at the gate, and (iv) maintain a pre-specified departure queue depth by providing “earlier-than-just-in-time” release times, so that the runway system is not “starved.”

The first goal of setting up the desired runway sequence is achieved by issuing STRs at merge points to ensure that flights merge according to the take-off order prescribed by the RS. A merge point STR can be issued in the ramp or taxiway area, and is necessary any time two departure flights headed to the same runway have merging taxi paths. Within the fast-time simulation environment, ground controller behavior is implicitly modeled by issuing merge point STRs. The merge point STRs would likely not be displayed in a fielded DST; instead, the tool might show the desired sequence and let the ramp and ground controllers issue instructions to achieve that sequence.

The second goal of managing the flow of aircraft into the movement area is achieved by issuing STRs at the spot. The third goal of minimizing ramp congestion is achieved by issuing STRs at the gate. Gate STRs are issued for every departure flight which has not yet left the gate.

The fourth goal of maintaining a pre-specified departure queue depth is achieved by modifying the scheduled release times so that flights are released early. By releasing flights early, a runway queue is allowed to build, which in turn ensures the runway system will not be starved. As mentioned previously, the DFW SRP computed release times by simply subtracting the estimated unimpeded taxi time from the scheduled take-off time. In this adaptation, the release times are computed by subtracting the estimated unimpeded taxi time plus a user-defined buffer time from the scheduled take-off time. A reasonable buffer time can be computed by multiplying the desired queue-depth with an average time separation between successive departures. The average inter-departure time can be estimated from surveillance data.

## IV. Experiment Setup and Modeling Approach

### A. Simulation Platform

The potential benefits of the surface optimization concept adapted for CLT were assessed using NASA's Surface Operations Simulator and Scheduler (SOSS) simulation platform. SOSS is a fast-time simulation platform used to simulate airport surface operations and to support rapid prototyping of surface scheduling algorithms.<sup>10</sup> SOSS was designed to be used in conjunction with external scheduling components (e.g., runway scheduler, spot release planner). When integrated with external schedulers, SOSS moves aircraft on the surface according to the recommended schedule, and monitors separation violations and scheduling conformance.

Through the SOSS graphical user interface (GUI), various scheduler parameters can be configured. The scheduler call interval and forecast window are two important parameters. The scheduler call interval determines how often a scheduler is called, and the forecast window determines how many flights are sent to the scheduler. For example, if the forecast window is set at 10 minutes, then SOSS sends out information on all aircraft that are currently on the surface, and aircraft that are predicted to be on the surface in the next 10-minute forecast window. In this experiment, the scheduler call interval and forecast window parameters were set at 60 seconds and 15 minutes, respectively.

SOSS models aircraft movement on the airport surface based on a linear kinematic model. Within the model, speed parameters are dependent on aircraft type and airport region (e.g., ramp, movement, and departure queuing areas). In addition, SOSS models controller/pilot decisions for inter-flight separation management, as well as runway sequence control and runway separation enforcement based on FAA separation standards. The aircraft dynamics model and tactical flight separation model used in SOSS are described in more detail in Ref. 6.

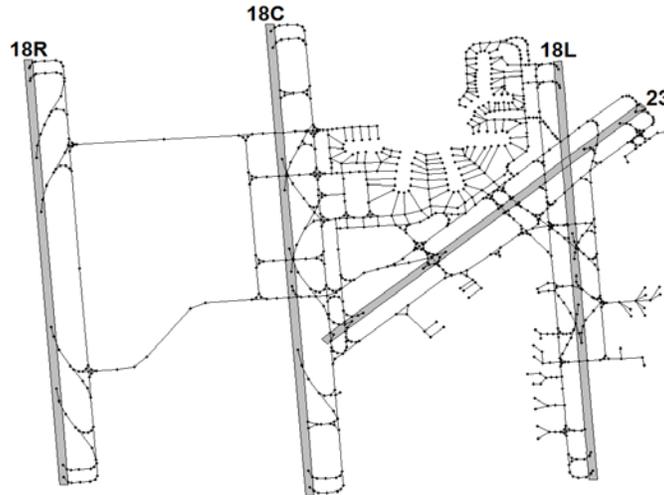
SOSS takes airport adaptation data, aircraft database information, and traffic scenario files as input, and outputs data describing simulated aircraft movement (e.g., position, time, speed), aircraft schedule conformance, scheduler performance, and detected safety violations. Additional details on SOSS inputs are presented in the following sections.

### B. CLT Airport Adaptation Data

SOSS airport adaptation data sets consist of (i) node-link airport surface graph, (ii) arrival and departure taxi routes, (iii) commonly used runway configurations, (iv) commonly used arrival and departure fixes, and (v) runway separation requirements. An airport adaptation data set for CLT was developed using operational information (e.g., taxi route, runway and spot usage, average taxi speeds) derived from recorded ASDE-X surveillance data and ASPM reported metrics. Site visits to the CLT air traffic control tower and US Airways ramp tower were conducted to gather additional details on airport operations, and this information was used to enhance the adaptation data.

#### 1. Node-Link Graph

SOSS uses a node-link graph to represent the geometry of the airport surface as a network of nodes connected by links. The CLT node-link graph is shown in Fig. 4. The graph had 982 nodes and 1142 links.



**Figure 4. CLT Node-Link Graph.**

In the SOSS simulation, issuing STRs at critical nodes is the primary control mechanism for external scheduling components. STRs can be issued for any node; however, in this experiment STRs are only issued for gate, spot, runway, and critical merge nodes. Critical merge nodes can be located in the ramp or movement area.

## 2. *Predefined Taxi Routes*

SOSS uses a list of predefined arrival and departure taxi routes, which are defined by specifying a sequence of nodes between gates and runways. An arrival route starts with an arrival node at the runway and ends with a gate node. A departure route starts with a gate node and ends with a departure node at the runway. The CLT predefined arrival and departure taxi routes were designed to reflect the common routes used at the airport. Common routes were identified by analyzing surface surveillance data over a period of one month (July, 2010).

## 3. *Runway Configuration*

The most commonly used runway configuration for CLT was determined by analyzing Aviation System Performance Metrics (ASPM) data from January 6<sup>th</sup>, 2010 to May 31<sup>st</sup>, 2011 (Ref. 11). The ASPM database reports meteorological conditions and runway configurations in 15-minute time bins. The most commonly used runway configuration for CLT under Visual Meteorological Conditions (VMC) conditions is arrivals on runways 18C, 18R, and 23, and departures on 18C and 18L. This runway configuration is known as the South Flow Configuration. During a site visit to the airport, CLT controllers corroborated that during the day the South Flow Configuration is the most used configuration and produces the highest airport throughput. Based on this feedback, the South Flow Configuration was selected for the airport model.

## 4. *Fixes*

The arrival and departure fixes for the CLT airport model were identified using multiple online resources such as AirNav.com and FAA OIS. The CLT arrival fixes included: CTF, MAJIC, SHINE, UNARM. The CLT departure fixes included: ANDYS, BUCKL, DEBIE, GANTS, JACAL, LILLS, MERIL, SPA, SUG, ZAVER. The miles-in-trail separation requirements discussed previously in Section III, subsection B, are defined in terms of temporal separation requirements at the departure fix. Two consecutive flights headed for the same departure fix were required to be separated by 80 seconds. A temporal spacing of 80 seconds at the departure fix translates into a five miles-in-trail restriction into the en route airspace.<sup>6</sup>

## 5. *Runway Separation Requirements*

Multiple runway separation matrices were defined for the CLT airport model using a previously established methodology.<sup>12</sup> The runway separation matrices derived using this methodology were validated using surface surveillance data. Each separation matrix defines the minimum required temporal separation between two successive operations on the same runway or a pair of interacting runways, and defines different separations for different leader-follower aircraft weight classes. Four aircraft weight classes—Heavy (H), Large (L), Boeing 757 (B757), and Small (S) are included in the matrices as per the standard FAA definitions.<sup>13</sup> The runway separation matrices were categorized according to runway usage (i.e., dedicated departure or arrival runways versus shared or mixed-use

runways), and the type and order of operations using the runway. For the CLT airport model, the following distinct categories were defined:

- 1) Same Runway, Departure after Departure: applies when two consecutive departures take off from the same runway. At CLT this can occur on 18C or 18L.
- 2) Shared Runway, Departure after Arrival: applies when a runway is shared by both arrival and departure operations, and the departure is taking off after the arrival lands. At CLT this can occur on 18C.
- 3) Shared Runway, Departure before Arrival: applies when a runway is shared by both arrival and departure operations, and the departure wants to take off before the arrival lands. At CLT this can occur on 18C.
- 4) Intersecting Runways, Departure after Arrival: applies when the departure and arrival runways intersect each other (either physically or virtually), and the departure is taking off after the arrival lands. At CLT this can occur if departures use 18C or 18L, and arrivals use 23. The 18C/23 intersection is a virtual intersection, and the 18L/23 intersection is a physical intersection.
- 5) Intersecting Runways, Departure before Arrival: applies when the departure and arrival runways intersect each other (either physically or virtually), and the departure wants to take off before the arrival lands. At CLT this can occur if departures use 18C or 18L, and arrivals use 23.

In the current implementation of SOSS, the arrival schedule is fixed, and there is no mechanism to monitor or control separation between consecutive arrivals. Therefore, no separation requirements were defined for consecutive arrivals.

The runway separation matrices for arrival and departure operations used in the CLT airport model are defined in the Appendix.

### C. CLT Traffic Scenario

SOSS uses a traffic scenario input data file, which contains relevant flight information for the set of flights to be simulated. The traffic scenario data for departures defines the scheduled gate pushback time (i.e., aircraft activation time), assigned departure runway, departure fix, and gate. Similar flight information is included in the traffic scenario data for arrivals; however, instead of gate pushback time, the aircraft activation time is equal to the time the arriving aircraft is at final runway descent.

The CLT traffic scenario was generated based on recorded ASDE-X, ASQP, and ASDI data from July 7<sup>th</sup>, 2010 from 12PM to 6PM local time, and included 221 departures and 223 arrivals. This day and time were chosen as the baseline scenario because (i) the South Flow Configuration was used predominantly throughout the day, and (ii) this time frame captured multiple arrival and departure peaks. Figure 5 shows a plot of the hourly operation count for July 7<sup>th</sup>, 2010, and shows that there were three somewhat overlapping arrival and departure peaks in the selected scenario from 12PM to 6PM.

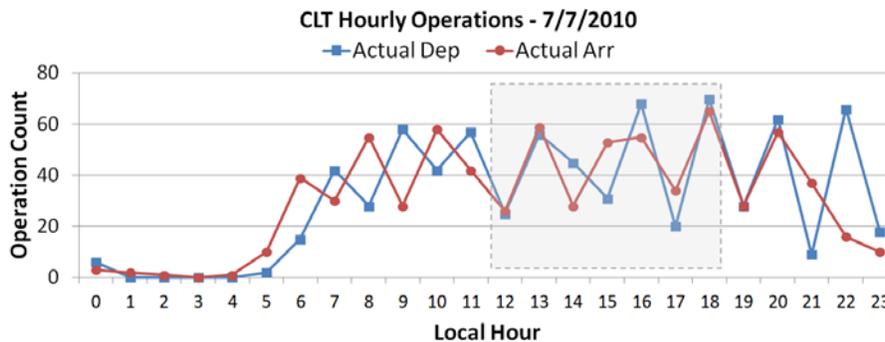


Figure 5. CLT Hourly Operation Count on 7/7/2010 (Ref. 11).

## V. Results

The benefits provided by the optimization schedulers were assessed by comparing performance metrics from a baseline simulation against simulations with the optimization schedulers. In the next section, the baseline simulation setup is described. In the subsequent section, the performance metrics from baseline and optimization scheduler simulations are compared.

### A. Baseline Simulation Setup

A pseudo-FCFS RS scheduler, designed to emulate current-day controller behavior, was used in the baseline simulation. Unlike the optimization scheduler, the baseline scheduler did not provide a globally optimal solution to

the runway sequencing problem. The runway operations sequence was computed using MILP optimization with FIFO constraints applied separately to the aircraft in the departure queue and the aircraft crossing the runway. The baseline scheduler also enforced blocking constraints, i.e., if flight A was physically ahead of flight B, then the scheduler enforced a “flight B after A” constraint. The blocking constraints were given priority over the FIFO constraints to ensure the solution was physically feasible.

In the baseline simulation, the SRP scheduling component was turned off. Since the SRP was turned off, aircraft were released from the gate on time and did not incur any gate or spot holding.

The baseline simulation was validated by comparing simulated taxi-out time data with taxi-out time data that were estimated from surveillance data. For this comparison, only the movement area taxi-out time was measured, due to the fact that ramp area surveillance data were not available. The same six-hour period that was used to generate the SOSS traffic scenario was used to validate the model. The results of the validation effort revealed the simulated taxi-out time and estimated taxi-out time datasets were similar in terms of central tendency and spread. However, there were more flights in the estimated taxi-out time dataset with longer taxi-out times. This indicates the simulated taxi-out times tend to be shorter than what is observed in the real operations. In addition to comparing taxi time, the simulated runway throughput was compared against runway throughput that was estimated from surveillance data. Over the six-hour period, there were small differences between the simulated throughput and the throughput that was estimated from surveillance data. Overall, the validation results suggested that the baseline simulation was not a perfect representation of current day CLT operations. However, it was sufficiently close for the purpose of evaluating the optimization schedulers within a fast-time simulation environment.

With the exception of the scheduler formulation, all other settings were consistent between the baseline and optimization scheduler simulations. Specifically, the same airport model and traffic scenario was used in both cases, and the scheduler call interval and forecast window parameters were set at 60 seconds and 15 minutes, respectively. Also, in both cases runway sequence control and runway separation was enforced by SOSS.

## **B. Baseline vs. Optimization Scheduler Performance Comparison**

The benefits provided by the optimization schedulers were assessed by comparing performance metrics from a baseline simulation against several simulations with the optimization schedulers each using different parameter settings. In the optimization scheduler simulations, the RS freeze horizon and SRP queue-depth/buffer time parameters were varied. The RS freeze horizon parameter was set at 0 and 60 seconds, and the SRP queue-depth/buffer time parameter was set at 0 and 120 seconds. The following key metrics were used in the performance evaluation: (1) delay, (2) taxi time, (3) throughput, and (4) fuel consumption and engine emissions. The results of the performance evaluation are presented in the following sections.

### *1. Delay*

A well-functioning surface optimization system reduces total departure delay (due to greater runway throughput) and reallocates whatever delay remains from the departure queue to the gate (enabling fuel savings, among other benefits). For departures, the delay metric consists of three parts: gate, ramp area, and movement area delay. Gate delay is computed as the difference between nominal scheduled gate departure time and actual gate release time. Ramp area delay is computed as the difference between unimpeded ramp taxi time and actual ramp taxi time. Movement area delay is computed as the difference between unimpeded movement area taxi time and actual movement area taxi time.

Figure 6 compares delay metrics for departures in the baseline and optimization scheduler simulations. This plot shows total delay across all 221 departure flights. Table 1 shows the average delays per departure for the baseline and optimization scheduler simulations. In the first optimization scheduler simulation, both the RS freeze horizon and SRP queue-depth/buffer time parameters were set to 0 seconds (FH\_0, BT\_0). In the second optimization scheduler simulation, the RS freeze horizon parameter was set to 60 seconds, and the SRP queue-depth/buffer time parameter was set to 0 seconds (FH\_60, BT\_0). In the third optimization scheduler simulation, the RS freeze horizon parameter was set to 0 seconds, and the SRP queue-depth/buffer time parameter was set to 120 seconds (FH\_0, BT\_120).

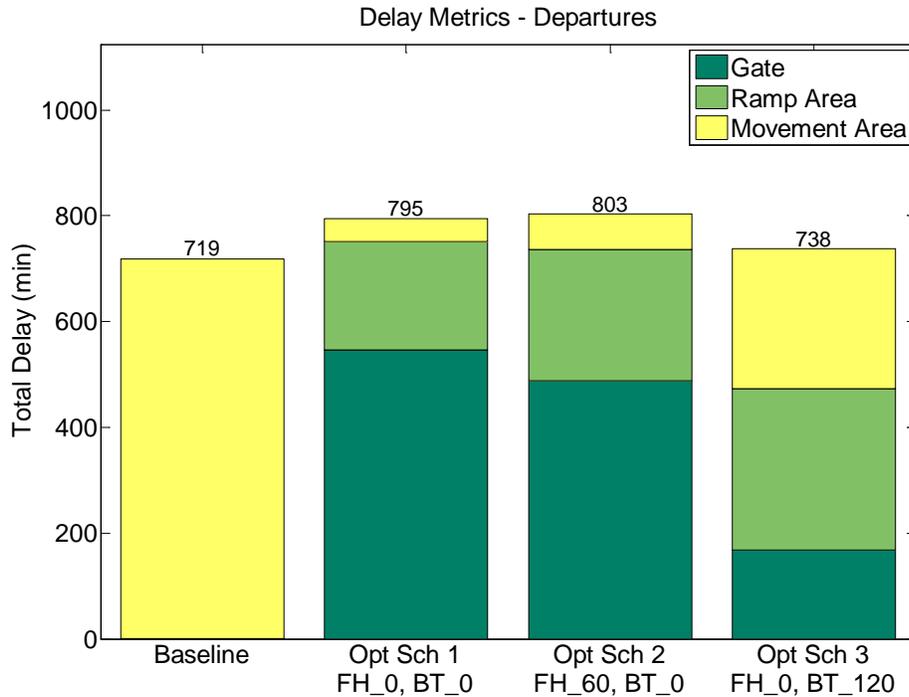


Figure 6. Performance Results – Total Delay Comparison.

Table 1. Average Delays per Departure (Units = Minutes).

	Baseline	Opt Scheduler 1 Freeze Horizon = 0 sec Buffer Time = 0 sec	Opt Scheduler 2 Freeze Horizon = 60 sec Buffer Time = 0 sec	Opt Scheduler 3 Freeze Horizon = 0 sec Buffer Time = 120 sec
Gate Delay	0.0	2.5	2.2	0.8
Ramp Area Delay	0.0	0.9	1.1	1.4
Movement Area Delay	3.2	0.2	0.3	1.2
Total Delay	3.2	3.6	3.6	3.3

In the baseline simulation, total delay for all departures was 719 minutes. Across the three optimization scheduler simulations, total delay ranged from 738 to 803 minutes. These results indicate the optimization schedulers were not able to reduce total delay. The optimization schedulers were able to reduce movement area delay but the delay savings were negated by the increase in gate and ramp area delay.

The only type of delay incurred in the baseline simulation was movement area delay. There was no gate or ramp delay in the baseline simulation because the SRP, which controls gate and spot holding and merge node metering, was not active. The absence of gate delay in the baseline case is due to the fact that aircraft were released from the gate on time. The absence of ramp delay in the baseline case is due to (i) the fact that aircraft were not held at the spot, and (ii) aircraft experienced unimpeded movement through the ramp area. The SOSS tactical flight separation model had to be disabled in the ramp area due to an unresolved issue with the SOSS conflict detection and resolution logic. The ramp area separation model was disabled in both the baseline and optimization scheduler simulations, which means all ramp delay introduced in the optimization scheduler simulations is due to spot holding and merge node metering.

The first optimization scheduler simulation, in which both the RS freeze horizon parameter and SRP queue-depth/buffer time parameter were set to 0 seconds, showed a large gate delay and very small movement area delay. This parameter setting represents a scheduler that attempts to release flights from the gates “just-in-time” to reach the runway at the RS-prescribed runway departure time. However, departure flights can still incur taxi-out delay due to congestion along the taxi route. If a flight is delayed it will not reach the runway on time to meet the RS-prescribed runway departure time, and could potentially receive a double-penalty delay (i.e., delayed once at the gate to achieve a target runway time, and delayed again at the runway because of missing the prescribed takeoff time).

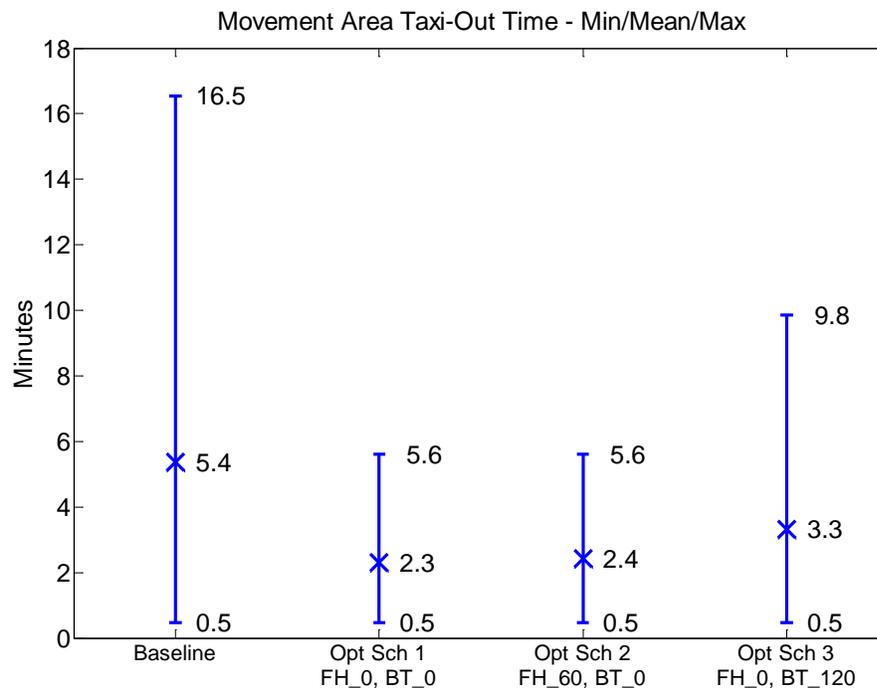
The second optimization scheduler simulation, in which the RS freeze horizon parameter was set at 60 seconds, showed slightly smaller gate delays than the first optimization scheduler, and slightly higher ramp and movement area delays. In this simulation, RS-prescribed runway departure times were fixed once they fell within a 60 second freeze horizon window. Since the SRP takes the RS-prescribed runway departure times as input, once the runway times were fixed the SRP-prescribed gate, spot, and merge node release times also became fixed. This gave the schedulers less freedom to optimize release times up until the last possible moment. However, a non-zero freeze horizon makes it easier for controllers to manage operations because they have fixed target operation times/sequences to work with. Results from the second optimization scheduler simulation quantify the trade-off between providing stable prescribed sequences to the controllers versus performing optimization up until the last possible moment.

The third optimization scheduler simulation, in which the SRP queue-depth/buffer time parameter was set at 120 seconds, produced the best (minimum) total delay results across the three optimization scheduler simulations. In this simulation, movement area delays were higher (still not as high as in the baseline simulation) and gate delays were lower. This parameter setting represents a scheduler that releases flights from the gates as much as 120 seconds earlier than their “just-in-time” release times. This gives the flights an opportunity to absorb any delays that are required to maintain safe separation with other flights along their taxi paths, and still reach the runway at or before their RS-prescribed runway departure times.

The results presented above provided some insights into how the RS freeze horizon parameter and the SRP queue-depth/buffer time parameter affect system delay, but further tests will need to be conducted to determine what the optimal settings are for these parameters. In addition, other scheduler parameters such as scheduler call interval and forecast window will need to be evaluated, as well.

## 2. Taxi Time

Figure 7 compares movement area taxi time for departures in the baseline and optimization scheduler simulations. This plot shows the minimum, maximum, and average movement area taxi-out times for each simulation. Movement area taxi-out time is computed as the difference between spot release time and runway release time. Since runway release marks the end of movement area taxiing, any time spent in the runway departure queue is included in this metric.



**Figure 7. Performance Results – Taxi Time Comparison.**

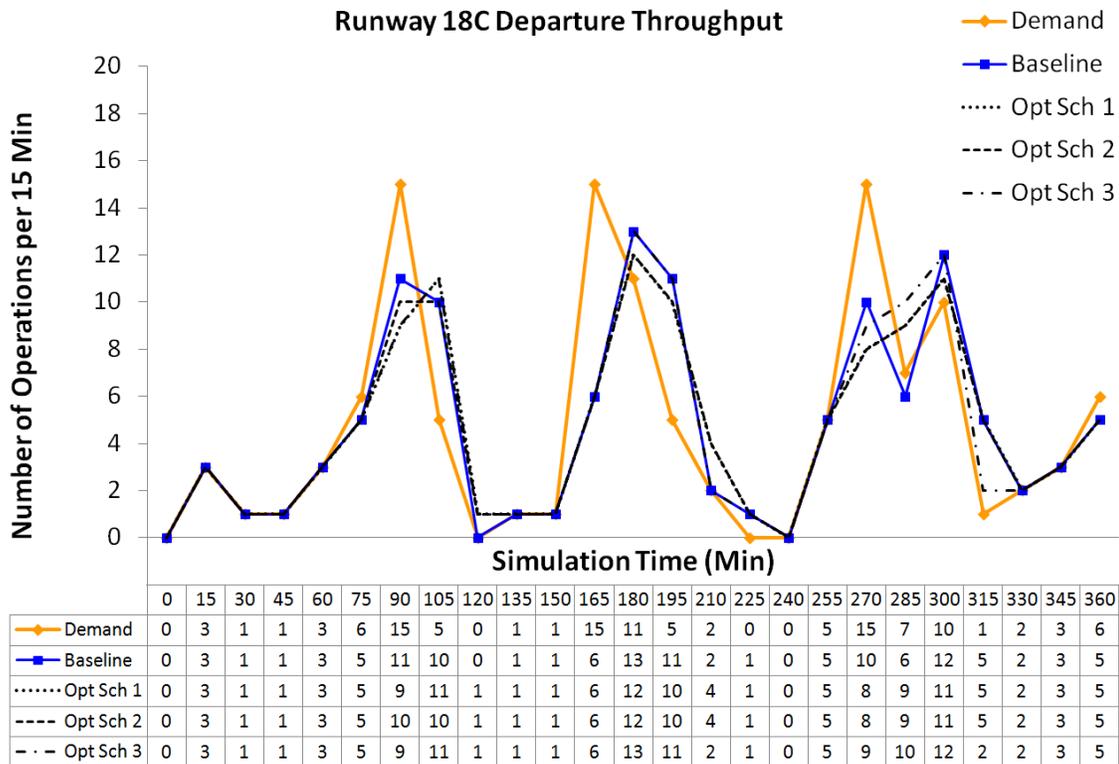
These results indicate the optimization schedulers reduced average movement area taxi-out time, and reduced the maximum movement area taxi-out time. The optimal results in terms of taxi time are observed in the first

optimization scheduler simulation, in which both the RS freeze horizon and SRP queue-depth/buffer time parameters were set to 0 seconds. When the RS freeze horizon parameter was set to 60 seconds, average movement area taxi-out time increased slightly, but the minimum and maximum values stayed the same. When the SRP queue-depth/buffer time parameter was set at 120 seconds, the average and maximum movement area taxi-out time values increased.

In summary, looking at Figs. 6 and 7 together, it is apparent the SRP queue-depth/buffer time parameter controls the scheduler's strategy. Maintaining a small non-zero queue depth leads to less total delay but achieves this at the cost of higher movement area delays/taxi-times (and smaller gate delays). Whereas a queue depth of zero leads to less movement area delays/taxi-times at the cost of higher overall delays (because of large gate delays).

### 3. Throughput

The throughput metric is computed for all operations on a per runway basis, and is computed by counting the total number of operations (e.g., departures, arrivals, and crossings) in each 15-minute time period. The throughput metric is also computed for unique operations (e.g., departures only) on a per runway basis. Figure 8 compares 18C departure throughput in the baseline and optimization scheduler simulations. In this plot the "demand" series represents the throughput that would occur if the schedule were followed exactly and there were no separation constraints at the runway or along the taxiways.



**Figure 8. Performance Results – Throughput Comparison.**

There appears to be some differences in 18C departure throughput between the baseline and optimization scheduler simulations. In the first demand peak around simulation time 90, the highest throughput rate occurred in the baseline simulation and the rates in the optimization scheduler simulations fell short by one to two aircraft. In the second demand peak around simulation time 180, the throughput rate in the third optimization scheduler simulation matched the baseline. In the third demand peak around simulation time 270, the highest throughput rate occurred in the baseline simulation and again the rates in the optimization scheduler simulations fell short by one to two aircraft. In the fourth and final demand peak around simulation time 300, the throughput rate in the third optimization scheduler simulation matched the baseline. The overall trend in these results indicates the optimization schedulers had little to no effect on throughput when compared against the baseline simulation.

It is expected that the scheduler would have provided benefits in terms of increased throughput had the arrivals on 18C not been fixed. Since 18C is a mixed use runway and the arrivals are fixed, the scheduler has to optimize the

departure operations around the arrivals. If there was some flexibility in modifying the arrival schedule there could have been more opportunities for optimization.

#### 4. Fuel Consumption and Engine Emissions

A reduction in taxi time translates directly to a reduction in fuel burn and emissions. Using aircraft type information and a database of fuel burn and emissions rates for each aircraft type,<sup>†</sup> the fuel burn and emissions for each simulated flight was calculated assuming single-engine taxiing. To the extent that aircraft taxi with multiple engines, the savings would be even higher. Table 2 shows the fuel burn and emissions statistics for departures in the baseline simulation and the first optimization scheduler simulation in which both the RS freeze horizon and SRP queue-depth/buffer time parameters were set to 0 seconds. The comparison is made for this particular optimization scheduler simulation because of the trend observed in the taxi time comparison. The fuel costs shown in Table 2 are calculated based on a \$0.90 per kilogram (kg) fuel price.<sup>‡</sup>

**Table 2. Fuel Burn and Emissions Statistics for Departures.**

Parameter	Baseline		Opt Scheduler 1		Savings	
	Per Dep	Total	Per Dep	Total	Per Dep	Percentage
Fuel (kg)	44.9	9920	34.2	7556	10.7	24%
Fuel Cost (\$)	40.40	8928	30.77	6801	9.63	24%
Carbon Dioxide (CO <sub>2</sub> ) (kg)	143.6	31744	109.4	24180	34.2	24%
Hydrocarbon (HC) (kg)	0.15	33.5	0.12	25.5	0.03	20%
Carbon Monoxide (CO) (kg)	1.46	323.7	1.10	243.8	0.36	25%
Nitrogen oxide (NO <sub>x</sub> ) (kg)	0.17	37.5	0.13	28.6	0.04	24%

According to these results, the fuel burn savings per departure was 10.7 kg or approximately 24%, and corresponding reductions in CO<sub>2</sub>, HC, CO, and NO<sub>x</sub> were observed. These results indicate the scheduling components are providing benefits in terms of reduced environmental impact.

To estimate annual fuel savings, the per-departure savings were multiplied by the number of daily departures and by the number of days per year. The daily departure count used in this estimate was 725 departures per day.<sup>§</sup> The annual fuel and fuel cost savings were estimated to be 2.8 million kg and \$2.6 million, respectively.

## VI. Future Work

The overarching objective of the current research work described herein is to assess whether or not SARDA can be effective at busy capacity-constrained airports where a variety of airport geometries and operational characteristics, dissimilar to DFW, exist. CLT was the first airport assessed. Work is currently underway to assess the benefits of SARDA-like surface management tools at two other airports, namely, JFK and BOS.

In addition to adapting SARDA to other airports and assessing the potential benefits, an extension of this research is to assess what airport-specific enhancements (pertaining to both the SOSS simulation platform as well as the schedulers) are necessary to extract the maximum benefit of surface optimization at each individual airport. For example, at CLT, addressing the interaction between controllers' conflict detection and resolution actions and the optimization scheduler-prescribed actions is a significant issue. This is a significant issue because of the high degree of complexity in the CLT ramp area where interactions and potential conflicts between aircraft are occurring frequently. For JFK, handling weather impacts (e.g., departure route closures due to convective weather) or NAS-wide traffic flow management (TFM) related constraints (e.g., flight-specific Expect Departure Clearance Times (EDCTs) issued during ground delay programs) are important considerations. For BOS, given its complex high-interaction runway system, enhancements to the runway scheduler could be addressed as a high priority next-step. Besides these airport-specific enhancements, surface optimization researchers can address some general research topics applicable to multiple airports—one being enhancing the robustness of the schedulers to deal with multiple sources of uncertainty, including those introduced by pushback time prediction errors or disruptive taxiing events (e.g., taxiway blockages). Furthermore, a concept of operations in which SARDA optimization tools are used collaboratively to schedule gate pushback times for flights from multiple airlines can be explored.

<sup>†</sup> <http://easa.europa.eu/environment/edb/aircraft-engine-emissions.php>

<sup>‡</sup> [http://www.iata.org/whatwedo/economics/fuel\\_monitor/Pages/index.aspx](http://www.iata.org/whatwedo/economics/fuel_monitor/Pages/index.aspx)

<sup>§</sup> <http://charmeck.org/city/charlotte/Airport/AboutCLT/Pages/Fast%20Facts.aspx>

## VII. Conclusion

This paper presents results from a fast-time simulation-based benefits assessment of NASA’s SARDA concept applied to CLT. Results from this assessment indicated the optimization schedulers reduced movement area delays by up to 3.1 minutes per departure on average over the six-hour simulation period, as compared to the baseline simulation (designed to emulate current-day CLT operations). The movement area delay savings translated to shorter movement area taxi-out times, and a reduction in the maximum movement area taxi-out time from 16.5 minutes in the baseline simulation down to 5.6 minutes in the optimization scheduler simulation. The movement area taxi-out time savings translated to an average reduction in fuel burn and emissions of approximately 24% per departure. The overall trend observed in the total delay (gate delay + ramp delay + movement area delay) comparison suggested the optimization schedulers were not able to reduce total delay; however, the redistribution of delay from the movement area to the gate yielded significant savings in terms of taxi-out time, fuel burn, and emissions. Lastly, the overall trend observed in the throughput comparison indicated the optimization schedulers had little to no effect on throughput.

The optimization scheduler simulations were run using different scheduler parameter settings. Adjusting the RS freeze horizon parameter quantified the trade-off between providing stable prescribed sequences to the controllers versus performing optimization up until the last possible moment. Adjusting the SRP queue-depth/buffer time parameter demonstrated how this parameter controls the scheduler’s optimization strategy. Further tests will need to be conducted to determine what the optimal settings are for these and other scheduler parameters (e.g., call interval, forecast window).

## Appendix

The runway separation matrices for arrival and departure operations used in the CLT airport model are defined in the following five tables, and the values are presented in seconds.

**Table 3. Same Runway, Departure after Departure on 18C or 18L.**

		Leading Aircraft (Dep)			
		B757	Heavy	Large	Small
Following Aircraft (Dep)	B757	90	90	60	60
	Heavy	90	90	60	60
	Large	120	120	60	60
	Small	120	120	60	60

**Table 4. Shared Runway, Departure after Arrival on 18C.**

		Leading Aircraft (Arr)			
		B757	Heavy	Large	Small
Following Aircraft (Dep)	B757	60	70	60	50
	Heavy	60	70	60	50
	Large	60	70	60	50
	Small	60	70	60	50

**Table 5. Shared Runway, Departure before Arrival on 18C.**

		Leading Aircraft (Dep)			
		B757	Heavy	Large	Small
Following Aircraft (Arr)	B757	28	28	28	28
	Heavy	24	24	24	24
	Large	28	28	28	28
	Small	40	40	40	40

**Table 6. Intersecting Runways, Departure after Arrival on 18C and 23 or 18L and 23.**

		Leading Aircraft (Arr)			
		B757	Heavy	Large	Small
Following Aircraft (Dep)	B757	5	4	5	7
	Heavy	5	4	5	7
	Large	5	4	5	7
	Small	5	4	5	7

**Table 7. Intersecting Runways, Departure before Arrival on 18C and 23 or 18L and 23.**

		Leading Aircraft (Dep)			
		B757	Heavy	Large	Small
Following Aircraft (Arr)	B757	40	40	40	40
	Heavy	40	40	40	40
	Large	40	40	40	40
	Small	40	40	40	40

### References

- <sup>1</sup>Federal Aviation Administration, “Operational Evolution Partnership Implementation Plan,” 2007
- <sup>2</sup>Jung, Y., T. Hoang, J. Montoya, G. Gupta, W. Malik, and L. Tobias, “A Concept and Implementation of Optimized Operations of Airport Surface Traffic”, Proceedings of the AIAA 10<sup>th</sup> Aviation Technology, Integration, and Operations Conference, AIAA, Fort Worth, TX, 2010.
- <sup>3</sup>Gupta, G., W. Malik, and Y. Jung, “A Mixed Integer Linear Program for Airport Departure Scheduling,” Proceedings of the AIAA 9<sup>th</sup> Aviation Technology, Integration, and Operations Conference, Hilton Head, SC, 2009.
- <sup>4</sup>Gupta, G., W. Malik, and Y. Jung., 2010. “Incorporating Active Runway Crossings in Airport Departure Scheduling,” Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, Canada, 2010.
- <sup>5</sup>Malik, W., G. Gupta, and Y. Jung., 2010. “Managing Departure Aircraft for Efficient Airport Surface Operations,” Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, Canada, 2010.
- <sup>6</sup>Windhorst, B., “Towards a Fast-time Simulation Analysis of Benefits of the Spot and Runway Departure Advisor,” AIAA Guidance, Navigation, and Control Conference, AIAA 2012-4975, Minneapolis MN, 2012.
- <sup>7</sup>Rathinam, S., Z. Wood, B. Sridhar, and Y. Jung. “A Generalized Dynamic Programming Approach for a Departure Scheduling Problem,” Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, 2009.
- <sup>8</sup>Montoya, J., Z. Wood, and S. Rathinam, “Runway Scheduling Using Generalized Dynamic Programming,” Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, 2011.
- <sup>9</sup>Jung, Y., T. Hoang, J. Montoya, G. Gupta, W. Malik, and L. Tobias, “Performance Evaluation of a Surface Traffic Management Tool for Dallas/Fort Worth International Airport,” Proceedings of the 9<sup>th</sup> USA/Europe Air Traffic Management Research and Development Seminar, Berlin, Germany, 2011.
- <sup>10</sup>Wood, Z., Kistler, M., Rathinam, S., and Jung, Y., “A Simulator for Modeling Aircraft Surface Operations at Airports,” AIAA Modeling and Simulation Technologies Conference, AIAA 2009-5912, Chicago IL, 2009.
- <sup>11</sup>Federal Aviation Administration, “Airport Efficiency Daily Configuration By Hour Report.” Aviation System Performance Metrics. December 2011 <<https://aspm.faa.gov/aspm/ASPMframe.asp>>
- <sup>12</sup>Bly, E., “Effects of Reduced IFR Arrival-Arrival Wake Vortex Separation Minima and Improved Runway Operations Sequencing on Flight Delay,” M.S. Thesis, Massachusetts Institute of Technology, 2005.
- <sup>13</sup>Federal Aviation Administration, “Aeronautical Information Manual: Official Guide to Basic Flight Information and ATC Procedures,” 2012