A Co-evolutionary Genetic Algorithm to Avoid Airline Passenger Denied Boarding

Kaelan Engholdt Isaac Lock Jacob McAllister David Mathias
Department of Computer Science
University of Wisconsin – La Crosse
La Crosse, WI 54601
dmathias@uwlax.edu

March 29, 2019

Abstract

Airline overbooking, a strategy used to reduce empty seats on flights, sometimes results in a flight that has more passengers than seats. When this occurs, airlines compensate passengers willing to voluntarily relinquish their seats to avoid incidents of involuntary denied boarding. In this work, we use a coevolutionary, multi-objective genetic algorithms to evolve strategies for both airlines and passengers for achieving their goals in the event of an overbooked flight. Each passenger evolves to decide when to accept an offer based on the number of volunteers needed and the current offer amount with the goal of maximizing the compensation received. Each airline evolves an effective sequence of offer amounts with the goals of eliminating involuntary denied boarding events and minimizing the incentives paid to passengers. We demonstrate that our approach results in reasonable strategies for both airlines and passengers.

1 Introduction

To airlines, an empty seat represents lost revenue. Even when a flight is sold out, it can depart with empty seats. This is because passengers sometimes don't show up for a flight for which they have purchased a ticket. To mitigate this effect, airlines carefully track the no-show rates for their flight. They use this information to sell tickets beyond the capacity of the airplane used for a flight, a practice known as *overbooking*. One consequence of overbooking is that a flight sometimes has more checked-in passengers than seats.

When these situations arise, airlines offer compensation to entice a small number of passengers to relinquish their seats voluntarily. These offers typically start at approximately $200 and increase until a sufficient number of volunteers have been found. *Involuntary denied boarding* occurs when an insufficient number of passengers volunteer to travel on a later flight. In these cases, some ticketed passengers are unable to travel on their scheduled flight.

In April 2017, a well-publicized incident on a United Airlines flight shined a harsh light on involuntary denied boarding. A passenger who had already boarded flight 3411 from Chicago to Louisville, Kentucky, was forcibly removed from the flight and injured in the process. Video taken by other passengers showed a very unpleasant scene. The airline industry response was rapid. Major American carriers changed policies in an attempt to prevent similar incidents. One stated that no passenger would be removed from a flight, once boarded, to accommodate another passenger. Others significantly increased the maximum payments gate agents are authorized to offer passengers to volunteer for later travel. Delta and United promised amounts of up to $10000, a substantial premium over the previous maximum of $1350.

While both the airline and the passengers want to avoid incidents of denied boarding, some of their goals are in opposition. The goals of the airline are to avoid denied boarding incidents (that is, to find enough volunteers) and to minimize the total payout. Neither goal alone is sufficient for an effective strategy. A passenger's primary goal is to maximize the compensation received for relinquishing their seat. Secondarily, they may also want to maximize the probability that when they choose to accept an offer they actually receive that offer despite possible competition from other passengers.

The surest way to avoid involuntary denied boarding is to bound the tickets sold for a flight by the number of seats on the aircraft. As mentioned above, no-show passengers mean that this would result in flights departing with empty seats and, consequently, lost revenue for the airlines.

Load factor is a measure of the percentage of available seats sold. *Seasonally-adjusted load factor* is scaled to account for variations in the number of days in a month, holiday, and other seasonal activities that affect air travel. In January 2000, the seasonally-adjusted load factor for US air carrier domestic flights was 69.4%. By May 2018, that figure had risen to 84.3% [10]. Given this increase, accurate booking models are increasingly important. Thus, much research has been done on modeling airline overbooking to allow accurate prediction of the number of seats that can be sold for any particular flight [8, 12, 11, 13, 14, 1, 9]. This research spans disciplines including marketing, management, and operations research.

Little research has been done on the effects of denied boarding compensation. Garrow,

et al examine the effects of increasing denied boarding compensation [6]. They find that compensation is a very small piece of a complex problem, particularly in light of the fact that airlines factored financial losses incurred by ill-will resulting from denied boarding into their calculations of how much compensation to offer.

We are unable to find previous research that uses evolutionary computation, or other algorithmic approaches, to develop the sequence of offers airlines make to passengers in overbooking scenarios. Thus, we believe that this work is the first attempt to evolve effective airline strategies. Our work has limitations. We do not have access to the highly detailed data airlines track on a flight-by-flight basis to inform their decisions. Clearly, such information should be included to achieve an accurate model. However, this work serves as a proof of concept, demonstrating that an evolutionary approach may well aid airlines in determining how best to compensate passengers to avoid denied boarding.

In addition, we evolve passengers who attempt to maximize their compensation. This provides an adversarial but adaptive population for the airlines, helping to ensure that the airline strategies are not unduly influenced by assumptions about passenger behavior.

To that end, we use a co-evolutionary, multi-objective genetic algorithm to evolve strategies for both airlines and passengers for achieving their goals in the event of an overbooked flight. The passengers evolve to decide when to accept an offer based on the number of volunteers needed and the current offer amount. The airline evolves an effective sequence of offer amounts to avoid involuntary denied boardings and minimize the amount of compensation paid.

2 An Overview of Genetic Algorithms

Optimization problems require minimizing or maximizing one or more quantities to solve a problem. Examples include visiting some number of destinations while traveling a minimum distance (the well-known traveling salesman problem) and filling a truck with as many packages as possible while obeying a weight constraint. Because many such problems are computationally intractable, the best algorithms we have provide approximate, rather than optimal, solutions. Such algorithms are known as approximation algorithms. *Genetic algorithms* (GAs) are approximation algorithms that use the principles of Darwinian evolution. A GA begins with a *population* of initially random solutions to the problem and evolves them through some number of *generations*. Let n represent the number of members in the population. While there are many variations, the general template for a generation in a GA is this: The population produces n children, new solutions obtained by combining and mutating existing solutions. The children are evaluated and combined with the parents resulting in a population of size $2n$. The "best" n of the $2n$ members are selected to survive and the remaining members are discarded. Over a number of generations, this results in iterative improvement of the solutions.

Each member of the population is represented by a genome, an encoding of the solution. The information stored in the chromosome is entirely domain-dependent. For example, in the traveling salesman problem referenced above, a commonly used genome is an ordered list of the destinations. In a game theoretic problem such as the iterated prisoner's dilemma, it might be a bit string representing the decisions for a player to make in each of some

number of game states.

Evolution occurs through some combination of three genetic operators, each with many variations. These are *crossover*, which is analogous to reproduction; *mutation*, which is much the same as in biological evolution; and *selection*, which corresponds to survival of the fittest. In crossover, is a randomize process in which two *parents* are recombined to create two children. Each child receives some of its genome from each parent. Random mutations are applied to introduce additional variation in the gene pool. Selection typically involves choosing the best members of the population to survive into the next generation. Choosing the best members requires a method for ranking individuals. This is done via a *fitness function*. In its simplest form, a fitness function consists of the value of the quantity being maximized or minimized. Considering once again the traveling salesman problem, this is the path length. Individuals with shorter paths are considered more fit than those with longer paths. It is worth noting that due to local minima this is not always the case.

In many problems, there are multiple optimization objectives. In such cases, fitness is a more complex concept. Consider the problem of navigating an environment that includes obstacles. The goal is to move from a starting location to a destination, minimizing the distance traveled and the number of obstacles struck. We know that the shortest distance is the straight line from source to destination, however, that path is likely to strike obstacles. To decrease the number of obstacles struck, we must necessarily increase the distance traveled. Thus, we cannot minimize both objectives simultaneously, making it difficult to rank the individuals.

One way to handle this is known as *Pareto optimization*, named for Italian economist Vilfredo Pareto. It is based on *domination*, the idea that some solutions are objectively better than others. Let O be the set of optimization objectives. One candidate solution X *dominates* another candidate solution Y if $\forall\ o \in O, X[o] \leq Y[o] \wedge \exists\ o \text{ s.t. } X[o] < Y[o]$ [?]. A solution that cannot be dominated is known as *Pareto optimal*. The set of all Pareto optimal solutions defines the *Pareto front*. Since it is likely that the algorithm will not find a Pareto optimal solution, each candidate solution is assigned to one of some number of *domination fronts* in which each candidate in front i is dominated by at least one candidate in front $i-1$. Thus, the fronts impose a partial order on the candidates. Front 0 contains all non-dominated candidates. In a search for an optimal solution, front 0 constitutes the best found approximation of the Pareto front.

Central to making a GA effective is finding a genome that encodes a solution and allows for efficient evolution. There are two aspects to a representation: the information being stored in the genome and the encoding used to store it. The information encoded in the chromosome is highly dependent on the problem domain. As we explain in Section 3, our passengers and airlines have very different genomes because the results needed for each are quite distinct. Choosing an encoding involves multiple factors including the problem domain and the way in which the values need to evolve. For example, when using binary encodings, mutations often involve flipping bits. However, flipping a high order bit can cause a very large change in the value of the quantity represented by the bit string. In some domains, this is not desirable.

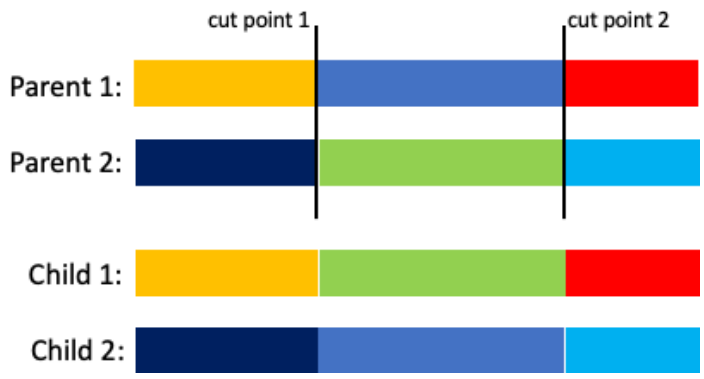

Figure 1: Figure Caption.

3 Our Algorithm

The problem we address in this work includes two populations: passengers and airlines. *Coevolutionary genetic algorithms*, first described in seminal work by Hillis [7], maintain multiple populations that are evolved in parallel. In a *cooperative coevolutionary algorithm*, multiple populations work together to solve a problem. In such a case, individuals are rewarded for working well with member of the other population. In this work, we implement a *competitive coevolutionary algorithm*. In this model, members in each population evolve at the expense of members in the other population. A gain for one is a loss for the other. In our problem, the passengers attempt to maximize the incentive received while airlines attempt to minimize payments made. Thus, each population is working against a moving target. The amounts of offers made to the passengers change from one generation to the next as do the behaviors of the passengers.

3.1 Airlines and Passengers

The passenger population is evolved using a single fitness objective: maximizing the incentives received. The genome consists of 256 bits. Each bit represents a decision: 0 indicates that the passenger will not accept an offer while 1 indicates that the will attempt to accept the offer (they may be unsuccessful as the number of offers is limited). The bit used to make a decision is determined by an index constructed from the offer amount and the number of offers being made (the number of volunteers needed by the airline). We use 5 bits to represent the offer amount, representing 32 buckets each with a range of $200. A further 3 bits represent the number of offers, representing decimal values from 0 to 7. Appending these two bit strings results in 8 bits. Converted to decimal, this gives indices from 0 to 255.

Two-point crossover is used for the passenger population. This method randomly chooses two indices, i, j in the parent genomes. Child 1 consists of genes $0..i-1$ and $j..255$ from parent 1 and $i..j$ from parent 2. Child2 is the symmetric case. Figure 1 illustrates this method. The mutation operator flips each bit in the genome with probability $\frac{1}{128}$ for an expectation of two bits flipped per genome. Because passengers are evaluated using a

single objective, selection simply chooses the passengers with the highest average incentive received.

The airline population has two fitness criteria: the payments made and the number of involuntary denied boardings. Both are minimized. Mirroring reality, denied boardings occur when the number of volunteering passengers is less than the number of seats needed by the airline. The airline genome consists of a sequence of 8 real-valued numbers. It represents the sequence of offers the airline will make to entice passengers to volunteer. Offers begin with the amount in gene 0 and progress through gene 7, as needed, to fund sufficient volunteers. Each gene value is limited to a predefined range. The ranges are shown in Table 1. The values were chosen with large overlap in the ranges for each round to allow airlines flexibility in evolution.

Gene	Min Value	Max Value
0	200	1200
1	400	1600
2	600	2400
3	1000	3000
4	1600	4000
5	2400	4800
6	3600	5400
7	4400	6400

Table 1: Allowable ranges for genes in the airline genome. Each gene represents an offer the airline will make.

The airline is evolved using simulated binary crossover [2]. This provides a search power when using a real-valued genome similar to that of one-point crossover when using a binary genome. The crossover operator is bounded to respect the ranges for each gene in the genome. Parameter η and the gene values of the parents determine each gene value for a child, within the defined ranges. Mutation is bounded polynomial in which each gene is altered by an amount proportional to the size of the range divided by parameter η [3]. Because the airlines use two optimization objectives, selection is via a form of Pareto optimization (see Section 2) due to Deb, *et al* and their seminal NSGA-II algorithm [4].

3.2 A Generation

The algorithm runs for a predetermined number of generations m. Each generation begins with a population of n_a airlines and n_p passengers. To create children, individuals within a population are paired and the crossover operator is applied with probability x_{over} p_a for the airlines and x_{over} p_p for the passengers. When a pair undergoes crossover, two children are produced. If crossover does not occur for a pair, each member is copied and the copies added to the children. After crossover, each child undergoes mutation as described above for both populations. The children are then combined with the parents to create airline and passenger populations with size $2n_a$ and $2n_p$, respectively.

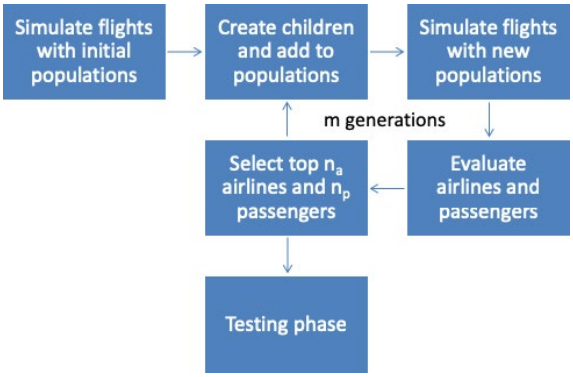

Figure 2: Depiction of the two-point crossover operator used for the passenger population.

With the extended populations, the algorithm simulates the overbooking procedure. This is done for each airline against the population of passengers. For each flight, the degree of overbooking is between two and seven seats, determined uniformly at random. To increase the opportunities of each passenger to accept an offer, the passenger population is divided into disjoint subsets with discrete sizes in $[10, 20, 50, 100]$. A flight is simulated for each subset of passengers.

During a simulated flight, the order of the passenger subset is randomized. Then each passenger is given an opportunity to accept an offer until all offers are accepted. As described in Section 3.1, a passenger's decision is determined by indexing into its genome using the current offer amount and the number of offers remaining. To introduce a degree of noise, or variability, a passenger will decline an offer with probability 0.05 even when its genome indicates accept. Within a generation, each passenger participates in k flights against each airline.

After all flights are complete, the populations are evaluated according to their respective optimization objectives based only on flights during the current generation. This ensures that individuals are evaluated according to their current state rather than continuing to consider the performance of their less evolved selves. With only one objective, the passenger population is sorted in decreasing order. The airline population is sorted using the non-dominated sorting algorithm outlined in Section 2. The top n_a airlines and top n_p passengers are retained and constitute the parents in the next generation. The remaining members are discarded. The algorithm is depicted in Figure 2.

Our algorithm is implemented in Python using the Distributed Evolutionary Algorithm in Python (DEAP) library [5]. DEAP includes implementations of common crossover, mutation, and selection operators.

4 Experimental Design

One of the hallmarks of genetic algorithms is the large number of parameters. Table 2 shows the parameters in this work and the values used. We include results for eight runs. Each run uses one of the two values for number of generations with one of the four values for the crossover and mutation parameter η. Runs 0, 2, 4, 6 evolve for 50 generations while

runs 1, 3, 5, 7 evolve for 100 generations. For runs 0 and 1, $\eta = 7$; for runs 2 and 3, $\eta = 3$; for runs 4 and 5, $\eta = 10$; and for runs 6 and 7, $\eta = 5$.
Each run consists of 18 trials using the same parameter values. During a trial, evolution occurs for a number of generations, determined by the parameter value, to train the populations followed by a testing phase that consists of 2500 flights. The populations are then evaluated and sorted.

Parameter	Values
Airline crossover prob	0.9
Airline crossover η	[3, 5, 7, 10]
Passenger crossover prob	0.9
Airline mutation η	[3, 5, 7, 10]
Passenger mutation prob	$\frac{2.0}{\text{genome size}}$
Passenger random reject prob	0.05
Airline population size	50
Passenger population size	50
Passenger random subpopulation size	[10, 20, 50, 100]
Flights per generation	100
Generations	[50, 100]

Table 2: Parameter values used in our experiments.

5 Results

The baseline for evolved passengers is a population that evolved against a single, static airline strategy. That strategy consists of this sequence of payments: [200, 400, 750, 1200, 2000, 3000, 4800, 6400]. Not surprisingly, passenger results when evolved against this fixes strategy are better than those achieved against an evolving airline population. This is due to the passengers' ability to develop strategies while the airlines are unable to counter those strategies through their own evolution. Figure 3 illustrates the effect of this advantage. Allowing the airlines to "fight back," so to speak, makes a significant difference.
We do not have an analogous baseline for airlines. The reason is that while it is relatively straightforward to devise a reasonable static airline strategy, it is much more difficult to create such a strategy for a passenger against which to evolve the airlines. Thus, airlines are evolved only in competition with a population of evolving passengers.
We find that more generations during evolution favors the airlines. This is likely due to the significantly smaller airline genome. Recall that the airline genome consists of 8 real-valued numbers, each representing an incentive value, while the passenger genome consists of 256 bits, each representing a decision of accept or decline. We believe that with only 8 genes, airlines are able to more quickly converge on good genome values.
Figure 4 shows passenger and airline incentive values for all runs. Within a run, data for the 18 trials are averaged. While there is little difference between Run 0 (50 generations)

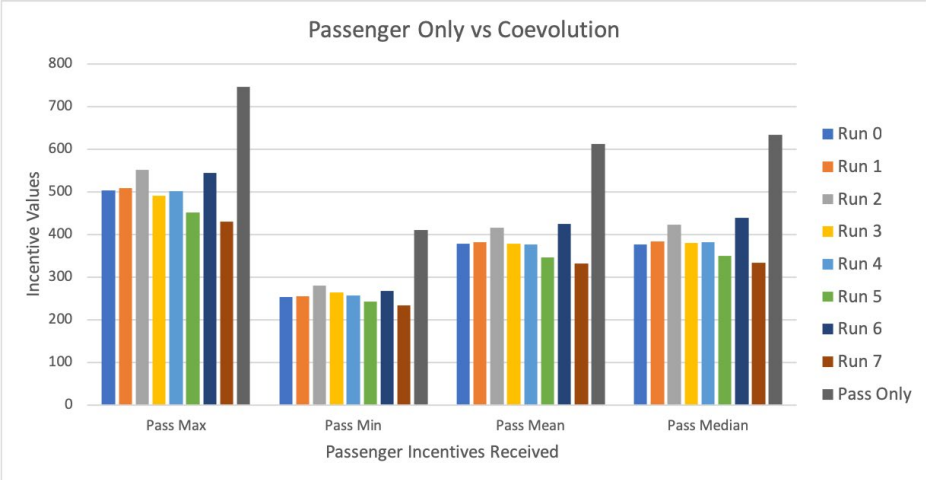

Figure 3: Results for passenger-only evolution against a fixed airline strategy vs coevolution. Each bar represents the mean of 18 trials. The 8 coevolutionary runs represent the various parameter settings described in Section 4.

and Run 1 (100 generations), the other three pairs of runs show significant benefit for the airlines in the 100 generation cases. We also note that the mean and median values are quite similar, suggesting that the effects of outliers is small. In addition, it appears that there is not a clear best value for η. The largest values, 7 and 10, appear best for 50 generations but 5 is best for 100 generations. We can conclude only that this value is not of primary importance.

Figures 5 and 6 show all 18 trials for passengers and airlines, respectively, for Run 4. All trials within a run are performed with the same parameter values. Of course, higher values are better for the passengers while lower values benefit the airlines. Run 4 was chosen arbitrarily as representative of all runs, as space does not allow showing all runs.

The range constraints placed on values in the airline genome are broad and overlapping. Despite this we find that airlines evolve sequences of offers that are sensible in nearly all cases. That is, the values are monotonically increasing. It makes little sense for incentive offers to decrease during the process of finding volunteers. Though our constraints do not enforce this condition, evolution selects out members that perform poorly, as non-increasing sequences would. We note that most instances of decreasing values occur near the end of the genome. This is due to the fact that most offers are accepted in earlier rounds. Thus, values later in the sequence have less impact on the fitness of an individual because they are not often used – any penalty incurred from their use is not incurred frequently. Table 3 show a selection of evolved genomes from one trial. They include the best and worst performing airlines.

6 Conclusions and Future Work

In this work, we use genetic algorithms to evolve strategies for passengers and airlines for the problem of using incentives to avoid involuntary denied boarding due to overbooking.

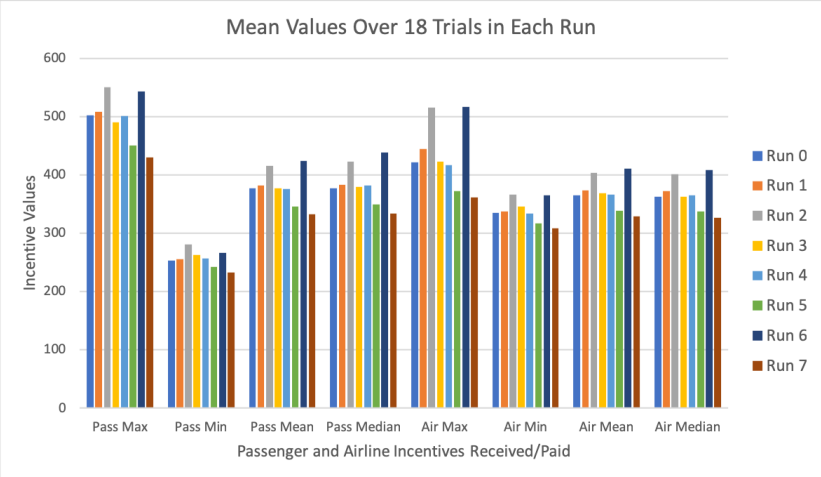

Figure 4: Passenger and airline results for all coevolutionary runs. In most cases, we see benefit for the airlines when evolution occurs for 100 generations (runs 1, 3, 5, 7) vs for 50 generations (runs 0, 2, 4, 6).

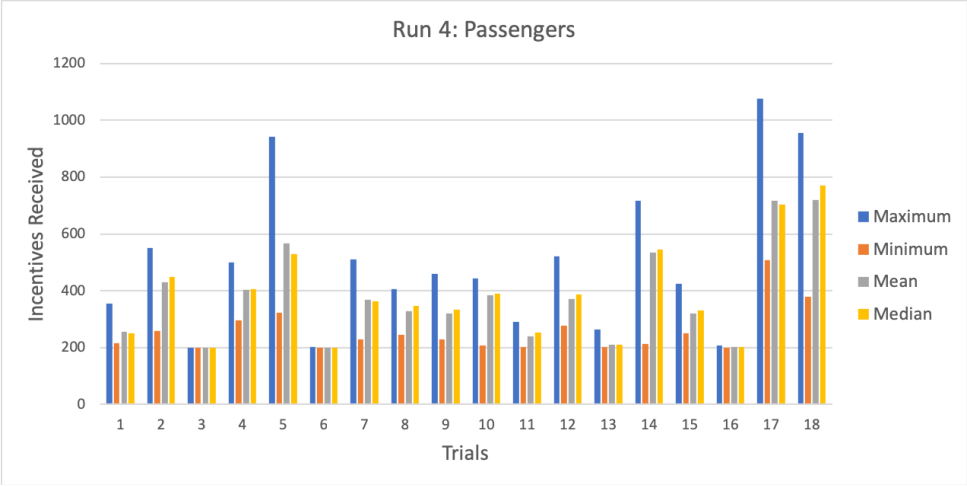

Figure 5: Passenger results for one coevolutionary run consisting of 18 trials.

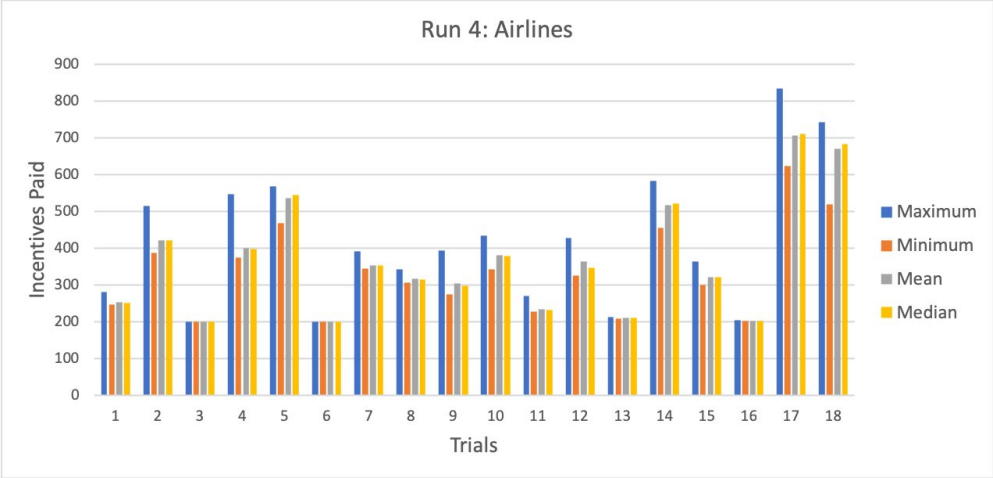

Figure 6: Airline results for one coevolutionary run consisting of 18 trials.

Avg Incentive	Genome
366.26	[200, 410, 717, 1039, 2516, 3178, 3993, 5192]
384.13	[200, 402, 1044, 1208, 2761, 4422, 3672, 5259]
391.90	[200, 532, 873, 1061, 2012, 3652, 4478, 5343]
424.44	[200, 685, 944, 1301, 1658, 4564, 5239, 5207]
499.36	[200, 400, 1554, 2521, 2678, 4361, 3617, 5227]

Table 3: A mix of airline genomes from one trial.

Our algorithms use competitive coevolution, in which each population evolves against the other. This work is a proof of concept, in which we demonstrate that it is possible to evolve such strategies. However, to be truly useful would require additional domain-specific data, such as typical degree of overbooking, number of resultant denied boarding events, and average incentives paid by the airlines. Ideally, this information would be used on a per-flight basis to develop strategies specific to each flight, including factors such as day of the week and time of year. Unfortunately, this information is difficult to find, at best. Airlines are required to report some information to the Department of Transportation but not all that is needed.

There is considerable opportunity for continued experimentation. There are many parameters that might be further tuned. For example, we experimented with several values for η but within a small range. Perhaps larger values would produce different results. In addition, it might be beneficial to create a different representation for the airline genome. A larger number of genes might help ensure greater diversity in the population, thus helping avoid early convergence.

It is also worth noting that our experiments are predicated on the model commonly used by airlines for handling overbooking: incentive offers made to passengers in real-time. Perhaps an entirely different model would be better for one or both populations. For example, passengers could place bids in advance of their flight, possibly even at the time of booking.

Their bid would represent the incentive they would be willing to accept in the event that volunteers are needed. Some airlines currently use a similar system for upgrades. At the time of booking, passengers can place a bid – the price they are willing to pay – to upgrade to a higher class of service. This can be mutually beneficial. It allows passengers the chance of getting an inexpensive upgrade and it helps the airline avoid unclaimed upgrade opportunities that occur when insufficiently many passengers are willing to pay the fixed upgrade fee.

References

[1] CHATWIN, R. E. Continuous-time airline overbooking with time-dependent fares and refunds. *Transporation Science 33*, 2 (May 1999).

[2] DEB, K., AND AGRAWAL, R. B. Simulated binary crossover for continuous search space. *Complex Systems 9*, 2 (1995).

[3] DEB, K., AND DEB, D. Analyzing mutation schemes for real-parameter genetic algorithms. Tech. Rep. 2012016, Indian Institute of Technology Kanpur, Kanpur, PIN 208016, India, 2012.

[4] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation 6*, 2 (2002).

[5] FORTIN, F.-A., RAINVILLE, F.-M. D., GARDNER, M.-A., PARIZEAU, M., AND GAGNÉ, C. Deap: Evolutionary algorithms made easy. *Journal of Machine Learning Research 13* (July 2012).

[6] GARROW, L. A., KRESSNER, J., AND MUMBOWER, S. Is increasing airline denied boarding compensation limits the answer? factors that contribute to denied boardings. *Journal of Air Transport Management 17*, 5 (September 2011).

[7] HILLIS, W. D. Co-evolving parasites improve simulated evolution in an optimization procedure. *Physica D: Nonlinear Phenomena 42*, 1-3 (June 1990).

[8] KLOPHAUS, R., AND POLT, S. Airline overbooking with dynamic spoilage costs. *Journal of Revenue and Pricing Management 6*, 1 (March 2007).

[9] LINDENMEIER, J., AND TSCHEULIN, D. K. The effects of inventory control and denied boarding on customer satisfaction: The case of capacity-based airline revenue management. *Tourism Management 29*, 1 (February 2008).

[10] OF TRANSPORTATION BUREAU OF TRANSPORTATION STATISTICS, U. D. May 2018 u.s. airline traffic data. https://www.bts.gov/newsroom/may-2018-us-airline-traffic-data, August 2018.

[11] ROTHSTEIN, M. An airline overbooking model. *Transportation Science 5*, 2 (May 1971).

[12] SUBRAMANIAN, J., STIDHAM, S., AND LAUTENBACHER, C. J. Airline yield management with overbooking, cancellations, and no-shows. *Transporation Science 33*, 2 (May 1999).

[13] SUZUKI, Y. An empirical analysis of the optimal overbooking policies for us major airlines. *Transportation Research Part E: Logistics and Transportation Review 38*, 2 (April 2002).

[14] V. WANGENHEIM, F., AND BAYON, T. Behavioral consequences of overbooking service capacity. *Journal of Marketing 71* (October 2007).