A Game-Theoretic Analysis of Reallocation Mechanisms for Airport Landing Slots

Jackie Baek and Hamsa Balakrishnan

***Abstract*—As airport arrival capacities increasingly constrain the air transportation system, there is a need for mechanisms by which airlines can exchange landing slots amongst each other. We analyze two such mechanisms, scaled airline preferences and two-for-two trades, from a game-theoretic perspective. This paper investigates the extent to which strategic behavior on part of the airlines can impact the performance of each mechanism. In addition to increasing system efficiency, the reallocation mechanisms should exhibit desirable fairness and incentive properties, notions that we formally investigate in this paper. We show that neither mechanism has good incentive properties, and we develop simple, non-truthful strategies that airlines can use. Our empirical results show that for the scaled airline preferences mechanism, the best performing strategy depends greatly on the extent to which fairness is enforced. For the two-for-two trades mechanism, a simple threshold strategy can yield significant cost savings relative to the best-response strategy, and system efficiency increases when all airlines use the threshold strategy in equilibrium.**

***Index Terms*—Air traffic management, game theory, mechanism design, optimization, algorithms**

I. Introduction

One of the most constrained resources in the air transportation system is airport landing capacity. When the number of aircraft arriving at an airport at some period of time is projected to exceed the airport's landing capacity, a Ground Delay Program (GDP) is issued by the Federal Aviation Administration (FAA). The objective of a GDP is to delay the departures of flights such that they arrive at the destination airport at a rate that matches its forecast capacity. These delays are assigned to flights in a first-scheduled-first-served manner.

However, because different flights have different delay costs, flights can be assigned to landing slots more efficiently if flights with high delay costs were prioritized. Unfortunately, the FAA does not have access to any flight-specific cost information, as this is private information to the airlines. Yet, there could be a way in which the FAA and the airlines work together to allocate delays to flights in a way that improves system efficiency. To do this, there must be some predetermined mechanism which provide a particular structure on this process. We refer to these mechanisms as **reallocation mechanisms**. We assume that slots are initially allocated to flights using the first-scheduled-first-served rule, and then the slots are exchanged between the airlines using a reallocation mechanism.

This work was supported by the National Science Foundation, through NSF CPS grant numbers 1239054 and 1739505.

J. Baek is with the Operations Research Center, MIT. baek@mit.edu.

H. Balakrishnan is with the Department of Aeronautics and Astronautics, MIT. hamsa@mit.edu.

In this paper, we investigate reallocation mechanisms from a game-theoretic perspective, evaluating them not only on efficiency, but other desirable properties such as incentives and fairness. As each airline that would participate in a reallocation mechanism is a selfish agent (i.e. their goal is to maximize their own utility), their incentives are not necessarily aligned with the incentives of the FAA. Therefore, one cannot assume that the airline will perfectly cooperate in these mechanisms. In this case, a game-theoretic analysis of the mechanism is necessary to evaluate how much strategic behavior can impact the system.

We focus on two reallocation mechanisms that have been previously proposed: scaled airline preferences (SAP) [1], and two-for-two trades [2]. The SAP mechanism scales the flight's delay costs to ensure equity among the airlines. The two-for-two trades mechanism allows an airline to make trade offers to reduce the delay of a flight it values more, in return for the delay increase of a less valuable flight. We first set up a generic framework for reallocation mechanisms, and then we evaluate these two mechanisms in this framework. We run simulations of these mechanisms using historical data, and our results show that the impact of the mechanisms on both system and airline efficiency can drastically change when airlines behave strategically.

A. Current Ground Delay Program Operations

An airport's **capacity**, or its **airport acceptance rate (AAR)**, is the maximum number of aircraft that can safely land during each hour. When a reduction in future capacity for an airport is anticipated, the FAA issues a ground delay program with the following information: the time period in which the GDP is in effect, and the hourly AARs for this time period. Usually, the GDP is initialized a couple hours before its start time. For example, suppose there is a forecast for a snowstorm in Boston starting at 1pm. At 10am, the FAA issues a GDP effective at San Francisco International Airport (SFO) for 1–4pm with AARs of 35, 35, 40. This means that the landing capacity of the airport is 35 per hour for 1–3pm, and 40 for 3–4pm. Then, the flights that were originally scheduled to land in SFO between 1pm and 4pm are given new arrival slots to match the announced capacities. The slots are allocated using a first-scheduled-first-served rule called **ration-by-schedule (RBS)**. The RBS rule is currently accepted as a fair allocation method.

After RBS outputs an intermediate schedule, we consider any further changes the schedule to be a reallocation mechanism. Although RBS allocates arrival slots to specific flights, airlines can reassign the slots among the flights that they own

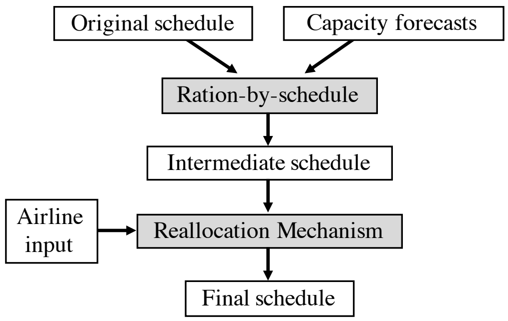


Fig. 1: The framework for GDP operations.

to optimize over their own costs, a method known as **intra-airline substitution**. If, however, an airline decides that it cannot use one of its slots due to a flight cancellation, the airline can report this to the FAA. Two mechanisms called **compression** and **slot-credit substitution** are currently in place to facilitate flight cancellations. Given a flight cancellation, the mechanisms redistribute this slot in a way that gives preference to flights of the airline who released the slot, which gives incentive for the airline to report their cancelled flights. This GDP framework is summarized in Fig. 1.

In practice, a GDP can get modified or cancelled during its operation once there are updates to weather forecasts. For simplicity, we assume that GDPs do not get updated. Additionally, we assume in this work that there is only one reallocation mechanism, and it is run exactly once. If there were multiple mechanisms or repeated runs of the same mechanism, an airline's strategy can change when taking the subsequent mechanisms into account. The impact of having multiple rounds of allocation is a direction for future research.

B. Need for Inter-Airline Reallocation

Assuming that flight cancellations are dealt with using compression and slot-credit substitution, the only reallocation mechanism currently in place for non-cancelled flights is intra-airline substitution. Because flights cannot depart earlier than their original departure time, reallocating flights within the same airline is oftentimes infeasible, especially for smaller airlines. Hence, there is an opportunity to decrease costs further using a mechanism that allows slots to be exchanged *between* airlines.

The main reason that designing such a mechanism is non-trivial is due to incentives. Flight-specific costs are private information to the airlines, and there is no reason for airlines to disclose any such information unless doing so will benefit them. If the FAA simply asked the airlines for their flights' delay costs and reallocated slots in a way that minimizes total delay costs, flights with high delay costs will benefit at the expense of delay increases for lower cost flights. Then, airlines have an incentive to misreport the costs of their flights to be higher than they are.

Other than incentives, there are other properties that an ideal mechanism should have — these will be listed and discussed in Section II-B. It may be that enforcing such desirable properties will come at a cost of a reduction in total efficiency gains.

C. Related Works

A method of scheduling aircraft using airline preferences was proposed in [1]. This work shows that this method can achieve significant cost savings for all airlines using empirical tests on European data. In our work, we modify and simplify this model into a reallocation mechanism, which we name the scaled airline preferences (SAP) mechanism.

The two-for-two trades mechanism, initially proposed in [2], provides an integer programming formulation that the central decision maker solves for the optimal allocation. Through experiments using historical data and simple airline strategies, the authors show that delay costs could be reduced significantly. The two-for-two mechanism was also incorporated in a separate model for air traffic flow management with fairness and collaboration [3]. Reallocation mechanisms in the presence of vacant slots have also been the focus of prior research [4], [5], [6], [7].

The impact of allocation mechanisms on the airlines is a practical concern, and has motivated the study of inter-airline equity in the context of air traffic management [8], [1], [9], [10]. The use of a combinatorial auction for airport landing slot allocation was first proposed in [11]. [12] provides an overview on the need for auctions in various aspects of air traffic management. Reallocation mechanisms with side payments between the airlines have also been investigated [5], [13].

Other than efficiency and incentive compatibility, there are other desirable properties in a general mechanism. Individual rationality is the property that no players will incur a loss by participating in the mechanism, and budget balanced is the property that the central decision maker should not lose money from the mechanism. The impossibility result of [14] demonstrates that no exchange can be efficient, budget-balanced, incentive compatible, and individual rational. Incentives in the context of airport landing slots has been studied [15], in which it was shown that any reallocation mechanism that is individual rational and Pareto-efficient can be manipulated in certain ways.

D. Contributions

In this paper, we formally model a reallocation mechanism as a game of incomplete information, and we analyze the SAP and two-for-two trades mechanism in this framework. We show that neither mechanism has good incentive properties.

For the SAP mechanism, we develop the *inflation* strategy, a strategy in which airlines either inflate or deflate the cost differences between their flights. We empirically show that the inflation strategy can perform better than the truthful strategy, but whether to deflate or inflate one's costs depends on whether or not fairness is enforced.

For the two-for-two trades mechanism, our results show that fairness does not have as large of an impact on this mechanism compared to the SAP mechanism. We develop the *naive* and *threshold* strategies, two simple strategies that determine which trades an airline should offer. We develop a branch and bound algorithm to compute the best response strategy. We show that the threshold strategy can achieve more than 70% of the cost savings as the best-response strategy, and system efficiency can increase when all airlines play the threshold strategy in equilibrium.

The rest of the paper is structured as follows. In Section II, we set up the framework for reallocation mechanisms and present desirable properties. In Section III, we analyze the SAP mechanism. In Section IV, we analyze the two-for-two trades mechanism, and we develop an algorithm to calculate the best response strategy in Section V. In Section VI, we run computational experiments of the mechanisms on historical data using various strategic assumptions on the airlines.

II. Reallocation Mechanism Framework

A. Formal Model

We define a framework for a reallocation mechanism, which we model as a game of incomplete information. Given an initial assignment of flights to arrival times, a reallocation mechanism takes input from the airlines and outputs a new assignment. Different mechanisms differ in the possible actions that airlines can take, and how the mechanism maps these actions to the final allocation of flights to arrival times.

Let $I = \{1, 2, \ldots, N\}$ be the set of airlines, F the set of all flights, F_i the set of flights for airline i, and T the set of all possible arrival times. Each flight f has a scheduled arrival time, arr_f, which is the earliest feasible time that the flight can land. Let $\phi_0 : F \to T$ be the initial allocation of flights to arrival times. Time is discretized (e.g. into 15 minute intervals), and more than one flight can be assigned to the same time interval. However, we assume that the number of flights assigned to each time does not change from the initial allocation. In other words, $\phi : F \to T$ is a **feasible** allocation if $|\{f : \phi(f) = t\}| = |\{f : \phi_0(f) = t\}| \ \forall t \in T$. Let Φ be the set of all feasible allocations.

Every airline i has a delay cost function, $d_i : \Phi \to \mathbb{R}$, which maps an allocation to the airline's delay costs from that allocation. The **utility** gained by an airline is defined as the reduction in delay costs. The terms "utility" and "cost savings" are used interchangeably. We assume that a flight cannot land before its scheduled arrival time, and this is modelled by assuming a delay cost of $+\infty$ for such an allocation.

An airline i takes an action within their strategy set S_i, which may be different for different mechanisms. A **direct** mechanism is a mechanism in which the action is to simply reveal their delay cost function d_i. In this case, S_i refers to the set of all possible delay cost functions that airline i can reveal. An **indirect** mechanism may have an action space that is completely different than the airline's delay cost function; for example, in the two-for-two trades mechanism, an action corresponds to a set of trade offers. We denote by $S_{-i} := \{(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_N) \mid s_j \in S_j, j \neq i\}$ the set of all possible strategy profiles of all airlines except i.

Lastly, the mechanism is defined by the allocation function $\mathbf{A} : S_1 \times \cdots \times S_N \to \Phi$, which takes all airlines' strategies as input, and outputs the resulting allocation. We overload the delay cost function d_i to also be able to take the strategies of all players as input, where $d_i(s_1, \ldots, s_N) = d_i(\mathbf{A}(s_1, \ldots, s_N))$.

Formally, a reallocation mechanism is defined as $(I, F, arr, T, \phi_0, d, S, \mathbf{A})$ of airlines, flights, scheduled arrival times for each flight, discretized times, an initial allocation, delay cost functions for each airline, strategy sets for each airline, and an allocation function. Each airline knows its own delay cost function, but not those of other airlines. All other information is assumed to be known to all players.

We make the following assumptions regarding the structure of the delay costs:

1) The delay cost for each airline is a sum of delay costs for each of its flights, each of which are independent. Specifically, for an airline i, it incurs a cost $d_i(f, t)$ when flight f lands at time t, and the airline's total delay cost for an allocation ϕ is $\sum_{f \in F_i} d_i(f, \phi(f))$. Then, in a direct mechanism, an airline can simply reveal $d_i(f, t)$ for every f and t, rather than $d_i(\phi)$ for every allocation ϕ. This assumption implies that an airline's delay cost does not depend on the flights of other airlines.
2) In our computational experiments, we assume that delay costs are linear in delay time. In this case, the delay cost function of each flight is characterized by just one number, the cost incurred for each unit of delay time. We refer to this as the **unit delay cost**. Specifically, if the unit delay cost of a flight f is α, then the delay cost function is:

$$d_i(f, t) = \begin{cases} +\infty & t < arr_f \\ \alpha n_f(t) & t \geq arr_f, \end{cases}$$

where $n_f(t)$ is the number of time steps from arr_f to t. We only use this assumption in our experiments (Section VI).

B. Desirable Properties of a Reallocation Mechanism

The main purpose of a reallocation mechanism is to increase efficiency (reduce total cost); however, there are other properties that we also desire in a reallocation mechanism. We focus on three such properties in this paper: individual rationality, incentive compatibility, and fairness.

1) Individual Rationality (IR): A reallocation mechanism is individually rational if no airline becomes worse off from the mechanism. Formally, a direct mechanism is individually rational if $d_i(d_i, s_{-i}) \leq d_i(\phi_0)$ for all $s_{-i} \in S_{-i}$. That is, if an airline i takes the action of revealing its true cost function, d_i, then its overall delay costs do not increase after the mechanism. We define an indirect mechanism to be individually rational if for any delay cost function d_i, there exists a strategy s_i such that $d_i(s_i, s_{-i}) \leq d_i(\phi_0)$ for all $s_{-i} \in S_{-i}$.

This property is clearly desirable since it gives airlines the guarantee that it can only benefit them. However, enforcing IR will come at the cost of efficiency. Because flight delay costs can vary widely between airlines, the allocation which simply minimizes total cost will be the one which increases delays for airlines with low delay costs and decreases delays for airlines with high delay costs. Such an allocation will be infeasible if IR is enforced.

2) Incentive Compatibility (IC): A reallocation mechanism is incentive compatible if the best strategy for each airline is to act truthfully. No airline should be able to lie or misrepresent their preferences to "game the system". A direct mechanism is **dominant strategy incentive compatible (DSIC)** if revealing their true delay costs gives them the highest payoff, regardless of the actions of other players. Formally, a direct mechanism is DSIC if $d_i(d_i, s_{-i}) \leq d_i(s_i, s_{-i})$ for any $s_i \in S_i$ and $s_{-i} \in S_{-i}$.

For an indirect mechanism, incentive compatibility is not as well-defined since there may not be a notion of a "truthful" strategy. However, by the revelation principle, the existence of a dominant strategy is equivalent to being DSIC. For an airline i with cost function d_i, s_i^* is a **dominant strategy** if $d_i(s_i^*, s_{-i}) \leq d_i(s_i, s_{-i})$ for any $s_i \in S_i$ and $s_{-i} \in S_{-i}$. If every airline has a dominant strategy, the mechanism can be converted into a direct mechanism that is DSIC. Hence, for indirect mechanisms, checking for incentive compatibility is equivalent to checking for a dominant strategy.

3) Fairness: While IR and IC can be rigorously defined using game theoretic notions, fairness is not universally well-defined. At a high level, fairness is the notion that every airline should benefit by an equal amount. Ideally, one can define "benefit" as the reduction in delay cost that the mechanism provides to each airline. However, the problem with this is that the mechanism does not know the airlines' true delay cost functions; and if the airlines are asked to self-report them, then we again run into the problem of incentives. Therefore, we define fairness in a way that does not have any incentive issues:

Definition 1. *Given an initial allocation ϕ_0 and a final allocation ϕ, let $\Delta_i := \sum_{f \in F_i}(\phi(f) - \phi_0(f))$ be the net movement of airline i's flights. For $\lambda \geq 0$, the allocation ϕ is λ-**fair** if $|\Delta_i| \leq \lambda$ for all airlines i. If $\lambda = 0$, we say the allocation is **strictly fair**.*

That is, an allocation is strictly fair if, for every airline, their flights "move up" as much as their other flights "move down". One can think of time as a currency, and a flight reducing its delay must be "paid for" by other flights increasing their delay by the same amount. This notion of fairness is easy to verify, without running into the issue of incentives. By setting the parameter λ to be strictly greater than 0, we relax the fairness constraint. In Section VI-B, we study how the value of λ affects the outcomes of the mechanism.

III. Scaled Airline Preferences (SAP) Mechanism

In the SAP mechanism, each airline reports the delay cost function for each of its flights. These cost functions are then scaled such that each airline has the same average delay cost. The mechanism outputs an allocation which minimizes the scaled cost. This idea of scaling delay costs was proposed in [1], where this approach is used as a primary ground holding model. Since an airline's action is to reveal their delay costs, this mechanism is a direct mechanism.[1]

[1]It is possible for an airline to instead reveal *scaled* delay costs, but since it is equivalent to revealing true delay costs, we will consider this to be a direct mechanism.

The goal of the scaling step is to provide equity to all participating airlines by taking only the *relative* cost differences between flights of the same airline into account. If there was no scaling step, then every airline would misreport their delay costs to be higher than they are, so that their flights are prioritized over flights of other airlines. The scaling step disincentivizes this type of dishonesty; however, our analysis shows that there are other ways that airlines can manipulate this mechanism.

The scaling method that we consider is a discretized version of the one in [1], and it works as follows: For each flight $f \in F_i$, let $T_f = \{t \in T \mid t \geq arr_f\}$ be the set of feasible landing times. Each airline i reveals $d_i(f,t)$ for all $f \in F_i$ and $t \in T_f$, the delay cost incurred by the airline if flight f lands at time t. Then, α_i is the scaling factor for airline i, which satisfies:

$$\frac{1}{|F_i|} \sum_{f \in F_i} \frac{\sum_{t \in T_f} \alpha_i d_i(f,t)}{S_{|T_f|}} = 1, \tag{1}$$

where $S_n = \sum_{k=1}^{n} k$ denotes the sum of integers up to n. The scaled delay cost for flight f landing at time t is $\hat{d}_i(f,t) = \alpha_i \cdot d_i(f,t)$.

The intuition for (1) is as follows. Recall that $n_f(t)$ is the number of time steps from time t to arr_f. After the scaling step, the "average" flight satisfies $\hat{d}_i(f,t) = n_f(t)$ and $\sum_{t \in T_f} \hat{d}_i(f,t) = S_{|T_f|}$. The term $\sum_{t \in T_f} \hat{d}_i(f,t)$ is the "area under the curve" of the delay cost function, as shown in Fig. 2. Then, for any flight f, we normalize the term $\sum_{t \in T_f} \alpha_i d_i(f,t)$ by the same term for an average flight, which is $S_{|T_f|}$.

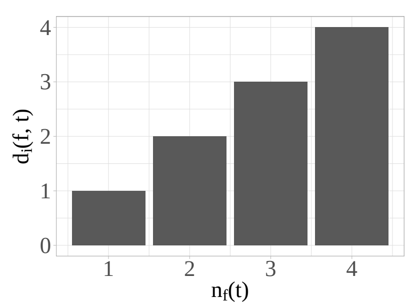


Fig. 2: The scaled delay cost function of an average flight. The "area under the curve" of this plot is $\sum_{t \in T_f} \hat{d}_i(f,t) = S_{|T_f|}$.

A. Implementation of the Allocation Function

The allocation function **A**, which chooses the allocation which minimizes the total scaled delay cost, is implemented as the following integer program with parameter λ:

$$\min \sum_{i \in I} \sum_{f \in F_i} \sum_{t \in T} \hat{d}_i(f,t) x_{ft} \tag{2}$$

$$\text{s.t.} \quad \sum_{t \in T} x_{ft} = 1 \quad \forall f \in F \tag{3}$$

$$\sum_{f \in F} x_{ft} = |\{f \mid \phi_0(f) = t\}| \quad \forall t \in T \tag{4}$$

$$\hat{d}_i(\phi_0) \geq \sum_{f \in F_i} \hat{d}_i(f,t) x_{ft} \quad \forall i \in I \tag{5}$$

$$\Delta_i = \sum_{f \in F_i} \sum_{t \in T} (t - \phi_0(f)) x_{ft} \quad \forall i \in I \tag{6}$$

$$|\Delta_i| \leq \lambda \quad \forall i \in I \tag{7}$$

$$x_{ft} \in \{0,1\} \quad \forall f \in F, t \in T \tag{8}$$

x_{ft} is a binary variable which is 1 if and only if flight f is allocated to land at time t. The objective function (2) minimizes the total scaled cost. (3) ensures that all flights are assigned a landing time, and (4) ensures the resulting allocation is feasible. (5) enforces individual rationality, and (6) and (7) enforce λ-fairness.

B. Incentive Compatibility

Although we can enforce individual rationality and fairness using constraints in the allocation function, the same cannot be done for incentive compatibility. Unfortunately, there are counterexamples which show that this mechanism is not dominant strategy incentive compatible.

Theorem 1. *The SAP mechanism is not dominant strategy incentive compatible.*

Proof. Consider an instance where there are three airlines, A, B, and C, with two flights each. The initial allocation of the six flights to arrival slots is shown on the left of Fig. 3. The numbers on the flights represent the true unit delay cost of each flight. Suppose flight A_2's scheduled arrival time is 6:06, hence it cannot arrive before then. If all airlines report truthfully, the SAP mechanism outputs the allocation shown on the right. Airline C's cost reduction from the mechanism is $1.1 - 0.9 = 0.2$.

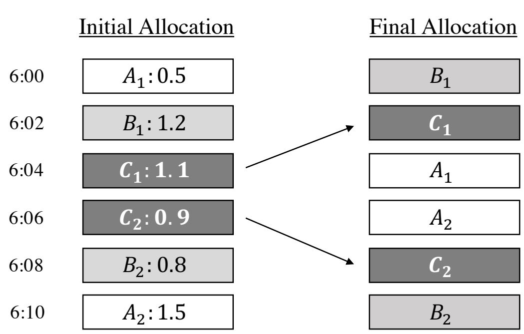


Fig. 3: The initial and final allocation if all airlines reported truthfully.

If airline C misreports its costs to be 1.3 and 0.7 for flights C_1 and C_2 respectively, as shown in Fig. 4, the SAP mechanism will output the allocation shown on the right. This is because the difference in costs of airline C's flights are now larger than the difference in costs of airline B's flights. This allocation gives airline C a cost reduction of

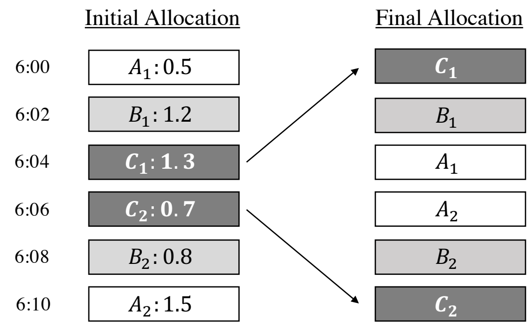


Fig. 4: The initial and final allocation if airline C misreports its costs, and the other airlines report truthfully.

$2(1.1) - 2(0.9) = 0.4$, which is a higher cost reduction than had it reported truthfully. Therefore, the SAP mechanism is not DSIC. □

C. Inflation Strategy

Since SAP is not incentive compatible, we investigate which other strategies perform better than truth-telling. From the counterexample in the proof of Theorem 1, we saw that an airline can manipulate its actions in response to the exact strategies of other airlines. However, it is not a practical assumption that an airline will know the strategies of all other airlines. Therefore, we develop a strategy in which an airline only uses information regarding their own flights, where the delay cost differences between flights are either inflated or deflated.

Definition 2. *The* ***inflation*** *strategy with rate $r \geq 0$ is to reveal the modified delay function $\hat{d}_i^r(f,t) := r(\hat{d}_i(f,t) - n_f(t)) + n_f(t)$.*

The inflation strategy modifies the delay costs such that the mean delay costs of all flights stays the same, but the difference between these costs are either inflated or deflated. A rate of $r > 1$ inflates the cost differences, whereas a rate of $r < 1$ deflates them. For example, suppose there are two flights with unit delay costs of 1.2 and 0.8. The inflation strategy with rate $r = 2$ will alter these unit delay costs to 1.4 and 0.6, whereas a rate of $r = 0.5$ will alter these unit delay costs to 1.1 and 0.9. A rate of $r = 0$ makes it so that all flights have the exact same delay cost, $\hat{d}_i(f,t) = n_f(t)$ for all f and t. This corresponds to a flight having a unit delay cost of 1.

In the simulations in Section VI-C, we show that the inflation strategy performs better than the truthful strategy for certain values of the inflation rate. However, the exact rate which maximizes cost savings both depends on the particular airline, and the value of λ which is used to enforce fairness.

IV. Two-for-Two Trades

The two-for-two trades mechanism allows airlines to decrease the delay of a high-valued flight in exchange for a delay increase of a low-valued flight. Each airline submits trade offers in the form $(f_d, t_d; f_u, t_u)$, which can be interpreted as: We are willing to move down flight f_d to time t_d in return for moving up flight f_u to time t_u.[2] f_u is the "valuable" flight which the airline wishes to reduce the delay of in exchange for the delay increase of flight f_d. Airlines give up two slots in return for two slots, hence the name two-for-two. Airlines submit as many offers of this form as they like, and the mechanism chooses an allocation which maximizes the number of accepted trades.

The action space of each player is the set of trade offers they can give. For airline i, let $O_i = \{(f_d, t_d; f_u, t_u) \mid f_d, f_u \in F_i, t_d > \phi_0(f_d), t_u < \phi_0(f_u)\}$ be the set of all trades which airline i can offer. Then, $S_i = 2^{O_i}$ is the set of all actions that airline i can take, where each action is a set of offers. If an offer $(f_d, t_d; f_u, t_u)$ is **accepted** (or executed), flight f_d is assigned slot t_d, and f_u is assigned t_u. From an initial allocation ϕ_0, a set of offers $O \subseteq \bigcup_i S_i$ is **feasible** if the acceptance of all offers in O results in a feasible allocation (i.e. the number of flights landing at each time period does not change).

We define the **utility** of an offer to be the reduction of delay cost if the offer is accepted. Formally, for offer $o = (f_d, t_d; f_u, t_u)$, we denote by its utility $u(o) = (d_i(f_d, \phi_0(f_d)) - d_i(f_d, t_d)) + (d_i(f_u, \phi_0(f_u)) - d_i(f_u, t_u))$.

Once all airlines submit their offers, the allocation function **A** outputs the allocation outputs an allocation that maximizes the number of accepted trades. Formally, **A** chooses a feasible set of offers $O^* \subseteq \bigcup_i S_i$ of maximum cardinality, and returns the allocation that results from accepting these offers. If there are multiple optimal solutions (i.e., multiple feasible offer sets of maximal cardinality), then we assume that one of the optimal solutions is picked uniformly at random. This assumption is revisited in Section V-A.

A. Implementation of the Allocation Function

The allocation function **A** is implemented as an integer program. Let $s_1, \ldots, s_n$ be the strategies of all airlines, and let $O = \bigcup_i s_i$ be the set of all submitted offers. Let $D = \bigcup_{(f_d,t_d;f_u,t_u)\in O}\{(f_d, t_d)\}$ be set of "downward movements" of all offers, and let $U = \bigcup_{(f_d,t_d;f_u,t_u)\in O}\{(f_u, t_u)\}$ be set of "upward movements" of all offers. For every flight f, let $T_f = \{t \mid (f, t) \in U \cup D\} \cup \{\phi_0(f)\}$ be the set of all possible landing times for f. The following variables are used:

- $x_{f,t} \in \{0, 1\}$ $\forall f \in F, t \in T$. $x_{f,t} = 1$ iff flight f is assigned to land at time t.
- $y_o \in \{0, 1\}$ $\forall o \in O$. $y_o = 1$ iff offer o is accepted.

Then, **A** is implemented using the following integer program, which has been slightly modified from [2]:

$$\max \sum_{o\in O} y_o \tag{9}$$

$$\text{s.t.} \quad \sum_{t\in T_f} x_{ft} = 1 \quad \forall f \in F \tag{10}$$

$$\sum_{f\in F} x_{ft} = |\{f \mid \phi_0(f) = t\}| \quad \forall t \in T \tag{11}$$

$$x_{f_d t_d} = \sum_{(f_d,t_d;f_u,t_u)\in O} y_{(f_d,t_d;f_u,t_u)} \quad \forall (f_d, t_d) \in D \tag{12}$$

$$x_{f_u t_u} = \sum_{(f_d,t_d;f_u,t_u)\in O} y_{(f_d,t_d;f_u,t_u)} \quad \forall (f_u, t_u) \in U \tag{13}$$

$$\Delta_i = \sum_{f\in F_i} \sum_{t\in T} (t - \phi_0(f))x_{ft} \quad \forall i \in I \tag{14}$$

$$|\Delta_i| \le \lambda \quad \forall i \in I \tag{15}$$

$$x_{ft} \in \{0, 1\} \quad \forall f \in F, t \in T \tag{16}$$

$$y_o \in \{0, 1\} \quad \forall o \in O \tag{17}$$

The objective function (9) maximizes the total number of accepted trades. (10) ensures that all flights are assigned to a feasible landing time, and (11) enforces that the resulting allocation is feasible. (12) and (13) ensure a flight lands at a particular time if and only if an offer corresponding to that flight and time is accepted. (14) and (15) enforce λ-fairness.

B. Incentive Compatibility

Since two-for-two trades is an indirect mechanism, to assess incentive compatibility, we check whether it has a dominant strategy. Unfortunately, we show that there are instances of this game in which a dominant strategy does not exist.

Theorem 2. *There exist instances of the two-for-two trades game in which a player has no dominant strategy.*

Proof. Consider the case when there are two airlines with three flights each, as shown in Fig. 5. Suppose that flight A_1 is very valuable, hence airline A would like A_1 to get the 6:00 slot. We consider two trade offers in which airline A can make in this scenario. Let $o_{A_1, A_2} = (A_2$, 6:06; A_1, 6:00$)$ and $o_{A_1, A_3} = (A_3$, 6:10; A_1, 6:00$)$, as shown in Fig. 5.

Suppose that airline A's unit delay costs for its flight A_1, A_2, and A_3 are 100, 99, and 10, respectively. This implies that the utility of offer o_{A_1,A_2} is 1, and the utility of offer o_{A_1,A_3} is 90. We show that airline A does not have a dominant strategy; in this game, the best strategy of airline A depends on airline B's strategy.

Because both o_{A_1,A_2} and o_{A_1,A_3} involve flight A_1, the trades are mutually exclusive. Intuitively, since offer o_{A_1,A_3} is 90 times more valuable than offer o_{A_1,A_2}, the airline may be better off by *only* submitting o_{A_1,A_3} — we will derive a contradiction using this idea.

We consider two cases of offers by airline B shown in Fig. 6. We proceed by contradiction, and suppose that a dominant strategy for airline A exists, denoted by s^*_A. s^*_A is a set of offers which maximizes airline A's utility in both case 1 and 2.

1) Case 1: Airline B gives one offer. This offer happens to be what is necessary for airline A's offer o_{A_1,A_2} to be executed.

[2]This is a slight modification from what is described in [2], which interprets an offer as f_d moving up to *at least* t_d, and f_u moving to *at most* t_u. This modification does not reduce the capabilities of the mechanism, since an offer of structure "at least, at most" can be represented as a finite number of offers of "exact" structure.

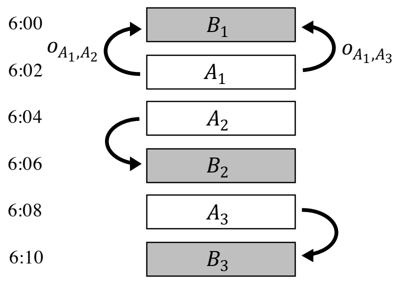


Fig. 5: There are two airlines, A and B, each with three flights. The initial slot allocation is as shown. The arrows on the left correspond to the offer o_{A_1,A_2}, and the arrows on the right correspond to o_{A_1,A_3}.

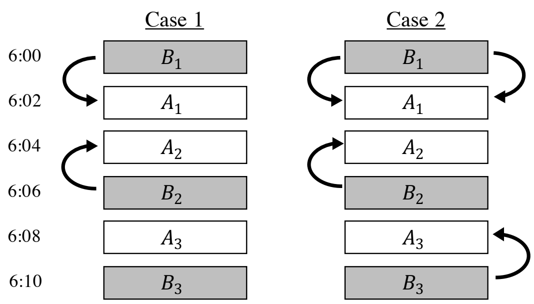


Fig. 6: Two cases of airline B's actions.

We claim that airline A must submit this offer to receive the highest utility.

Claim: $o_{A_1,A_2} \in s_A^*$.

From the perspective of airline B, there are two outcomes: their offer gets executed, or it does not. If the offer is not accepted, both airlines receive zero utility. Suppose it does get accepted. Then, flight B_1 takes slot 6:02 and B_2 takes 6:04. Then, there are three spots left for airline A's flights: 6:00, 6:06, and 6:08. The optimal way for airline A to assign flights to these times is to assign A_1 to 6:00, A_2 to 6:06, and A_3 to 6:08. This allocation will occur if and only if offer o_{A_1,A_2} is executed. Furthermore, this allocation is feasible if $s_A^* = \{o_{A_1,A_2}\}$. Therefore, it must be that $o_{A_1,A_2} \in s_A^*$.

2) Case 2: In this case, airline B submits two offers. Note that because each airline only has three flights, at most one trade will be executed for each airline. Therefore, for airline B, there are three outcomes: no trades are executed, o_{B_1,B_2} is executed, and o_{B_1,B_3} is executed. By considering all three of these cases and using an analogous argument to Case 1, the best case for airline A is that o_{A_1,A_3} gets accepted (and this is feasible if the airline only submits o_{A_1,A_3}). Therefore, airline A receives the highest utility when o_{A_1,A_3} is accepted, and hence it must be that $o_{A_1,A_3} \in s_A^*$.

From these two cases, we have shown that both o_{A_1,A_2} and o_{A_1,A_3} must be in the dominant strategy, s_A^*. However, including both of these offers is not optimal for case 2. When airline A submits both of these trades in case 2, there are multiple optimal solutions for the mechanism to pick from; it can either accept o_{A_1,A_2}, o_{A_1,A_3} (or possibly another offer that airline A submits). In such scenarios, we assume that the mechanism picks any optimal solution with equal probability. Then, there is a positive probability that o_{A_1,A_2} will be accepted; and in this case, the expected utility for airline A is strictly less than 90. However, if airline A submitted only o_{A_1,A_3}, their expected reward is exactly 90. Therefore, an action containing both o_{A_1,A_2} and o_{A_1,A_3} cannot be a dominant strategy. By contradiction, no dominant strategy exists. $\square$

It is not the case that *for every* instance of the two-for-two trades game that there is no dominant strategy. For example, if an airline only has one flight, their dominant strategy is to make no offers, since that is the only action in their strategy space.

C. Simple Strategies

Since searching for a dominant strategy is futile, we look for strategies that perform well in practice. We develop two simple strategies called *naive* and *threshold*. Both strategies are "simple", in that they only take into consideration information on the airline's own flights and their cost functions. They both only submit offers of positive utility, so that the airlines are still guaranteed individual rationality. Furthermore, the decision rules are easy to implement.

Definition 3. *The* ***naive*** *strategy for an airline i is to submit offers $\{o \in O_i \mid u(o) > 0\}$.*

In this strategy, an airline submits as many offers as it can, as long as the offers give a positive payoff. Since the airline does not know the actions of other airlines, intuitively, this strategy attempts to maximize the airline's total number of accepted trades. However, the drawback of this strategy arises from the fact that the airline's utility function is not a function of the *number* of their trades that are accepted, but of the utility of each accepted trade. If there is a high discrepancy between the utilities of their offers that are mutually exclusive, similar to the situation described in the proof of Theorem 2, it may not necessarily be better to submit *all* positively valued offers.

To combat this problem of mutually exclusive offers, instead of blindly throwing out all offers of positive value, one can submit the ones whose values exceed a certain threshold. This threshold is chosen not as an absolute number, but as a percentile relative to the utilities of other mutually exclusive offers. For an offer $o = (f_d, t_d; f_u, t_u)$ of airline i, let $\omega(o) \subseteq O_i$ be the set of offers with positive utility and are mutually exclusive with o. Offers o and o' are mutually exclusive if the offers involve a flight in common.

Definition 4. *The* ***threshold*** *strategy with threshold $p \in [0, 1]$ is as follows: For every possible offer $o \in O_i$, submit trade o if and only if $u(o)$ is above the p-th percentile of $\{u(o') : o' \in \omega(o)\}$.*

This strategy depends on a parameter p, and we will see how the performance of this strategy changes as p is changed. A high value of p implies that the airline is more selective about the offers that they provide. $p = 0$ corresponds to submitting all positive utility offers, which corresponds to the naive strategy.

Note that both the naive and threshold strategies do not have any performance guarantees. It is easy to construct an example in which there exists a strategy which provides positive utility, but both the naive and threshold strategies give zero utility. (In such an example, the airline may have to offer a trade of negative utility.) However, the simulations on real world data in Section VI-D suggest that these strategies perform well on practical instances.

V. Best Response for the Two-For-Two Mechanism

While an airline may not necessarily have a dominant strategy in the two-for-two trades mechanism, given the strategies of other airlines, a "best" strategy always exists. In this section, we develop an algorithm to calculate this best response strategy, using a branch and bound algorithm. We note that in practice, airlines will not know the exact actions of the other airlines. However, the best response strategy gives a *lower bound* on the delay costs the airline will incur from this mechanism. We use this lower bound to benchmark the performance of the naive and threshold strategies. Additionally, the best response strategy can give insight into what makes a good strategy.

A. Definition of Best Response

We slightly alter the definition of best response due to the existence of multiple optimal solutions. The standard definition of the best response strategy is defined as the strategy that gives the lowest delay costs *in expectation* over all possible optimal solutions. This definition works in theory, but is difficult to work with in practice since optimization solvers only return one solution, and the chosen solution is not necessarily drawn uniformly at random. Therefore, we define a best response strategy to be a strategy in which there exists *an* optimal solution which emits a delay cost lower than any other strategy.

Let $\Psi(s, s_{-i}) \subseteq \Phi$ be the set of all optimal allocations of $\mathbf{A}(s, s_{-i})$.

Definition 5. *Given the actions of other airlines s_{-i}, $s_{BR} \in S_i$ is a* ***best response*** *strategy for airline i if there exists an optimal allocation $\phi^* \in \Psi(s, s_{-i})$ such that for all other strategies $s' \in S_i$, $d_i(\phi^*) \leq d_i(\phi')$ $\forall \phi' \in \Psi(s', s_{-i})$. The* ***best response cost*** *is $d_i(\phi^*)$. s_{BR} is a* ***minimal best response*** *strategy if it is a best response strategy, and no proper subset of it is a best response strategy.*

In other words, the best response cost is the lowest cost that the airline would incur assuming that when there are multiple optimal solutions, the solver always picks the solution that is the most favorable for this airline. Hence, this definition of the best response cost gives an *lower bound* of the standard definition of the best response cost. Since our main purpose in calculating the best response is to use it as an lower bound comparison for other strategies, this definition actually gives a stricter comparison.

If a strategy s_{BR} is a *minimal* best response, then it means that all of its offers were accepted by the mechanism. If there was an unaccepted offer, we could simply remove it.

We define the best response problem to be the problem of finding the minimal best response for a particular airline in a two-for-two game and the other airlines' strategies. An instance of a best response problem for airline i is (G, s_{-i}), where G is the two-for-two game, and s_{-i} is a vector of strategies of the other airlines.

Since each airline has a finite number of flights and time is discretized, the strategy space of each airline is also finite. Therefore, one could find the best response strategy by enumerating all possible strategies. However, the number of possible strategies is $2^{|O_i|}$, where O_i is the set of feasible offers for airline i. In a typical practical example, the number of feasible offers is in the order of thousands, making the enumeration of strategies intractable. Therefore, we develop an algorithm to find the best response which uses heuristics to decide which strategies to check. This algorithm is not theoretically guaranteed to be faster than the brute-force search in the worst case. However, we see that the algorithm performs well on practical instances.

B. Modification of the Allocation Function

Before we describe the algorithm for the best response, we describe a couple modifications of the allocation function, $\mathbf{A}$, implemented as an IP formulation as described in Section IV-A. These modifications will be the main tools that the best response algorithm uses. Recall that $\mathbf{A}$ takes the offers from all airlines as input, and outputs the allocation which maximizes the number of trades.

The best response algorithm modifies the allocation function by taking advantage of three facts about this IP formulation:

1) It takes on the order of seconds to complete.
2) The objective function can be changed.
3) Constraints concerning the execution of certain offers can be added.

Our algorithm runs $\mathbf{A}$ several times at each iteration, so it must be able to run quickly. We sometimes change the objective function of $\mathbf{A}$ to be to minimize the delay cost of an airline, d_i, rather than to maximize the total number of trades. Each iteration of the algorithm has a restricted solution space to search for the optimal solution, which corresponds to adding constraints regarding the offers to $\mathbf{A}$. Given a set of offers O, there are three types of constraints that we may add:

1) *all*(O): All offers in O must be accepted. Corresponds to the constraint $\sum_{o \in O} y_o = |O|$.
2) *at-least*(O): At least one offer in O must be accepted. Corresponds to the constraint $\sum_{o \in O} y_o \geq 1$.
3) *not-all*(O): At least one offer in O must *not* be accepted. Corresponds to the constraint $\sum_{o \in O} y_o \leq |O| - 1$.

More than one of the above constraints can be added.

Given an instance of the best response problem, (G, s_{-i}), denote $\hat{\mathbf{A}}_f^C : 2^{O_i} \to 2^{O_i}$ to be the allocation function which takes the airline i's offers as input, runs the two-for-two mechanism with the objective function of minimizing f subject to constraints C, and outputs the subset of airline i's offers that were accepted. Since the initial allocation, ϕ_0, and the strategies of other airlines, s_{-i}, are known,

$\hat{\mathbf{A}}_f^C$ can be computing by modifying the formulation of the original allocation function $\mathbf{A}$, using the three facts about the formulation $\mathbf{A}$ as stated previously. If the objective function is the default one, to maximize the number of trades, we will denote this by not specifying f; hence we simply write $\hat{\mathbf{A}}^C$. C is a set of constraints, where each constraint is in one of the three forms described above. An example of C is $\{\textit{all}(O), \textit{all}(O'), \textit{not-all}(O'')\}$, where O, O', O'' are sets of offers.

C. Best Response Algorithm

We develop an algorithm for the best response problem using a branch and bound approach, which is similar to a divide and conquer method. The algorithm generates a tree during its search, in which each node of the tree is allocated a subset of the solution space to search for the optimal solution. A subset of the solution space is characterized by a set of constraints.

A node is *processed* by running the bounding and branching stage. If a node's solution space is deemed to not contain the optimal solution (the *bound* stage), then the node is pruned. Otherwise, the node *branches* into two children nodes, where each child node has a further restriction on the solution space it searches on. Every constraint of a parent node is passed on to its children nodes, which implies that the solution space for a child node is a subset of the solution space of its parent node. We process all nodes in the tree until all leaf nodes are pruned. A high-level description of how a node is processed is shown in Fig. 7.

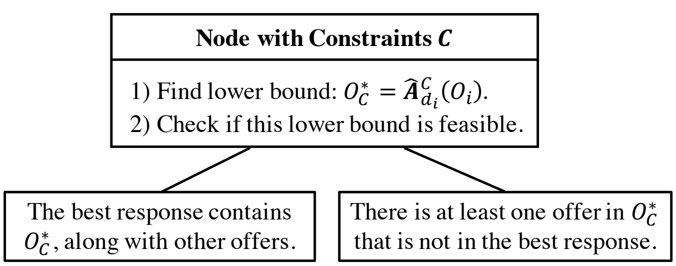


Fig. 7: An illustration of how a node in the branch and bound tree is processed.

The tree initially has one root node with no constraints, and the algorithm starts by processing this root node. We maintain a value B, initialized at $+\infty$, which represents the lowest delay cost found in a feasible solution found thus far. In other words, there exists a set of offers O for airline i such that $d_i(\mathbf{A}(O, s_{-i})) = B$.

1) Bounding Stage: A node has a set of constraints C that must be satisfied. Using these constraints, we calculate an lower bound and a feasible solution.

Lower Bound: We compute $O_C^* = \hat{\mathbf{A}}_{d_i}^C(O_i)$, which runs the modified allocation function whose objective function is to minimize airline i's costs, while satisfying constraints C. Airline i's input is O_i, the set of *all* possible trades for airline i. The modified allocation function returns a set of offers $O_C^* \subseteq O_i$, the set of airline i's offers that were accepted. Then $v_C^* = \sum_{o \in O_C^*} v(o)$ is the lowest delay cost that airline i can ever achieve with constraints C, which corresponds to the lower bound for this node. If $v_C^* \geq B$, we prune this node.

Feasible Solution: We use the set of accepted offers found in the lower bound, O_C^*, and use that as airline i's input to the original allocation function, $\phi = \mathbf{A}(O_C^*, s_{-i})$ (which maximizes the total number of accepted offers). If the allocation decides to accept all offers provided by the airline, O_C^*, then this is a feasible solution that is equal to the lower bound for this node. In this case, we prune this node. Otherwise, a strict subset of offers in O_C^* were accepted. If airline i's cost from this allocation is less than B, then we can update B to be this value.

This allocation ϕ is the one that maximizes the total number of trades; denote the total number of trades accepted by n^*. We check whether there exists another optimal solution (with n^* total accepted trades), where the set of accepted offers of airline i is exactly O_C^*. If one exists, it corresponds to a best response. To do this, we run $\hat{\mathbf{A}}^D(O_C^*)$, where $D = \{\textit{all}(O_C^*)\}$. That is, we run the allocation function which maximizes the total number of accepted trades, but with the constraint that all offers in O_C^* are accepted. Airline i submits only the trades O_C^*, so its delay cost from this allocation is exactly v_C^*. If the total number of trades executed from this allocation is equal to n^*, then there exists an optimal solution where exactly offers in O_C^* are accepted. By our definition, O_C^* is a best response (subject to constraints C), and we can prune this node.

If we have not pruned this node in this bounding stage, the node moves on to the branching stage.

2) Branching Stage: We branch the node into two child nodes. Each child node is further restricted from the parent node's constraints, while making sure the two child nodes span all possible solutions from the parent. From the bounding stage, we found that O_C^* is not a best response. Then, we can search for the best response by either adding offers to O_C^*, or removing offers from O_C^*. We branch on these two cases:

1) The best response is $O_C^* \cup O'$, for some $O' \subseteq O_i \setminus O_C^*$, $O' \neq \emptyset$. In other words, O_C^* is included in the best response, along with at least one other offer. This child's constraint can be written as $C \cup \{\textit{all}(O_C^*), \textit{at-least}(O_i \setminus O_C^*)\}$.
2) There is at least one offer in O_C^* that is not included in the best response. This child node's constraint can be written as $C \cup \{\textit{not-all}(O_C^*)\}$.

Note that for every child node, we incorporate its parent's constraints. Furthermore, each child node has a constraint that the solution space will not include O_C^*, and both children have solution spaces that do not overlap. Therefore, at each node, we eliminate one strategy (namely, O_C^*) from being the best response, which the algorithm never checks again. Thus, this branch and bound algorithm is guaranteed to terminate in a finite number of steps.

VI. Results

A. Computational Experiments

We created problem instances using historical data from ground delay programs that occurred between June and August

2016 in San Francisco International Airport (SFO) and LaGuardia Airport (LGA) retrieved from the Air Traffic Control System Command Center database [16]. These airports were chosen based on their propensity for GDPs. In the five-year period 2012-2016, SFO experienced the largest number of GDPs among any US airport (916 GDPs, with an average duration of 5.6 hours). While LGA had the third-largest number of GDPs in this period (590 GDPs, behind SFO and EWR), it had the longest average duration of any airport (9.7 hours). The choice of these airports thereby allows us to consider airports with high propensity to GDPs, while also considering one airport with a very large number of shorter-duration and another with a smaller number of long-duration GDPs. Another desirable attribute, as we will see shortly, is that the two airports have very different airline mixes.

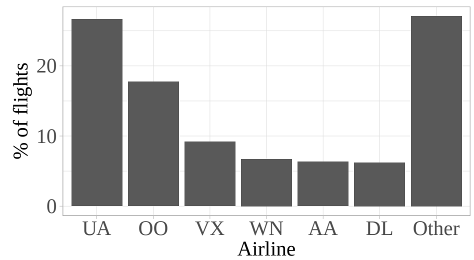
For each GDP, we retrieved the original flight schedules from Aviation System Performance Metrics [17] and Bureau of Transportation Statistics [18]. A summary of these GDPs are shown in Table I. We discretized time into 15-minute intervals; therefore all departure and landing times were rounded to the nearest 15 minutes. Since the announced GDP capacities are on an hourly basis, we divided this number by 4 obtain the capacities for every 15 minutes, while rounding appropriately to whole numbers (e.g. an hourly capacity of 43 turns into 15 minutes capacities of 11, 11, 11, 10). Using these new capacities, we rescheduled the affected flights using RBS (first-scheduled-first-served). This new schedule of delayed flights was used as the initial allocation, ϕ_0, for the reallocation mechanisms.

The last thing needed to set up a reallocation mechanism are the airlines' cost functions, which unfortunately is not possible to attain for each historical flight. Thus, we made assumptions about the cost function structure and developed estimates of flight costs using the aircraft capacities. In particular, we assumed that costs are linear in delay time, and also linear in the capacity of the aircraft. This implies that airlines which operate larger aircrafts have larger absolute delay costs. For each flight from the major airlines, we retrieved aircraft capacity data from Bureau of Transportation Statistics, and then multiplied this number with a normally distributed random multiplier with mean 1 and standard deviation 0.1. We then normalized these costs so that the average delay cost for all flights between all airlines for one hour of ground delay is 1.

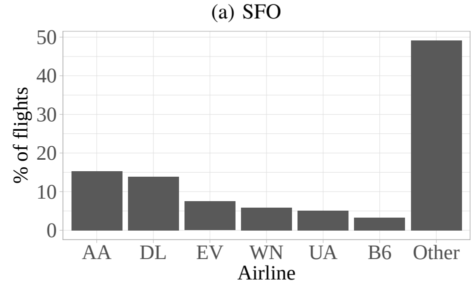
TABLE I: Summary statistics for the GDPs which were used for the experiments in this section.

	SFO	LGA
Number of GDPs	62	36
Average duration (hrs)	5.26	9.79
Average # of delayed flights	60.3	163.3
Average delay of delayed flights (hrs)	0.50	0.55

The airline composition in each airport can be seen in Fig. 8, and the assumed distribution of flight unit delay costs for each airline can be seen in Fig. 9. The airports SFO and LGA were chosen because the type of GDPs and the airline composition are very different at the two airports.

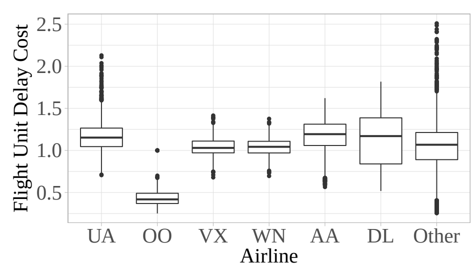


(a) SFO

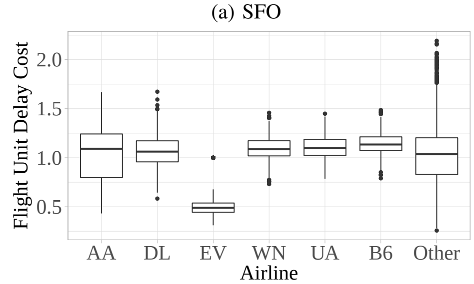


(b) LGA

Fig. 8: Airline composition for flights flying into SFO and LGA. The six biggest airlines flying into SFO are: United (UA), SkyWest (OO), Virgin America (VX), Southwest (WN), American (AA), and Delta (DL), and the six biggest airlines flying into LGA are American (AA), Delta (DL), ExpressJet (EV), Southwest (WN), United (UA), and JetBlue (B6).



(a) SFO

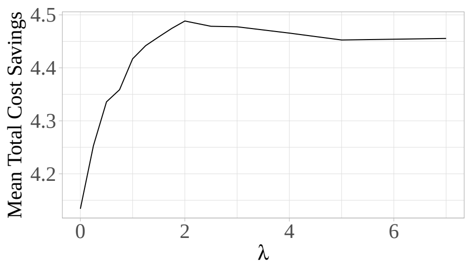


(b) LGA

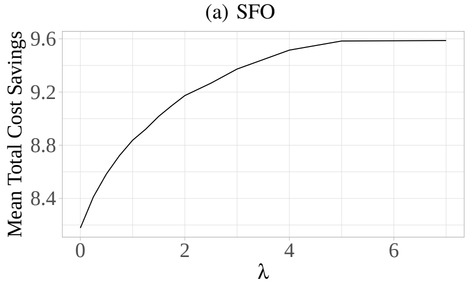
Fig. 9: Distribution of flight unit delay costs for each airline.

B. *Effect of Fairness on Total Cost Savings*

Recall that an allocation is λ-fair if for every airline, the absolute values of the net movements of their flights is less than or equal to λ. We see the effect of changing the value of λ on total cost savings. In this section, we disregard the issue of incentive compatibility, and assume that airlines are truthful. For SAP, the airlines reveal their true delay costs, and for the two-for-two trades mechanism, each airline uses the naive strategy.



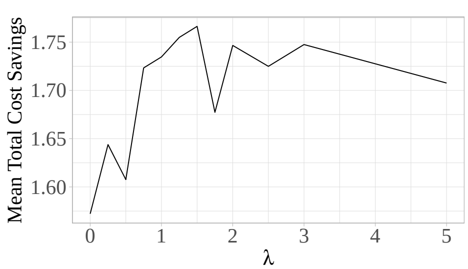
(a) SFO



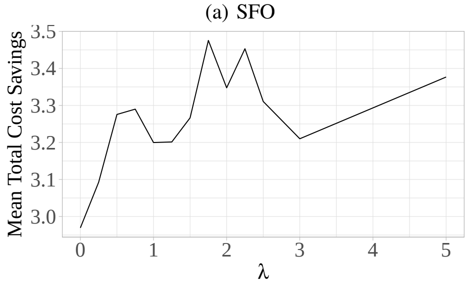
(b) LGA

Fig. 10: The mean total cost savings and the mean total delta from the SAP mechanism as λ is varied.

Because a low value of λ restricts the space of allocation that a mechanism can choose from, we expect that increasing λ will increase cost savings. We see this general trend for the SAP mechanism, as seen in Fig. 10. Because the SAP mechanism minimizes the total *scaled* cost, increasing λ does not necessarily imply that the actual cost savings will increase; and we see that this is not strictly the case for SFO.



(a) SFO



(b) LGA

Fig. 11: The mean total cost savings and the mean total delta from the two-for-two mechanism as λ is varied.

For the two-for-two mechanism, the relationship between λ and cost savings is not as clear as in the SAP mechanism, as seen in Fig. 11. The two-for-two mechanism maximizes the number of accepted trades, and it does not have any information on any individual flight costs. When we looked at the *number* of accepted trades, there was almost no change from when fairness was enforced or not (Table II). Therefore, from the mechanism's perspective, λ hardly has any impact in the resulting allocation.

TABLE II: The decrease in the number of accepted offers when the two-for-two mechanism was run with strict fairness enforced ($\lambda = 0$), compared to when it was run with no fairness constraint.

# decrease of accepted offers	% instances	
	SFO	LGA
0	93.5	97.2
1	6.5	2.8

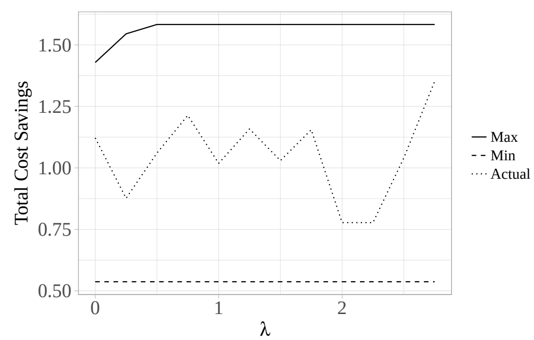


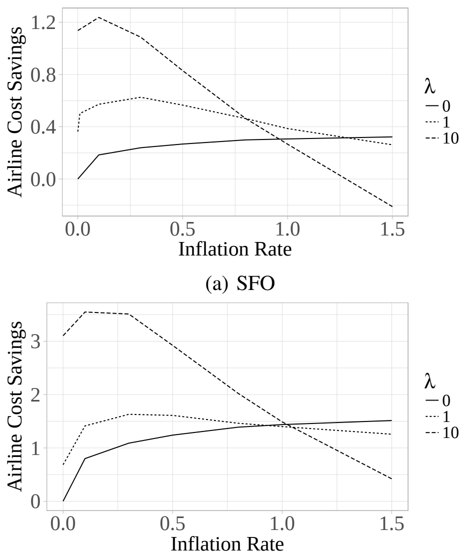
Fig. 12: The range of total cost savings of all optimal solution for different values of λ for a GDP in SFO which occurred on 06/13/2016,. The "Actual" value is the cost savings from the solution that the solver chose.

Taking a closer look into the solutions, the reason for this is because there happens to be multiple optimal solutions in the two-for-two mechanism. Fig. 12 shows the range of total cost savings of all optimal solutions for one particular GDP. We see that there is a large discrepancy in the minimum and maximum cost savings for an optimal solution, and it is not the value of λ that determines the cost savings, but rather the particular solution that the optimization solver happens to pick. This trend is similar for other GDPs.

We conclude that for the two-for-two mechanism, λ does not have much of an impact on the total cost savings; but rather, the cost savings is determined by the particular optimal solution that the solver happens to choose. The presence of these multiple optimal solutions suggest that we could increase efficiency by getting further input from the airlines on tie-breaking preferences. However, asking for additional input from the airlines will necessitate further analysis on incentive compatibility of those inputs.

C. *Inflation Strategy for SAP*

Next, we simulated the inflation strategy for various values of the inflation rate for the SAP mechanism. Recall that an inflation rate of 1 corresponds to the truthful strategy, and a rate of 0 implies that all flights have the exact same delay cost. For each of the GDPs in SFO and LGA and each of the six largest airlines in each airport, we ran one experiment where that airline used the inflation strategy, and everyone else used the truthful strategy. The average cost savings by the airline using the inflation strategy is shown in Fig. 13.



(a) SFO

(b) LGA

Fig. 13: The mean cost savings for the airline using the inflation strategy, as the rate for the inflation strategy is varied, for different value of λ which controls the level of fairness. An inflation rate of 1 corresponds to the truthful strategy.

We see that the best rate depends greatly on the value of λ that was chosen for fairness. When $\lambda = 0$, the cost savings increase as the inflation rate increases. On the other hand, hen $\lambda = 10$, the best inflation rate to use is 0.1, and the cost savings sharply decrease as the inflation rate increases.

We give intuition on these results using a simple example. Suppose there is one airline with two flights, f_1, and f_2, and f_1 is twice as valuable as f_2. Then, the truthful scaled unit delay cost of f_1 and f_2 would be 1.33 and 0.67 respectively. Denote by $\phi(n_1, n_2)$ the allocation where f_1 moves up by n_1 slots (moving down by $|n_1|$ slots if $n_1 < 0$), and f_2 moves up by n_2 slots. Suppose strict fairness is enforced ($\lambda = 0$). In this case, if this airline's flights were to be reallocated, then the only valid allocation is $\phi(n, -n)$, for some $n \in \mathbb{N}$, since the resulting allocation must be fair and individually rational. The net delay cost reduction by such a movement is $0.66n$. However, suppose they misreported by inflating their costs by a rate of 2, making their unit delay costs 1.66 and 0.34 respectively. Then, the mechanism believes that the cost reduced by the allocation $\phi(n, -n)$ is $1.32n$, rather than the actual $0.66n$. The mechanism therefore has a bigger incentive to make this allocation with a larger value of n. Therefore, when fairness is strictly enforced, it is better to inflate the cost differences as much as possible. On the other hand, suppose $\lambda = 10$. Then, since fairness is not heavily enforced, it is *not* the case that the only type of movements for the airline are $\phi(n, -n)$. If the airline used the inflation strategy with a rate of 2, then the allocation $\phi(n, -4n)$ is still a valid allocation. With that allocation, the airline's delay cost will actually *increase* by $1.35n$. If the airline reports truthfully, then $\phi(n, -2n)$ is still a valid allocation, which will not change their total delay costs. However, if the airline actually deflates their costs, then $\phi(n, -2n)$ does not become valid since IR is enforced. When the airline deflates their costs, only the favorable allocations for the airline such as $\phi(n, -n)$ or even $\phi(n, m)$ with $n, m > 0$ remain valid. These observations suggest that whether fairness is enforced or not, it is not difficult to manipulate the mechanism.

Now, suppose *all* airlines use the inflation strategy with the *same* value of the inflation rate. Then, the SAP mechanism would output the same allocation as if all airlines used the truthful strategy. This is because the SAP mechanism reallocates flights based on the relative difference in flight delay costs. If all airlines use the inflation strategy with the same inflation rate, then the relative difference in flight delay costs do not change.

This fact, along with the trends in Fig. 13, suggest that this mechanism does not have an equilibrium solution when all airlines use the inflation strategy. Suppose that the trends in Fig. 13 hold for all airlines. When fairness is enforced, it is desirable for all airlines to inflate their costs. However, if everyone inflates their costs, the outcome from the mechanism is the same as if no airlines inflated their costs. Then, it is desirable for airlines to inflate their costs even further. In this scenario, the airlines' strategies do not converge to an equilibrium. A similar argument can be made for deflating their costs when fairness it not enforced. Therefore, even when airlines are restricted to use the simple inflation strategy, our results suggest that there is no Nash Equilibrium.[3] A theoretical analysis of this conjecture is a direction for future research.

D. Threshold Strategy for Two-for-Two Trades

Next, we analyze the threshold strategy for the two-for-two trades mechanism. Since we saw from Section VI-B that the level of fairness does not have a large impact on this mechanism, we simply assume fairness to not be enforced.

1) Best Response Algorithm: We first assessed the performance the branch and bound algorithm for the best response problem. For each of the 62 GDPs, we made one instance for each of the six biggest airlines in SFO. For each of these airlines, we ran the best response algorithm given that the strategies of all other airlines is the naive strategy. For each instance, we ran the branch and bound algorithm for up to 5,000 iterations, where processing one node of the branch and bound tree represents one iteration. If the algorithm did not terminate by then, then we simply recorded the best strategy found so far, and what percentage of the upper bound it achieved. We evaluated the performance of the algorithm on the percentage of the upper bound a feasible solution was able to achieve (if an instance terminated, then it would achieve 100%), and the number of iterations of the algorithm took. The results, grouped by airline, are shown in Table III.

From the table, we see that the algorithm performed worse on larger airlines, since an airline with many flights will have a larger strategy space. Out of the instances that did

[3]The existence of a Nash Equilibrium is not guaranteed in this mechanism because the strategy space for each player is infinite and noncompact.

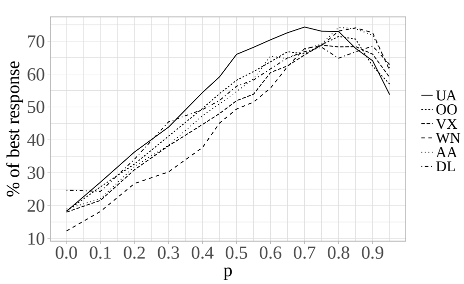
TABLE III: Performance of the branch and bound algorithm for the best response problem for SFO.

Airline	Average # flights	% instances that terminated	% of upper bound achieved		# of iterations	
			Mean	Median	Mean	Median
UA	56.6	79.0	96.9	100.0	1136.7	8.5
OO	37.5	83.0	97.0	100.0	633.4	4.0
VX	19.6	98.0	99.5	100.0	59.3	1.0
WN	14.5	100.0	99.9	100.0	27.5	1.0
AA	13.7	100.0	100.0	100.0	1.6	1.0
DL	13.4	100.0	100.0	100.0	8.1	1.0
Total	25.9	93.0	98.9	100.0	439.3	1.0

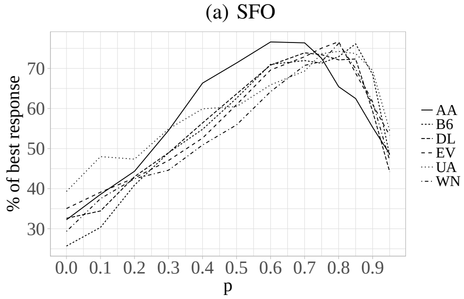
not terminate in 5,000 iterations, the average number of all possible offers for that airline was around 60,000. However, most of the time, the algorithm terminated very quickly; it converged after one iteration for 71.2% of all instances.

One of the key features of a best response strategy is that it has the possibility of including offers of negative utility. However, 79% of best response strategies did not include any offers of negative utility, and 11.5% had exactly one negative utility offer. This implies that airlines do not lose very much by only considering offers of positive utility, which is what the naive and threshold strategies do.

2) Threshold Strategy: Next, we simulated the threshold strategy and evaluated their performance compared to the best response strategy. Similar to the inflation strategy, we assumed that all other airlines used the naive strategy, and we let one airline use the threshold strategy with varying values of p, from 0 to 0.95. The results are shown in Fig. 14. If the algorithm to compute the best response did not terminate, then we used the upper bound of the best response given by the branch and bound tree instead.



(a) SFO



(b) LGA

Fig. 14: The mean cost savings of the threshold strategy as p varies from 0 to 0.95, as a percentage of the mean cost reduction of the best response. The naive strategy corresponds to the threshold strategy with $p = 0$.

We see from Fig. 14 that every airline in both airports have a very similar pattern of cost savings as p changes. The savings increase steadily as p increases, peaking when p is between 0.7 and 0.9, then it decreases past that. For SFO, a threshold of $p = 0.8$ achieves 71.6% of the best response cost savings on average, and for LGA, $p = 0.75$ achieves 73.1% of the best response savings. The best response strategy depends on knowing the exact strategies of other airlines, which is an impractical assumption. By only taking the airline's own flights into consideration, the threshold strategy is able to recover a significant percentage of the best response cost savings.

3) Equilibrium: Suppose all airlines used the threshold strategy. If all airlines used the same value of the threshold p, the average total cost savings from all airlines is shown by the solid lines in Fig. 15. However, in reality, airlines would choose the threshold p that is best for them. The dotted line shows the average total cost savings when the airlines play in equilibrium of threshold strategies, where the equilibrium was approximated using the following iterative algorithm:

1) For every airline i, initialize S_i to be the threshold strategy with $p = 0$.
2) For every airline i:
   a) Let $p_i \in \{0, 0.1, \dots, 0.9\}$ be the best response threshold strategy when all other airlines use strategy S_{-i}.
   b) Update S_i to be the threshold strategy for airline i with threshold p_i.
3) Repeat 2) until there are no changes to the best response for all airlines.

We note that not every instance converged to equilibrium after the algorithm ran for 24 hours — 16 out of 62 instance for SFO and 22 out of 36 instances for LGA did not converge. For these instances, we used the state at the end of the algorithm as the approximate equilibrium solution. We see that the equilibrium solution has higher total cost savings than when all airlines use the naive strategy, and significantly higher than if all airlines used a high threshold. This shows that the strategic behavior of airlines actually increases the social welfare of the system.

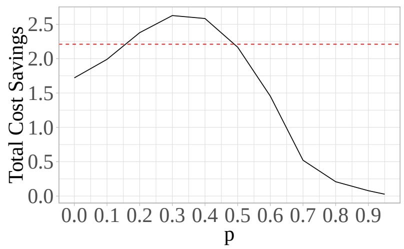
VII. Conclusions

In this paper, we evaluated mechanisms for landing slot exchange, and we found that assuming the airlines to be strategic can have a large impact on the outcome and performance of the mechanism.

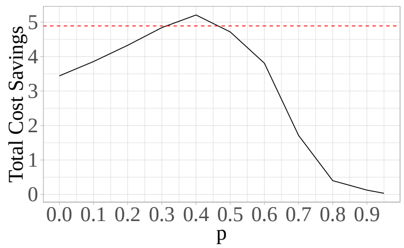
We showed that the SAP mechanism is not incentive compatible, and empirically showed that the best strategy strongly depends on the extent in which fairness is enforced.

We found that the two-for-two mechanism does not have a dominant strategy. We empirically showed that the simple threshold strategy performs almost as well as the best response strategy. We saw that airlines being strategic and playing in equilibrium of threshold strategies actually increased the total cost savings, compared to when airlines submit all positively valued offers.

A direction for future work is the design of other reallocation mechanisms, including in the presence of side payments or a virtual currency.



(a) SFO



(b) LGA

Fig. 15: The solid line represents mean total cost savings when all airlines use a threshold strategy with the same value of p. The dotted line is the total cost savings when airlines play in an equilibrium of threshold strategies.

References

[1] M. Soomer and G. J. Franx, "Scheduling aircraft landings using airlines' preferences," *European Journal of Operational Research*, vol. 190, no. 1, pp. 277–291, 2008.

[2] T. W. Vossen and M. O. Ball, "Slot trading opportunities in collaborative ground delay programs," *Transportation Science*, vol. 40, no. 1, pp. 29–43, 2006.

[3] D. Bertsimas and S. Gupta, "A proposal for network air traffic flow management incorporating fairness and airline collaboration," *Operations Research*, 2011.

[4] T. Vossen and M. Ball, "Optimization and mediated bartering models for ground delay programs," *Naval Research Logistics (NRL)*, vol. 53, no. 1, pp. 75–90, 2006.

[5] H. Balakrishnan, "Techniques for reallocating airport resources during adverse weather," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 2949–2956.

[6] J. Schummer and R. V. Vohra, "Assignment of arrival slots," *American Economic Journal: Microeconomics*, vol. 5, no. 2, pp. 164–85, 2013.

[7] M. O. Ball, R. Hoffman, D. Lovell, and A. Mukherjee, "Response mechanisms for dynamic air traffic flow management," in *Proceedings of the 6th Europe-USA ATM Seminar. Baltimore. US*, 2005.

[8] C. Barnhart, D. Bertsimas, C. Caramanis, and D. Fearing, "Equitable and efficient coordination in traffic flow management," *Transportation science*, vol. 46, no. 2, pp. 262–280, 2012.

[14] R. B. Myerson and M. A. Satterthwaite, "Efficient mechanisms for bilateral trading," *Journal of economic theory*, vol. 29, no. 2, pp. 265–281, 1983.

[9] T. Vossen, M. Ball, R. Hoffman, and M. Wambsganss, "A general approach to equity in traffic flow management and its application to mitigating exemption bias in ground delay programs," *Air Traffic Control Quarterly*, vol. 11, no. 4, pp. 277–292, 2003.

[10] T. Vossen, "Fair allocation concepts in air traffic management," Ph.D. dissertation, PhD thesis, Supervisor: MO Ball, University of Martyland, College Park, Md, 2002.

[11] S. J. Rassenti, V. L. Smith, and R. L. Bulfin, "A combinatorial auction mechanism for airport time slot allocation," *The Bell Journal of Economics*, pp. 402–417, 1982.

[12] M. Ball, G. Donohue, and K. Hoffman, "Auctions for the safe, efficient, and equitable allocation of airspace system resources," *Combinatorial auctions*, vol. 1, 2006.

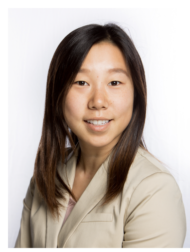
[13] L. Castelli, R. Pesenti, and A. Ranieri, "The design of a market mechanism to allocate air traffic flow management slots," *Transportation research part C: Emerging technologies*, vol. 19, no. 5, pp. 931–943, 2011.

[15] J. Schummer and A. Abizada, "Incentives in landing slot problems," *Journal of Economic Theory*, vol. 170, pp. 29–55, 2017.

[16] Federal Aviation Administration, "ATCSCC database," 2017, `http://www.fly.faa.gov/adv/advAdvisoryForm.jsp`.

[17] ——, "ASPM database," 2017, `https://aspm.faa.gov/apm/sys/main.asp`.

[18] United States Department of Transportation, "Bureau of Transportation Statistics," 2017, `https://www.bts.gov/`.



Jackie Baek is a Ph.D. candidate in the Operations Research Center at the Massachusetts Institute of Technology. She received her B.Math degree from the University of Waterloo in computer science and combinatorics and optimization.

Hamsa Balakrishnan is an Associate Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology (MIT). Before joining MIT, she was at the NASA Ames Research Center, after receiving her PhD from Stanford University and a B.Tech. from the Indian Institute of Technology Madras. Her research is in the design, analysis, and implementation of control and optimization algorithms for large-scale cyber-physical infrastructures, with an emphasis on air transportation systems. She was a recipient of the NSF CAREER Award in 2008, the Kevin Corker Award for Best Paper of ATM-2011, the inaugural CNA Award for Operational Analysis in 2012, the AIAA Lawrence Sperry Award in 2012, and the American Automatic Control Council's Donald P. Eckman Award in 2014.