STRUCTURED DEPLANING VIA SIMULATION AND OPTIMIZATION

Andrew Wald (andrewjohnwald@gmail.com)
Mark Harmon (markharmon2012@u.northwestern.edu)
Diego Klabjan (d-klabjan@northwestern.edu)
Department of Industrial Engineering and Management Sciences
Northwestern University, Evanston, IL

ABSTRACT

Deplaning naturally occurs row by row down the length of an aircraft. Using simulation and optimization, we design deplaning strategies (e.g., deplane by group and/or column) that significantly reduce the overall unstructured deplaning time. The evaluations derived from a combination of optimization and simulation were tested across several equipment types using data gathered through field observations for calibration.

1. INTRODUCTION

Quick, consistent aircraft turns are critical to an airline's success and customer's satisfaction. The Air Transport Association estimates that delay minutes cost U.S. airlines a total of $6.1 billion in 2009 [ATA 2011]. Every cost of crew time, ramp procedures, and fuel burned add up and thus every minute counts.

An aircraft turn can be divided into two major parallel streams of work: tasks that take place "above the wings," and those that take place "below the wings." Major "below the wings" operations include catering, fueling, and cargo loading/unloading; boarding, deboarding and cabin cleaning take place "above the wings." The total turn time is determined by the longer of the two work streams. The ability to turn a plane quickly helps an airline:

- maintain on-time arrivals and departures,
- recover from irregular operations (weather, maintenance, etc.),
- reduce costs, and
- keep customers happy.

Airlines currently allow passengers to deplane in an unstructured fashion. We spoke with an industrial engineer at a legacy carrier, who has conducted many aircraft turn time studies to determine how quickly deplaning takes place. His studies show that customers deplane at an average rate of 15 to 17 passengers per minute, independent of aircraft size. Since passengers deplane faster than they board, airlines have focused their analysis efforts on enplanement. Structuring the deplaning process is a hidden improvement opportunity.

There are inefficiencies in current-state deplaning. Given that passenger speed remains constant, a steady stream of passengers would be preferred since gaps would indicate flow stoppage. These gaps represent opportunity for improvement. The main cause of these gaps is aisle interference, which is discussed below.

Both boarding and deboarding involve passengers walking down the aisle, the stowing/retrieving of luggage, and customer interactions in the cabin. The deplaning process, however, is significantly different than boarding. The aim of a structured boarding strategy is to minimize the amount of aisle interference (a passenger is prevented from advancing down the aisle) and seat interference (a passenger is prevented from moving into her row because another passenger is in the way). When passengers deplane, seat interference is not a factor – only aisle interference. During deplaning passengers within a row on the same side of the aisle have to follow their order predetermined by seat arrangement with no flexibility. When boarding, a passenger having an aisle seat must allow later coming passengers with a window/middle seat to enter into the row. This is the main reason why we believe that a dedicated deplaning simulation model is needed to study deplaning strategies; a boarding simulation "in reverse" is not representative of the process.

This work provides a study of deplaning by developing a novel stochastic deplaning model which is solved by optimization algorithms that use simulations as function evaluators. The simulation model takes a deplaning strategy and simulates the deboarding time. The main problem is to find an optimal strategy, i.e., how many groups should a deplaning strategy have and which rows or/and columns should be included in each group. We present and compare two stochastic optimization methods - a genetic algorithm and a stochastic gradient method - applied to this deplaning model in the search for good deplaning strategies. Finally, we evaluate strategies that have experimentally been shown to decrease deplaning time across multiple aircraft types, Zhao [2007].

Promising deplaning strategies are compared to current-state deplaning times to assess the degree to which they could expedite the deboarding process. Our simulations suggest that structured deplaning may reduce deplaning time by over 40% on a full aircraft, while significantly reducing deboard time variability.

Our work has significant contributions for deplaning of aircraft. First, we propose a simulation model that takes the unique features of deplaning into account, e.g., the handling of carry-on bags and the interference of passengers. Second, we apply simulation-optimization techniques to compute good deboarding strategies. The majority of the boarding research work is purely simulation based with only a few manuscripts focusing on the optimization aspect. On the deplaning side, we are not aware of a single work exploring optimization. Third, we propose a stochastic gradient-based optimization approach. To the best of our knowledge, gradient-based stochastic algorithms have never been applied for aircraft boarding or deboarding.

In section 2 we provide a literature review. Section 3 introduces our deplaning model logic, assumptions, and model parameters (collected from time studies on a major legacy carrier). The custom-built optimization methodologies we applied to this simulation model are outlined in Section 4. Evaluation of these structured deplaning strategies is presented in Section 5. We take a critical look at strategy implementation from a real-world standpoint, and the balance between potential benefits and disadvantages of structured deplaning, in Section 6.

2. LITERATURE REVIEW

The topic of deplaning is relatively new in research. Several papers have been written on strategies that can be used to reduce the time it takes to board passengers – primarily because airlines already enforce how and when people board. Many studies focus on simulation models for boarding, e.g., Van Landeghem and Beuselinck [2002], Ferrari [2005], Van den Briel et.al. [2002], and select works focus on analytical solutions, Bachmat et.al. [2009]. Optimization-based methods are either based on Markov chains, Steffen [2008], or genetic algorithms, Soolaki et.al. [2012], Li et.al. [2007]. We next summarize work dealing with deboarding.

Zhao et.al. [2007] suggest that the best deplaning strategies are essentially backward boarding strategies. While this technique would probably result in an improvement over unstructured deboarding, the expected benefits of this tactic were not explored in their work. The authors did not build a deplaning model or attempt to optimize the deplaning process, which is the focus of our work.

Li et.al. [2007] focus on boarding but they reverse their boarding simulation to simulate the deplaning process. While they use genetic algorithms for boarding, they only consider unstructured deboarding. They model the aircraft as a series of "processors" which logically route passengers around the aircraft. The model accounts for random perturbations down the aisle of the aircraft (i.e., a fumbling passenger, a bag getting caught in an aisle), and the authors simulate aisle interference by constructing a distribution for stowing/retrieving baggage. The parameters and assumptions of this model were never validated with real-world data. Under their model, the time needed to retrieve a bag decreases exponentially as the occupancy of the total aircraft decreases. However, it is known that bag interactions are a local phenomenon and that the time to retrieve a bag does not vary as an aircraft deplanes. These are known facts in the airline industry and were also observed in our field study. Our simulation work differs since we do not make assumptions without basis in observational facts. Baggage is not modeled with our simulation because the act of collecting baggage is rolled into an "aisle blockage parameter" that is discussed later. Our work also applies optimization algorithms to the deplaning model to explore how efficient a deplaning strategy can possibly be.

The most relevant deplaning work has been done by Yuan et.al. [2007]. The authors use a dedicated deplaning simulation model to test boarding strategies in reverse to see if any significantly reduce deplaning time. They model the aircraft as an array of cells that hold passengers. Passengers get up from their seats, block the aisle for a deterministic amount of time (retrieve luggage), and exit the aircraft. In our work the blockage time is stochastic as opposed to deterministic. The authors examined the effects of four deplaning strategies on the A320, and concluded that a good structured deplaning results in a 21% reduction in deplaning time. While their deplaning strategies are determined in advance, we obtain such strategies by optimization. Our work also considers three additional equipment types – CRJ-200, B763, B752 – and we compare their deplaning time reduction of 21% with our experimental findings.

There are two main differences between the status quo deplaning models mentioned above and the model created herein: the way baggage is simulated and the way passengers are processed. In the aforementioned works, customers retrieve baggage for an amount of time that is either deterministic or dependent on the load factor of the aircraft. While complex, there is no real-world data presented to support the validity of these assumptions. In our simulations, the time for which each passenger blocks the aisle is generated from a distribution based on a real-world deplaning time study.

There is an extensive amount of literature regarding iterative optimization algorithms. David Goldberg suggests that an evolutionary algorithm, such as a genetic algorithm, is a good solution technique for many problems, Goldberg [1989]. We applied a genetic algorithm to our simulation model to see how it handles the stochastic nature of deplaning. Our algorithm formulation and its parameters closely follow the general-purpose, well-studied formulation suggested by Masatoshi Sakawa's book on genetic algorithms, Sakawa [2002]. Stochastic gradient approaches have also been around for many years and extensively used in applications. We refer the reader to the book by Ruszczynski et.al. [2003] for further details.

3. SIMULATION MODEL

The only practical method to structure the deplaning process is to assign a “deplaning group” to each passenger, with the expectation that a passenger deplanes when her group is called. Similar to boarding, deplaning groups can be assigned to passengers as a function of where they are seated on the aircraft. Deplaning groups can be assigned to individual passengers or they can be assigned en mass to passengers seated in certain zones of the aircraft. Our objective is to minimize the deplaning time of the aircraft through the use of deplaning group assignments.

The goal of the model is to capture the inherent benefits of a structured deplaning strategy without complicating the model with unsubstantiated assumptions. The simulation model is developed to assess how various deplaning approaches perform. An aircraft is represented by a rectangular array comprised of a set of cells. Each cell represents a space that can only be occupied by one passenger at a time; cells can be seats or units of aisle space. The simulation model begins by allowing passengers assigned to the first deplaning group to unbuckle, collect their things, and occupy the aisle. Just as in real-world deplaning, the door does not immediately open; in the model, the aircraft door opens only after steady-state has been achieved in the aircraft. Once the flight door has opened and all members of the first deplaning group have exited, the next group is permitted to deplane. This process continues until all deplaning groups have been called and the last passenger has exited the aircraft. *Figure 1* illustrates passenger cells, the group each row is in, and the direction of travel for the passengers down the aisle.

The simulation models passenger interaction by “processing” aisle cells, in order, from the front of the plane to the back of the plane. If two or more passengers want to occupy the same aisle cell (i.e., a customer in the aisle wants to advance, but a seated customer also wants to move to that aisle cell to collect her belongings), they have an equal chance of occupying the cell.

If a passenger moves from her seat into the aisle, she will occupy that cell for a period of time known as an “aisle delay.” Passengers behind the delayed person will remain “stuck” until the passenger has finished collecting her belongings. Once the aisle delay is complete, the passenger will advance down the string of aisle cells toward the exit (and so will those behind her).

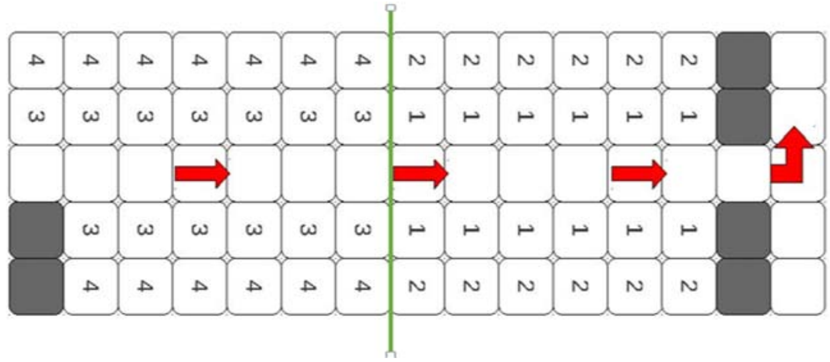

Figure 1: Model of a CRJ-200 with Deplaning Assignments

Baggage is not explicitly modeled – the time it takes to retrieve baggage is rolled into the aisle delay distribution. It is assumed that a passenger's baggage is approximately stowed at her feet and in the overhead bins above her head. The extent to which this assumption is true depends mostly on how customers board the aircraft. The assumption is more valid if customers are boarded using an outside-in approach; if passengers board by zone, it may be less true that a given passenger's baggage is close to her seat.

4. METHODOLOGIES

The large number of possible deplaning strategies, paired with the stochastic nature of the simulation model, necessitates optimization algorithms.

Three optimization approaches were applied to our model to search for better deplaning strategies: a stochastic gradient, nested stochastic gradient, and genetic algorithm. These algorithms were used to solve the problem of optimal passenger deplaning group assignment with the objective to minimize deplaning time. Formally, we model the problem as

$$min_X \mathbf{E}[f(X,\omega)]$$

where f is our merit or objective function, a function that returns the total time needed to deplane an aircraft given a deplaning assignment X, and ω is the exogenous stochastic process governing a deplaning. Function f and stochastic process ω (representing the aisle delay) were described in Section 3. It is clear that f is not given by a closed form expression but by a numerical simulation.

The vector X, which encodes a deboarding strategy has a group number for each seat, i.e., its dimension equals the number of sets on the aircraft and each coordinate has values 1,2,3, … up until the total number of groups (which can go as high as the number of seats). This encoding easily captures groups of rows and groups of seats by column, and any combination of these two strategies.

In Figure 1, we exhibit a sample deplaning strategy. This strategy is split into two groups by the diving line in the middle. Group 1 has a row strategy of (2,1,1,2) and group 2 has a strategy of (4,3,3,4). In this example, X, which is simply a representation of the number of rows pertaining to a row assignment, equals to $X= (x_1,x_2)$=(6,7) where x_1 and x_2 correspond to the first and second groups. Later, we will be taking the gradient of our variable X, i.e., slightly perturbing x_1 and/or x_2, e.g. $f((6,7), \omega)$-$f(5,8, \omega)$ for a realization ω.

Stochastic Gradient

In each iteration of the stochastic gradient we store a feasible deboarding plan. First, a sample is generated (or a partial simulation is executed) and then an approximation to the gradient at the incumbent solution given a realization is computed. The new solution is then moved in the direction of the gradient governed by a specified step size.

Since our decision space is discrete we use the notion of a discrete gradient by perturbing coordinates by integer values. Since the computation of f over a few samples is expensive, it is computationally prohibitive to perturb the decision vector in each individual coordinate. For this reason we select several coordinates and perturb them simultaneously.

At a high level, the algorithm begins by assigning large chunks of rows to deplaning groups (each row is assigned to a group). In each group, there is a certain pattern that does not change during the algorithm. In essence, this algorithm is trying to find the best number of rows assigned to each group given an a priori column strategy in each group. For example, *Figure 2* shows a possible configuration with three groups. Within each of these groups, all of the rows have the same strategy (i.e., all of the rows in group 1 have the strategy {2,1,1,2} which means that the aisle seats deplane first, followed by windows seats in these rows). Given such deplaning group assignments, the deplaning of the aircraft is simulated multiple times to establish a baseline solution value. Then, the number of rows containing each deplaning strategy is perturbed. For instance, a few simulations would be conducted with 6 rows in group 1 (which also implies that the number of rows in group 2 becomes 3). Once the relative advantage (or disadvantage) of the strategy in group 1 is determined, the algorithm chooses to add a row (or remove a row) in group 1 based on the incremental change in the objective function value. Essentially, group perturbations generate an approximate gradient. The optimization algorithm is as follows.

1. *Initialization:* Divide the aircraft evenly into $\boldsymbol{N}$ groups of rows, where N is a parameter.
2. In each group, assign seats according to a "column strategy." This implies that each row within a group has the same strategy.
3. *Evaluate incumbent solution:* Simulate the deplaning of the aircraft a number of times, and establish an average deplaning time.
4. *Gradient estimation:* Perturb the aircraft's deplaning assignment by adding or subtracting a row from each of the groups. We independently generate a perturbation for each group (for a group we add a row and subtract a row to an adjacent group). Simulate deplaning a number of times under each perturbation, and establish average deplaning times under the new assignments.
5. Define the value of the gradient as the marginal benefit of adding a row: $\mathbf{E}[f(X+u_i,\omega)]$ - $\mathbf{E}[f(X,\omega)]$ where X is the incumbent solution and u_i encodes the vector representing an extra row in group i.
6. *Select group to add:* If one or more gradient values are negative (adding a row to these groups of the aircraft is likely to reduce the overall deplaning time), select a group at random where the selection probability is proportional to the gradient value. A row will be added to this group.
   - If none of the gradient values are negative (adding to any of the groups is likely to increase the overall deplaning time), then exit the algorithm.

7. *Select group to remove:* If one or more gradient values are positive (adding a row to these sections of the aircraft is likely to increase the overall deplaning time), select a group at random where the selection probability is proportional to the gradient value. A row will be removed from this group.
    - If none of the elements of the gradient are positive, select a group uniformly at random. A row will be removed from this group.
8. Adjust the two selected groups and go to step 3.

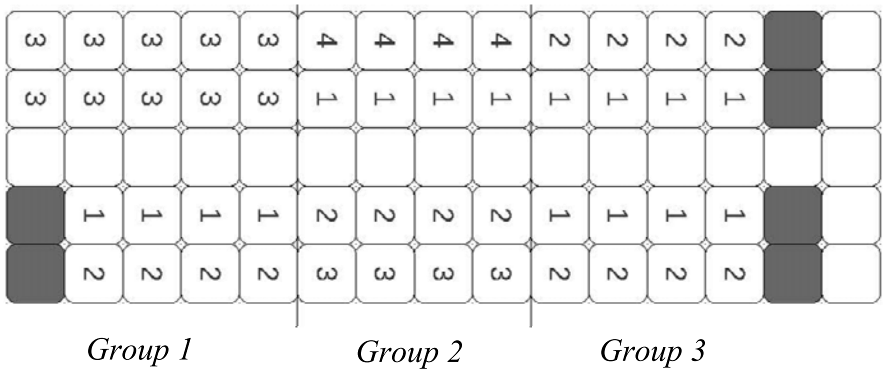

Figure 2: Group Deplaning Assignments for CRJ-200

In each iteration, if both positive and negative gradient values are observed, then the algorithm adjusts four groups: it adds a row to the selected group in step 6 and subtracts a row in an adjacent group; and it subtracts a row from the selected group in step 7 and adds a row to an adjacent group. An alternative variant is to perform these two steps in two separate iterations. Due to the high execution time of the simulation, the presented strategy performs better.

We note that it is not required to specify if we add the 'left' or 'right' row to a group. It only suffices to know the number of rows in each group and then the specific configuration can be obtained by following these numbers from the beginning of the aircraft to the end. In *Figure 2*, the incumbent assignment is (5,4,4), 5 rows in group 1, etc. If the perturbation corresponds to (6,4,3), then this uniquely defines the next configuration.

Nested Stochastic Gradient

A restriction of the stochastic gradient algorithm presented in the previous section is the fixed configuration in each group. For greater flexibility, these configurations should also be dynamically adjusted. The nested stochastic gradient algorithm introduces an extra level of flexibility by taking into account different configurations within groups.

This method results in a greater variety of solutions considered, but it is also more computationally expensive. The algorithm is identical to the stochastic gradient method, except that within each loop the deplaning strategy assigned to each group, i.e. the ordering of deplaning by-passenger, of the aircraft is optimized. For instance, the three groups in Figure 2 are constantly reassigned new deplaning strategies. Groups 1, 2, and 3 are isolated, deplaned separately, and assigned the best possible deplaning strategy. The algorithm goes on to determine the currently best number of rows that the three groups should occupy in the aircraft.

Between steps 3 and 4 we now have the following step that perturbs configurations within groups and the number of rows in each group independently.

- For each group
  - We randomly select a column in the group and assign it a random column number.
  - We simulate the deplaning for this group independently of the other parts, i.e., the number of rows in each group remains the same and the configurations within other groups also do not change.
- We compute the gradient value for each group and accordingly adjust the configuration in the group with the highest gradient value.

Genetic Algorithm

A genetic algorithm was also used in the search for good structured deplaning tactics. The book written by Sakawa [2002] in particular provides a great introduction to this class of algorithms. We applied a genetic algorithm to our problem because the method has been used in other papers to optimize boarding strategies and we also want to benchmark the performance of the two stochastic gradient-based algorithms.

A chromosome corresponds to our candidate solution X. The fitness of each chromosome is determined by simulating deplaning per the chromosome's encoding. Several genetic operations allow for the population to retain good traits and shed bad traits, and evolve on better deplaning strategies. The algorithm used is detailed next.

1. Divide aircraft into N randomly sized groups of rows.
2. In each group, assign rows according to random "column strategies."
3. Encode each obtained strategy as a chromosome, initializing a population of size P by repeating steps 1 and 2 a total of P times.
4. Simulate deplaning using each chromosome, and determine its average fitness.
5. Use roulette-wheel selection, an operator used for selection in the genetic algorithm, to discard half of the population based on each chromosome's relative fitness level. Double the selected chromosomes and restore the population to size P by applying steps 1 and 2 as needed.
6. Randomly pair chromosomes and perform crossover operations. The probability of crossover used is 60% for any pair of genes.
7. Mutate all chromosomes, with a probability of 0.5% for any given gene.
8. Repeat steps 4 through 7.

5. RESULTS

All algorithms were performed on a Linux server equipped with a 64-bit, 3.2 GHz Intel Xeon processor and 6 GB of RAM. The deplaning simulation was written in MATLAB to leverage the language's built-in functions and animation capabilities. The entire deplaning process can be animated on the screen.

Calibration

The simulation model is configured to study structured deboarding strategies across the following aircraft types:

- CRJ-200, 50 seats, 1 cabin
- A320-200, 144 seats, 2 cabins

- B757-200 a.k.a. B752, 182 seats, 2 cabins

The time needed for one passenger to advance one row down the aircraft is assumed to be equal to one second of real time. This assumption is in line with previous time studies of aircraft boarding. Van den Briel et al. [2003] as cited by Yuan et.al. [2007] reports that the average time a customer takes traversing one row of an aircraft is approximately .95 seconds when boarding.

Observations based on a major legacy carrier, during the summer of 2010, were used to model the total time that a passenger blocks the aisle of the aircraft in the act of moving from her seat into the aisle and collecting her belongings. A summary of the flight segments, equipment types, and load factors in the time study are presented in Figure 3. Each observation corresponds to a particular flight on the given segment. During each observation, i.e. segment, we manually observed and recorded the various moves during deplaning (we were passengers on these segments).

Aisle Delay Observations			
Date	**Segment**	**Equipment Type**	**Observations**
April 2010	ORD-MSP	A320	7
	MSP-ORD	A319	6
	ORD-MSP	A319	6
	MSP-ORD	A320	7
May 2010	ORD-FSD	CRJ200	5
	FSD-ORD	ERJ145	5
June 2010	ORD-DCA	A320	7
	DCA-ORD	A320	5
	ORD-BWI	A320	7
	BWI-ORD	B752	7
			Total: 62

Note: Passenger loads for all flights were between 75% and 100%.

Figure 3: Aisle Delay Observations

There are not any significant differences in aisle delay times amongst the various equipment types or segments observed. The 62 observations followed an approximately normal distribution with a mean of 5.3 and standard deviation of 1.3 seconds. This distribution was used to generate the "aisle delay" parameter for each passenger object in the simulation model.

To validate the model, we ran unstructured deplaning scenarios on the four equipment types studied. Shown in Figure 4, we generated prediction intervals of total deplaning time for the aircraft obtained by simulation (95 percent confidence interval), and compared these intervals to the expected rate of 16 passengers per minute described by experts in the field (Observed Average). We observe that the model is consistent with our expectations for the narrow-body aircraft. Additionally, front-to-back behavior emerges during deplaning, which further confirms that the model is well-suited to describe the deplaning process. A picture of this emerging behavior is shown in Figure 5.

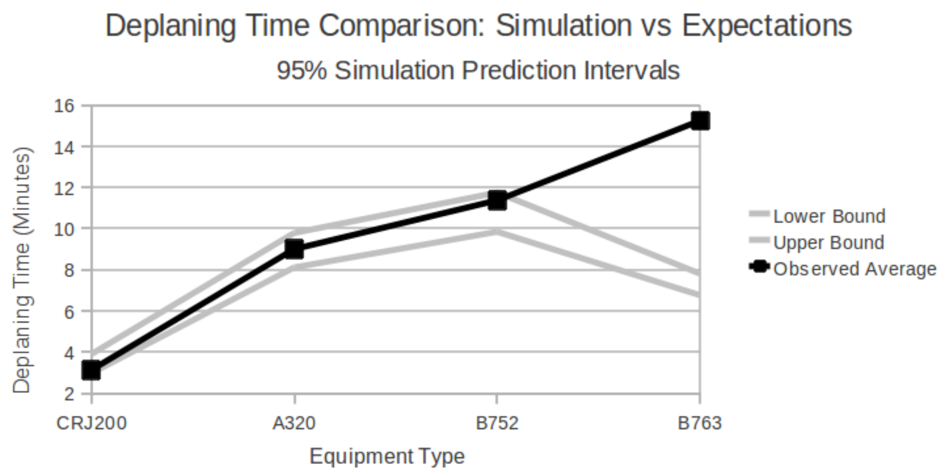

Figure 4: Simulation Validation, 100% Load Factor

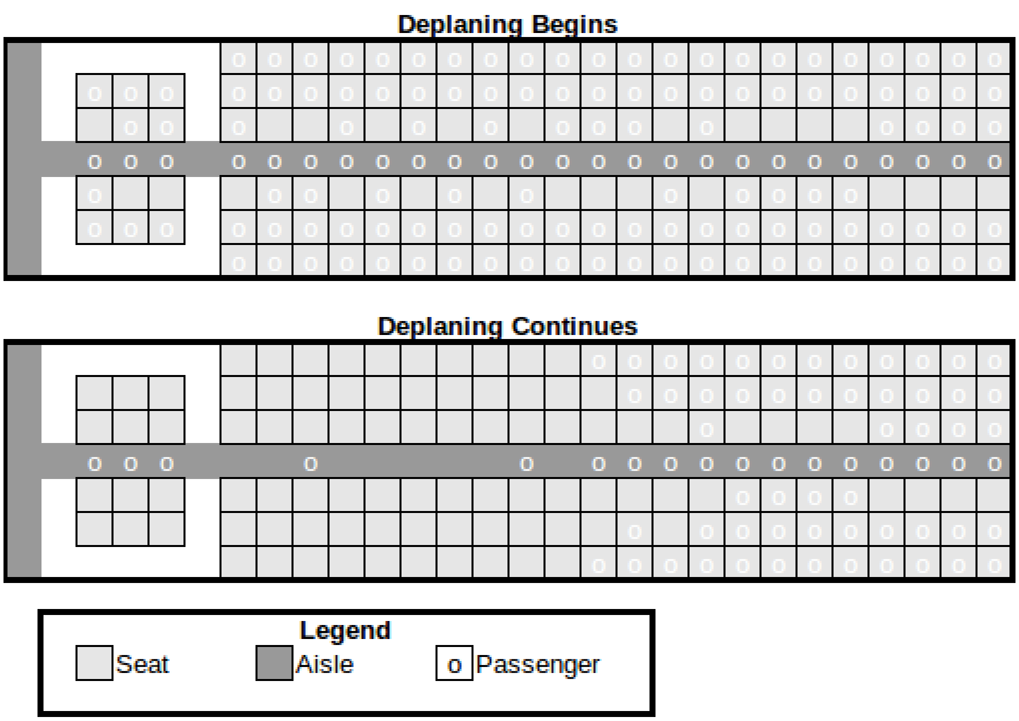

Figure 5: Front-to-Back Deplaning Behavior, A320, 100% Load Factor

Deboarding Strategies

In this section we focus on good deplaning strategies and compare algorithmic performance later. Solutions presented here correspond to the best solution obtained from the three algorithms.

The best solutions found by optimization across all three algorithms for the CRJ-200 are displayed in Figures 7 and 8. All strategies that significantly decrease deplaning time were found to exhibit substantial either "inside-out" properties (Figure 7) or "one-column" properties (Figure 8).

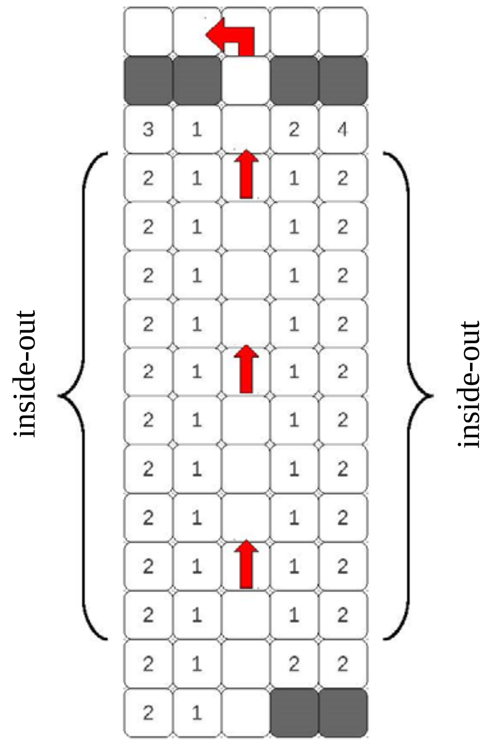

Figure 7: Almost Inside-Out Solution, CRJ-20

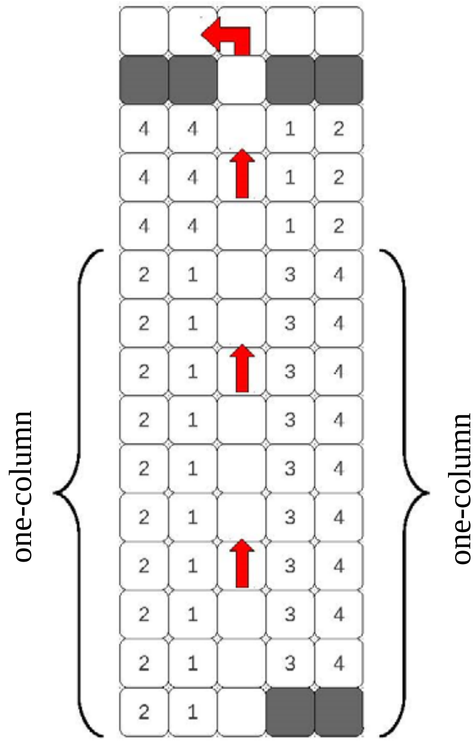

Figure 8: Almost One-Column Solution, CRJ-200

Inside-out strategies are characteristic of the bracketed section in Figure 7. The passengers with aisle seats deplane first; once those passengers have fully deplaned, the next-closest columns to the aisle deplane, and so on, until all passengers have deplaned. With four seats in a row, a CRJ-200 has two deplaning assignments following an inside-out strategy; an A320 or B752 would have three deplaning assignments, as they are six seats across. The one-column strategy bracketed in Figure 8 allows one column of the aircraft to deplane at a time. A CRJ-200 would require four deplaning assignments to follow the strategy and an A320 or B752 would need six deplaning assignments. The optimal solutions have large portions of inside-out or one-column. We call them almost inside-out/one-column as opposed to pure inside-out/one-column that strictly follow the underlying pattern.

In Figure 9, we compare the best solutions our algorithms found with aircraft entirely deboarded with pure one-column and inside-out strategies. We see from this figure that the pure one-column strategy is the best performer of all deplaning strategies for the simulated narrow-body aircraft. The optimization algorithms never find a pure strategy, but they always contain a large portion which is a pure strategy. For this reason they are slightly inferior to the pure strategy derived from them. The 'unstructured' portion of solutions obtained by optimization of CRJ-200 constitute less than 10% of seats. For A320 and B752 this percentage is slightly higher.

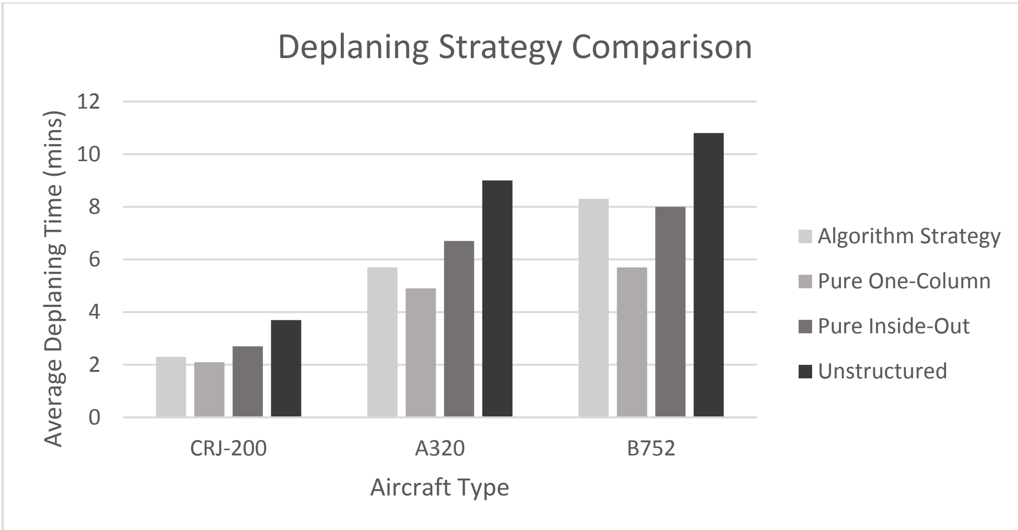

Figure 9: Deplaning Strategy Comparison for All Aircraft Types, 100% Load Factor

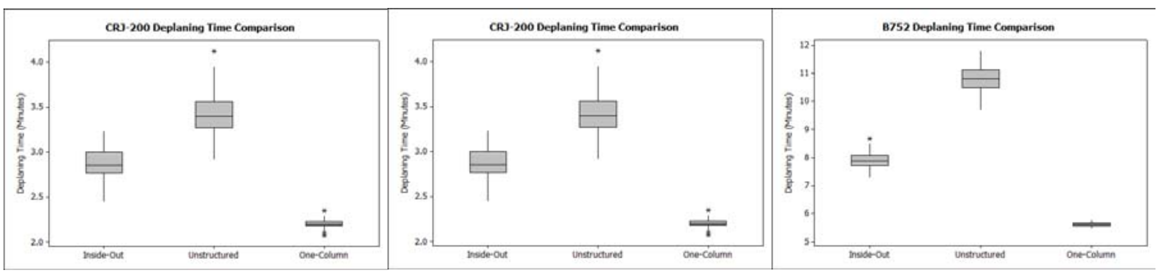

Figure 10: Deplaning Time Comparison under 100% Load Factor

The variability of the deplaning strategies is compared in Figure 10. The results suggest that the best implementable strategy is the one-column strategy. The best solutions contain large pockets of the one-column strategy; when deplaning is uniformly one-column, average deplaning time and deplaning variability are minimized for the simulated aircraft types. Figure 11 summarizes the predicted deplaning time improvements from the one-column deplaning strategy in comparison to unstructured deplaning.

Structured vs. Unstructured Deplaning Comparison				
Equipment	**Average Unstructured**	**Average Structured**	**Average Time Saved**	**StDev Reduction**
CRJ-200	3.4 Minutes	2.2 Minutes	1.2 Minutes (36%)	81.00%
A320	9.0 Minutes	4.8 Minutes	4.2 Minutes (47%)	85.00%
B752	10.8 Minutes	5.6 Minutes	5.2 Minutes (48%)	87.00%
B763	7.3 Minutes	5.9 Minutes	1.4 Minutes (19%)	62.00%

Figure 11: Structured vs. Unstructured Deplaning Comparison, 100% Load Factor

Optimization Algorithms

From Figure 9 we observe that the optimum strategies become less and less effective as the aircraft becomes larger. The best solution is only 8% slower (suboptimal) than the one-column strategy for the CRJ-200 while it outperforms unstructured deplaning by 30%. For the B752, the best solution is over 52% slower than the one-column strategy for the B752, and it manages to improve on unstructured deplaning by 21%. This is to be expected, since as the aircraft gets larger, computations get more and more expensive – slowing progress toward better solutions. The gradient-based solver has difficulty filling the entire aircraft with a good deplaning strategy because each iteration of our algorithm estimates a gradient stochastically, then acting on that information.

To illustrate how the different optimization algorithms compare, Figure 12 shows sample paths for the stochastic gradient and nested stochastic gradient applied to the CRJ-200. A sample CRJ-200 genetic algorithm sample path is shown in Figure 12.

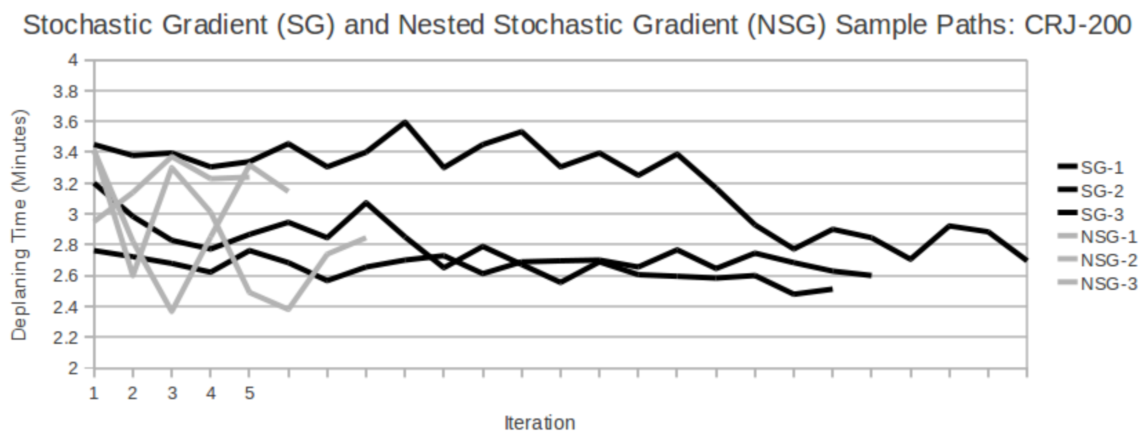

Figure 12: Stochastic Gradient and Nested Stochastic Gradient Sample Paths, CRJ-200

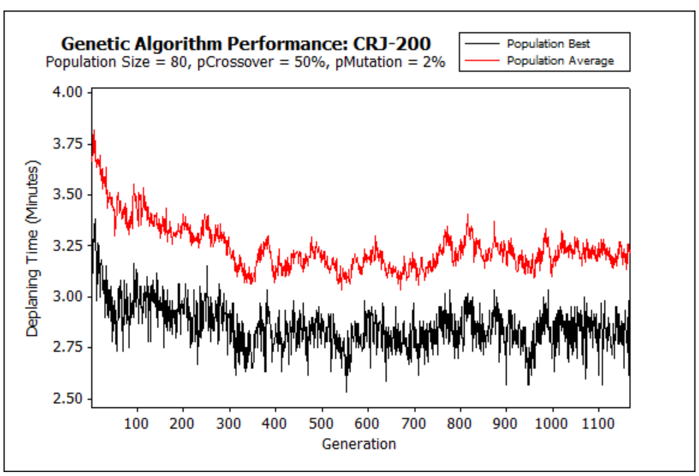

Figure 13: Genetic Algorithm Sample Path, CRJ-200

The stochastic gradient approach quickly converges in a stable, predictable manner. Though the method is quick to converge, it is unable to dive deeply enough into the problem to find better deplaning structures (those that are pure one-column). It is interesting to note that the algorithm relatively quickly finds an almost one-column strategy but then it keeps ‘tweaking’ it so that it never results into a pure one-column strategy. The nested stochastic gradient method is relatively slow and unpredictable across all of the equipment types. The sample paths show the non-convergence of the nested stochastic gradient algorithm. The method tends to oscillate between very good solutions and very bad solutions, until the exit conditions are finally tripped. The reason for this behavior lies in the greater flexibility of perturbing solutions. The genetic algorithm does improve aircraft deplaning throughout its runtime, but only after hundreds of computationally intensive iterations the improvement becomes substantial. In each iteration, the improvement is minor and thus it accumulates very slowly. Even when primed with known good solutions; it never makes significant progress toward better solutions.

It is interesting to note that the optimization algorithms are able to reduce the number of groups to a single group with a fixed column strategy regardless of the number of starting groups *N*. Note that a fixed column strategy for a single group is not the same as unstructured deplaning. This clearly indicates that they are efficient as guidance toward the absolute best solution.

A table comparing the three methods as applied to the CRJ-200 is shown as Figure 14. The average run time reports the execution time of the algorithm. The second row describes the average reduction in deplaning time compared to that of unstructured deplaning. ``Suggestion Variability” scores the oscillations in the deplaning time between different simulation runs (realizations).

Algorithm Comparison: CRJ-200			
	Stochastic Gradient	**Nested Stochastic Gradient**	**Genetic Algorithm**
Average Runtime	19 Minutes	55 Minutes	> 1000 Minutes
Average Deplaning Time Reduction	25 Seconds	27 Seconds	32 Seconds
Suggestion Variability	Low	Very High	Low

Figure 14: Overall Algorithm Comparison, CRJ-200

Across all three equipment types, the same trends were observed. While the least convergent of the three, the best structured deplaning strategy suggestions come from the nested stochastic gradient output. Despite its high variability, good solution quality and acceptable running times make this algorithm the most useful.

The optimization algorithms run quickly for the CRJ-200 but slowly for the A320 and B752. Compared to the CRJ-200, runs for the A320 took an average of 7.5 times longer and those for the B752 took 13 times longer. Nearly all of the extra runtime can be attributed to the deplaning simulation, not the optimization algorithms themselves. The time to simulate the deplaning of an aircraft increases quadratically with respect to the number of passengers simulated. The simulation model needs to check for all passenger conflicts for every simulated timestep in order to accurately represent the deplaning process.

6. MANAGERIAL INSIGHTS

A feasible, implementable deplaning strategy must have the following characteristics:

- consistently reduces deplaning time over unstructured deplaning,
- simple to understand and follow,
- consistent across fleet types, and
- easily enforceable and not manpower intensive.

An inside-out or one-column deplaning strategy can realistically accomplish these conditions. There are multiple ways to implement this strategy. One way is for the flight attendants to announce the strategy. Other possible processes include printing the deplaning order on the boarding passes or have the assignment number printed on a passenger's seat for each row. Structured deplaning is probably best suited for domestic segments since domestic turns must be quick and are scheduled much more tightly than international turns.

However, there are some potential issues that could reduce the effectiveness of a structured deplaning strategy:

- failure to recognize "premier" customer status,
- could anger customers, especially frequent fliers or business travelers,
- need to account for passengers with tight connections.

We remark that these drawbacks apply also in the current unstructured deplaning.

We also performed simulations to see how lower load factors affect our proposed one-column deplaning strategy. As the load of the aircraft decreases, the use of a structured deplaning strategy becomes less advantageous. As depicted in Figure 15, we observe that the one-column strategy for narrow-body aircraft quickly loses ground to unstructured deplaning, even under the ideal simulation case where customer compliance is 100%. Larger aircraft maintain a large structured deplaning benefit for longer than smaller aircraft. This makes intuitive sense. Longer, larger aircraft will have more aisle conflicts than smaller, shorter aircraft. This difference is even more pronounced when load factors are low. A CRJ-200 has few aisle conflicts at load factors of 50% or less, so there is relatively little to gain from a deplaning strategy, but the same cannot be said for a B752.

When we inspect strictly the number of passengers by row, as in Figure 16, we see that the trend does continue. As the number of passengers decreases, a structured deplaning strategy does become less advantageous. However, we note that the smaller planes still yield significant gains on a by-passenger basis vs. the larger planes. Note that three passengers per row for the CRJ-200 is still an 80% load factor while the other two planes are at the 50% load factor for the same number of passengers per row.

We performed simulations to study how advantageous it would be to implement the one-column deboarding strategy only in the economy section of aircraft, letting the first-class passengers exit first. We note in Figure 17 that the effectiveness of a one-column deboarding strategy is severely compromised in smaller aircraft, but much of the benefits are retained for larger aircraft. Since the CRJ-200 configuration studied does not contain a first-class cabin, we assumed that "premier customers" were seated in the first two rows – and that they deplane first.

From the study, we conclude that a one-column strategy for the economy section should be considered for a domestic network narrow body aircraft fleet with a high load factor. This strategy can be easily applied to the economy class.

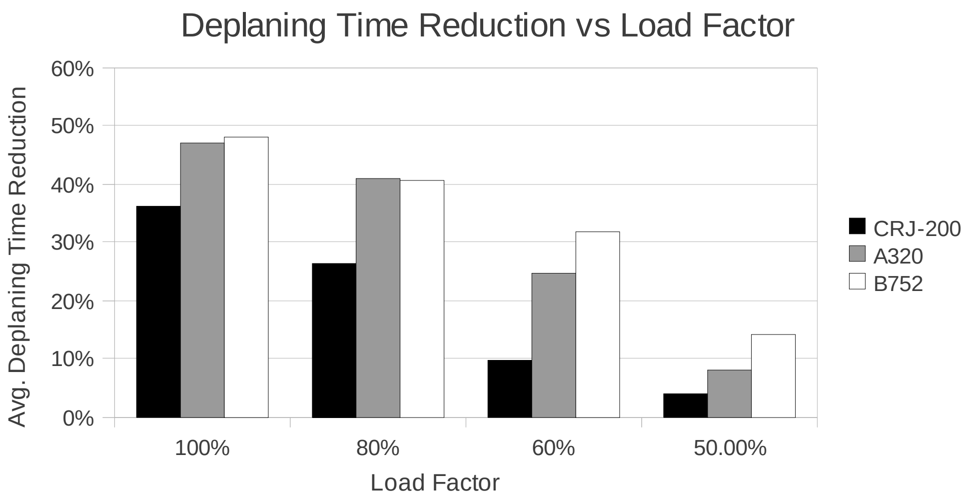

Figure 15: Average Deplaning Time Reduction vs Load Factor

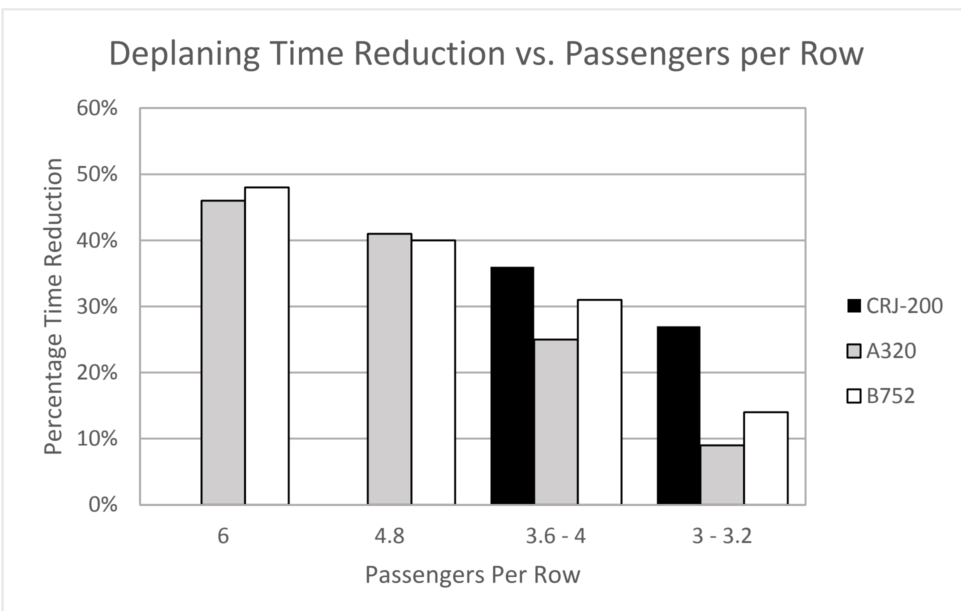

Figure 16: Average Deplaning Time Reduction vs. Passengers per Row

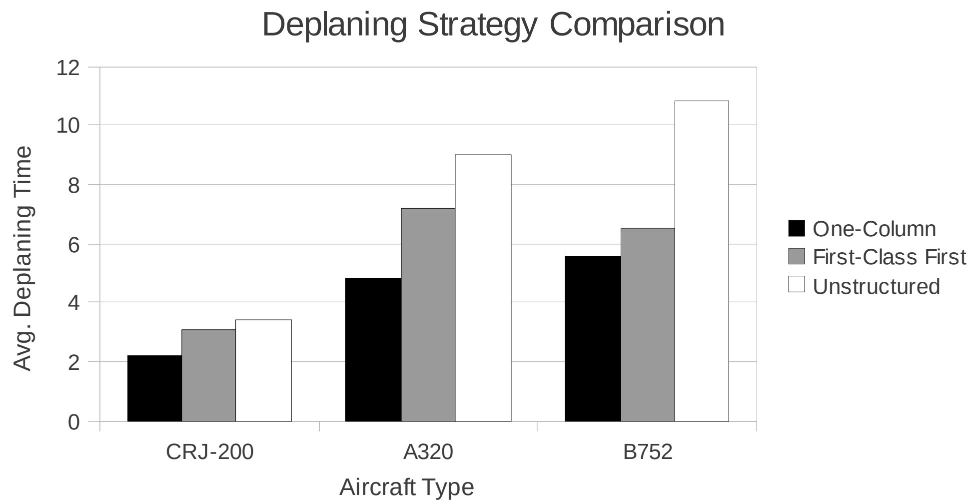

Figure 17: First-Class First Deplaning Strategy Comparison, 100% Load Factor

7. CONCLUSION

A good deplaning strategy is a tool that could be used to keep potentially delayed aircraft on track when turns are tight – helping to avert down-line problems throughout the day.

We conclude that a one-column deplaning strategy is the best way to deplane a moderately-full to full narrow-body aircraft. The optimization algorithms are able to find an almost optimal strategy. More importantly, a quick look at an optimized strategy reveals the pattern of the one-column strategy. It is extremely encouraging that the optimization algorithms are able to realize that there should be a single group.

Yuan et.al. [2007] report reduced deplaning times of 21% over the unstructured strategy. In Table 13 we report these reductions to be in excess of 35% for narrow-body aircraft. Since our parameters are based on field observations and they comply with the knowledge of industry experts, we are confident in reliability of our study.

Future research will focus on validating a model for wide-body aircraft and improving optimization techniques in search for good deplaning strategies. Other future work may also focus on field experiments implementing different strategies to obtain data through customer feedback and deplaning times.

BIBLIOGRAPHY

Air Transportation Association of America (2011) Annual and Per-Minute Cost of Delays to U.S. Airlines. Accessed in Feb, 2011. http://www.airlines.org/Economics/DataAnalysis

Bachmat, E., Berend, D., Sapir, L., Skiena, S., & Stolyarov, N. (2009) Analysis of airplane boarding times, *Operations Research*, 57, 499-513.

Ferrari, P. (2005). Improving passenger boarding in airplanes using computer simulations. *International Airport Review*.

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*.

Boston, MA, Addison-Wesley Longman Publishing Co.
Li, Q., Mehta, A., & Wise, A. (2007) Novel approaches to airline boarding. *Undergraduate Mathematics and Applications*, *28*, 333-352.
Ruszczynksi, A., & Shapiro, A. (2003) Stochastic Programming. Handbooks in Operations Research and Management Science, New York City, NY, Elsevier Science.
Sakawa, M. (2002) *Genetic Algorithms and Fuzzy Multi-objective Optimization*. Norwell, MA, Kluwer Academic Publishers.
Soolaki, M., Mahdavi, I., Mahdavi-Amiri, N., Hassanzadeh, R., & Aghajani, A. (2012) A new linear programming approach and genetic algorithm for solving airline boarding problem. *Applied Mathematical Modelling*, 36, 4060-4072.
Steffen, J. H. (2008) Optimal boarding method for airline passengers. *Journal of Air Transport Management*, 14, 146-150.
Van Landeghem, H., & Beuselinck, A. (2002) Reducing passenger boarding time in airplanes: A simulation approach. *European Journal of Operational Research*, 142, 294–308.
Van den Briel, M., Villalobos, J., Hogg, G., Lindemann, T., & Mule, A. V. (2005) America West develops efficient boarding strategies. *Interfaces*, 35, 191–201.
Yuan, B., Yin, J., & Wang, M. (2007) Star: (saving time, adding revenues) boarding/deboarding strategy. *Undergraduate Mathematics and Applications*, *28*, 371-384.
Zhao, L., Zhou, F., & Wang, G. (2007) Airliner boarding and deplaning strategy. *Undergraduate Mathematics and Applications*, *28*, 405-420.