Property Rights in Runway Slots Allocation.∗

Ken C. Ho† Alexander Rodivilov‡

October 24, 2016

For the latest version, please click here.

Abstract

This paper examines the runway slots allocation challenge that arises at airports during inclement weather. We argue that the ground delay program (GDP) violates property rights, as it prevents airlines from using runways during the time intervals for which they have paid. We propose a mechanism that takes into account the slots owned by airlines before the GDP begins. Building off the endogenous trading cycle, our mechanism is feasible, efficient, and strategy-proof.

JEL: C78, D47, D82, L93, L98, P14, R41.

Keywords: Many-to-One Matching, Sequential Dictatorship, Endogenous Hierarchy.

∗We are thankful for the helpful comments of Quan Wen and participants in UW Microeconomics seminar.
†Department of Economics, University of Washington, Seattle. Email: kh89@uw.edu.
‡Department of Economics, University of Washington, Seattle. Email: arodivil@uw.edu.
1 Introduction

Landing and takeoff slots provide the rights to use airport runways at specific times. Allocation of these slots affects industry competition and airline profits, which in turn influences the amount passengers pay for flights. Although slots are assigned to scheduled flights in advance, unpredictable conditions such as changes in weather may lead to the reallocation of slots among carriers. This paper examines slot allocation challenges and proposes a mechanism that improves recognition of economically relevant property: respecting airlines’ property rights.

Around the world, the airline industry suffers from inclement weather conditions. Conditions such as high winds and thunderstorms are responsible for 40 percent of the total amount of time flights are delayed. They limit the rate at which planes can take off and land on runways safely, and in most countries, they cause the airport’s landing schedule to be modified for safety issues, as prescribed by law. In the United States, the Federal Aviation Administration (FAA) is responsible for mitigating the negative consequences of weather change. It reduces the allowed arrival rates of flights at an affected airport, meaning some flights are postponed or canceled. To alter the schedule efficiently, the FAA may elicit some of the airline’s private information, such as the earliest feasible departure time from the origin airport for each flight. This opens the door to a matching problem under asymmetric information that is a subject of this paper.

Decades ago, the FAA used the Grover-Jack algorithm to manage this problem. This algorithm assigned arrival slots based on feasible departure times reported by the airlines themselves. If, for example, a plane was experiencing mechanical issues, under Grover-Jack, the delay was added to the weather delay, penalizing the airline twice. This is known as the “double penalty” problem,\(^1\) and it led to airlines intentionally withholding information about mechanical delays and other issues. This sometimes led to landing slots remaining unused when they could have been used to land other airlines’ flights. Eventually, the FAA and industry experts agreed the Grover-Jack needed improvement.

The FAA has developed a new scheme that assigns arrival slots based on original schedules. It currently is employing the GDP, a two-step procedure. The first stage of the GDP involves the ration-by-schedule (RBS) algorithm, which assigns the arrival slots among the participating airlines on a first-come, first-serve basis, according to the originally scheduled times of arrival (unlike the self-reported times used in Grover-Jack) For example, if the airport could land one flight every minute initially, then, after the GDP has been implemented, it can land one flight every two minutes. The first 30 flights from the initial schedule would

\(^1\)See http://cdm.fly.faa.gov for more information on the "Double Penalty" issue.
be assigned newly created thirty 2-minute slots.\textsuperscript{2}

The second step of the GDP involves the compression algorithm, which is an exchange mechanism that allows one company that cannot use a slot assigned during RBS to either reassign this slot to a later flight it operates or to exchange it with another airline.\textsuperscript{3} This trading feature gives compression a presumable advantage over the Grover-Jack mechanism. However, GDP prescribes how the slots must be exchanged: If an airline chooses not to use a slot, it can be used only by the earliest flight of another company.

We argue that a solution to the runway slot allocation problem should be mindful of property rights. Airlines buy\textsuperscript{4} runway slots for significant amounts of money, and in exchange, the airport grants airlines the right to use landing slots at specified times. Both the government and the airlines have agreed that purchased slots should be treated as property. A report funded by the European Commission suggests that airlines should be allowed to buy and sell their runway slots as they please, just like any other asset.\textsuperscript{5} In the United States, slots are treated as property under certain circumstances. For example, on June 9, 2015, United Airlines exchanged its takeoff and landing slots at JFK International Airport with Delta Air Lines for slots at Newark. This deal was approved by the regulator.\textsuperscript{6}

While the current RBS procedure assigns flights based on their initial schedules, we propose a mechanism that takes into account initial slot assignment through a different method. If an airline initially owns adjacent slots that constitute a new slot after the arrival rate is lessened, our method designates this new slot as belonging to the airline. The slot is free for any flight the airline chooses, and it may choose to give up the slot for a better alternative. The airline also may exchange other slots it owns for newly available ones; the more slots an airline initially owns, the more chances it has to pick its most desired slots.

Throughout the paper we assume that each airline has a priority ranking system for its flights such that all flights can be ranked strictly\textsuperscript{7} according to importance. We also assume\footnotetext[2]{According to RBS, the other 30 flights initially scheduled for the second 30-minute period would be delayed to an undetermined time period.}
\footnotetext[3]{When an airline has decided a slot cannot be used, it trades it for the best feasible substitute, usually owned by the airline that took the initial slot. The airline that gives up the initial slot is better off because the one it accepts typically is earlier than the original. Airlines are assumed to prefer scheduling flights as early as possible when feasible.}
\footnotetext[4]{We cautiously use word “buy,” as a slot is not a physical object, but constitutes the right to use a runway.}
\footnotetext[5]{See https://www.theguardian.com/business/2004/apr/11/theairlineindustry.theobserver for further information.}
\footnotetext[7]{Strict ranking is a crucial assumption, and relaxing it could exacerbate the problem significantly, which}
that for each flight, the airline prefers to assign the earliest feasible slot, where feasibility is defined as enough time for the flight take off from the origin airport and arrive at its destination. Thus, each airline is assumed to have preferences for sets of slots determined by lexicographic preferences over flights and the earliest feasible arrival times.

In this paper, we propose a mechanism that mitigates the negative effects of inclement weather conditions and respects carriers’ property rights over slots. Unlike the current GDP procedure, which does not improve carrier circumstances any more than RBS does, we take the initial slot allocation as a starting point. We define adjacent slots initially owned by the same airline as constituting a new slot the airline is endowed with after the arrival rate is reduced: The airline is free to either use this slot itself or exchange it with any other airline. For initial slots that become part of a newly created slot, we indicate that airlines jointly own a group of indivisible objects. If this is the case, we will reallocate these slots using a hybrid mechanism that combines serial dictatorship with endogenous priorities and the top-trading cycle (TTC) procedure.

When a persistent overload due to weather is predicted for the next several hours at an airport, the FAA announces that the arrival rate must be reduced. This results in a shortage of slots. During the length of time in which access restrictions are in force, the earliest feasible arrival times for initially scheduled flights may change due to delays at other airports or technical reasons. Only carriers know if some flights have to be postponed or even canceled. Our proposed mechanism starts with airlines reporting their private information to a central clearinghouse. First, airlines are asked to report a ranking profile of their initially scheduled flights during the announced time period. Second, each carrier reports a vector of earliest feasible arrival times for each flight.

After all the airlines report their private information, the central clearing house assigns newly created slots. In case there is a slot that can be used by only one airline (possibly by several flights), this slot is permanently assigned to that airline and the flight that was ranked first according to the submitted preferences. If there is a flight that can use several slots feasibly, the flight is assigned to the earliest one. In the case that one flight is assigned and some slots continue to be demanded by one airline only, the previous step is repeated. If airlines report their preferences and the earliest feasible arrival time honestly, which is the case in equilibrium, then all such slots will be assigned to carriers.

At this point of the algorithm, the only remaining slots are those that are demanded by more than one airline. We then distinguish two types of slots. If a new slot consists of several initial adjacent slots owned by the same airline, it will be designated as property of

is beyond the scope of this paper. See Abdulkadiroglu et al. (2009) and Edril and Ergin (2008) for the importance of tie-breaking rules.
the airline. The second type is assumed as jointly owned by all participating carriers, and positive probability may be assigned to any airline. If each slot has an assigned owner, a variant of the TTC can be used to determine to whom these slot should be assigned. If none have an assigned owner, a modification of the deferred acceptance (DA) algorithm or a form of a serial dictatorship can be implemented instead. Because our approach combines both the slots that are jointly owned and those owned by individual carriers, we use a hybrid mechanism that employs features of TTC and serial dictatorship with endogenous order.

Each remaining slot will be allocated via a lottery that proceeds as follows. Each airline receives a specific amount of “tickets” that is equal to the amount of initially owned slots. The chances to win a lottery for each airline is the ratio of the amount of lottery tickets it owns to the total amount of initially scheduled unassigned flights. The winner of a lottery points to a slot that is matches the earliest feasible arrival time of the winner’s highest-ranked flight. Two alternatives can then occur: either the slot will already belong to another carrier, or it will not. In the latter case, the slot would be assigned to the winner of the lottery and its highest-ranked flight. In the former case, the original owner of the slot is allowed to pick its preferred slot. If there is no conflict, the two slots are assigned to each airline. A cycle of several airlines may form at this step. Each airline that gets a slot at this point is sacrificing an amount of tickets equal to the ratio of the new arrival rate to the initial one. The total amount of tickets is reduced based on the amount of newly assigned flights. In addition to respecting property rights, our mechanism satisfies relevant variables: It is feasible and efficient. Moreover, it provides an outcome from the core and is strategy-proof.

The rest of the paper is organized as follows: Section 2 explores related literature; Section 3 explains the model and definitions; Section 4 provides a description of the mechanism; Section 5 describes the properties of the mechanism; Section 6 provides an illustration of how the mechanism works and compares it to compression; and Section 7 concludes the paper. All the proofs are presented in the appendix.

2 Related Literature

Our paper contributes to the literature on the allocation of heterogeneous indivisible objects without monetary transfers. We also complement the literature on matching houses with tenants. In general, one could distinguish three major models. First is Shapley and Scarf’s (1974) house market model, which consists of several agents who each initially own an indivisible good: a house. The agents have heterogeneous preferences and are considering switching the houses among themselves. The second model Hylland and Zeckhauser’s (1977) house allocation model, which differs from the first model in that no single agent owns a
house initially.

The third model, attributed to Abdulkadiroglu and Sonmez (1999), is a hybrid of the two. This model includes existing tenants who already own houses and newcomers who do not possess houses, but would like to. In this environment, each airline would own several parts of the indivisible objects (i.e., the newly created runway slots). This does not eliminate the possibility that an airline could be the sole owner of a slot. Given this, the model more closely resembles the house market model than it does the house allocation problem. Abdulkadiroglu and Sonmez’s (1999) hybrid is a limited case of our model, in which some airlines do not have any slots. However, in our model, each airline has the chance to gain more than one slot in the end, which is not the case in these versions of the house matching problem.

The three described models differ in how they define property rights over indivisible objects. In the house allocation problem, a group of agents jointly own a group of indivisible objects: Only a grand coalition can claim ownership of a particular object, whereas no smaller coalition, including a single agent, owns a house. The house market approach takes the opposite stance: Each agent possesses full control over one item only. An alternative interpretation is that every agent has a right to uniform distribution over initial endowments. This perspective is closer to the house allocation problem, since in our model, a part of a slot only is of no use to any airline. In all these papers, the number of objects to be allocated is equal to the number of agents; supply equals demand. We, in contrast, consider a problem of a different nature; after the GDP starts, some flights might be canceled so the amount of slots to be allocated may generally exceed the amount of flights.

We have found that only three papers in the economic literature address the airport slot allocation problem. Schummer and Vohra (2013) demonstrated that the currently used compression algorithm may not provide a core outcome, as some airlines might find more success in exchanging slots among themselves only. The authors propose an alternative mechanism based on TTC that yields a core outcome. The TTC mechanism belongs to the class of fixed endowment hierarchical exchange mechanism. However, the authors demonstrated that both the compression mechanism and the one they proposed might create circumstances in which it is not optimal for airlines to give up unusable slots. Our mechanism resolves the issue of failing to vacate unusable slots. While Schummer and Vohra (2013) took the outcome of the RBS as given, we endow airlines with newly created slots if they have paid for them already.

---

8However, this is not the case in the Operations Research and Management Science (OR & MS). See Handbook in OR & MS, Vol. 14, and references therein.

9See Papai (2000) and references therein.
Schummer and Abizada (2016) introduced a mechanism that makes vacating unusable slots optimal. This improves the arrival rate of other flights the airline owns, but marginally. The mechanism involves deferred acceptance and allows airlines to self-optimize; that is, it allocates several slots to an airline, which then freely assigns its flights to these slots. This solution provides only weak incentives to promptly report delays. In addition, this mechanism requires airlines to have a priority order over each slots, which makes comparing results with this study difficult. Chun and Park (2016) have studied the slot allocation problem assuming monetary transfers are feasible. We, in contrast, assume that monetary transfers are not feasible but that airlines have heterogeneous preferences over flights.

We differ from Schummer and Vohra (2013) and Schummer and Abizada (2016) in how we treat ownership of runway slots. We assume that if an airline paid for slots initially, it can continue to use them if they constitute a newly created slot. Schummer and Vohra (2013) and Schummer and Abizada (2016), meanwhile, respected property rights of the airline after RBS.

3 Model

There is a finite set of airlines $A = \{a_1, a_2, ..., a_{|A|}\}$ and a finite set of controlled flights $F^o = \bigcup_{a \in A} F^o_a = \{f_0, f_1, ..., f_{|F^o| - 1}\}$, where the $F^o_a$ denotes all flights owned by airline $a$. Some flights might be canceled during or before the start of GDP; we use $F \subseteq F^o$ to denote the set of non-canceled flights and $F_a \subseteq F^o_a$ to denotes all flights owned by airline $a$ that are not canceled. There is a set of original slots $S^o = \{s^o_0, s^o_1, ..., s^o_{|L| - 1}\}$, where the length of each slot is normalized to one unit of time. Note that $|F^o|$ of the $|L|$ original slots were owned by some airlines. Let the set of available GDP slots be $S = \{s_0, s_1, ...\}$, where the length of each slot is $l > 1$ unit of time. For $n = 0, 1, ..., \) slot $s_n$ is the time interval $[nl, (n + 1)l]$. We use $s_n$ to represent $nl$ on the time line. 

There is an earliest feasible arrival time $e_f \in S$ for each flight $f \in F$; each flight $f$ can be feasibly assigned to slot $n$ only if $e_f \leq s_n$. Let $e = (e_f)_{a \in F} \in \mathbb{R}^{|F|}_+$ be the vector of all earliest feasible arrival times. A landing schedule is an injective function $\Pi : F \to S$.

---

10 A flight is “controlled” means it is included in the GDP program. Note that we do not used the term “affected” since there are more flights affected by the launch of GDP in different airports.

11 Note that available GDP slots do not have to be adjacent since there are exempted flights in GDPs (for examples, international and airborne flights). The number of arrivals an airport can accept each hour is called the Airport Acceptance Rate/Airport Arrival Rate (AAR); if 1 unit of time is 1 minute, then $l = \frac{60}{AAR}$.

12 We model one runway, but the model can be extended to a problem with several runways, in which there will be multiple identical slots available at a time.

13 Following the literature, we call $e_f$ the earliest feasible arrival time for $f$, but strictly speaking, $e_f$ is the earliest feasible arrival slot for $f$. 

7
Let $M$ be the set of all landing schedules. A partial landing schedule $\Pi_a : F_a \rightarrow \Phi(a)$ for each $a$. A landing schedule $\Pi$ is feasible if $\forall f \in F$, $\Pi(f) \geq e_f$. A landing schedule $\Pi$ is non-wasteful if $\#f \in F$ and $s \in S$ such that $\Pi^{-1}(s) = \emptyset$ and either $\Pi(f) = \emptyset$ or $e_f \leq s < \Pi(f)$.

$\Phi : A \rightarrow 2^S$ is a slot ownership function if $a \neq a' \implies \Phi(a) \cap \Phi(a') = \emptyset$, where $\Phi(a)$ is the set of slots owned by airline $a$. Let $\Pi_a : F_a \rightarrow \Phi(a)$ be a partial landing schedule for $a$. $\Phi$ is consistent with $\Pi$ if $\forall a \in A$, $\forall f \in F_a$, $\Pi(f) \in \Phi(a)$. A pair $(\Pi, \Phi)$ satisfying this consistency condition is an assignment. The original assignment $(\Pi^o, \Phi^o)$ is assumed to be consistent, so given the initial landing schedule $\Pi^o : F^o \rightarrow S^o$, we can derive the initial slot ownership function $\Phi^o : A \rightarrow 2^{S^o}$. Let $S_a = \{s_n \in S|s_n = [nl, (n+1)l]| \subseteq \cup_{o \in \Phi^o(a)}[n, n+1]\}$ be the set of available GDP slots that their time intervals are entirely owned by airline $a$. Let $S_A = \cup_{a \in A} S_a$.

### 3.1 Preferences

We assume each airline company $a \in A$ has an importance ranking of flights that it owns. Formally, let $R_a$ be a strict total order over $F_a$; we interpret $R_a$ as $a$’s importance ranking. If $f \in F_a$ is more important than $f' \in F_a$, we write $f R_a f'$. Given $F_a$ and $R_a$, let $e_a = (e_{f_a,1}, e_{f_a,2}, ..., e_{f_a,|F_a|}) \in \mathbb{R}^{|F_a|}$ be a vector of earliest feasible arrival times such that $f_{a,i} R_a f_{a,i+1}$ for $i \in \{1, ..., |F_a|\}$. Let $R = (R_a)_{a \in A}$ be the importance ranking profile.

Airline $a$’s preference over landing schedules is induced by $R_a$ and $e_a$. All else being equal, airline $a$ prefers flight $f \in F_a$ with $e_f$ to land as early as possible. Given a landing schedule $\Pi$, we define the delay for $f$ by $d_f(\Pi) = s_f - e_f$, where $s_f$ is the slot assigned to $f$, and if $s_f = \emptyset$ or $s_f < e_f$, the we set $d_f(\Pi) = M$, where $M$ is a sufficient large positive number such that $M > s - e_f$ for all $s \in S$ and $e_f \in T$.$^{14}$

For all landing schedules $\Pi$ and $\Pi'$, airline $a$ lexicographically prefers $\Pi$ to $\Pi'$ if and only if the first non-zero coordinate of $x_a = (x_1, x_2, ..., x_{|F_a|})$ is positive, where for $i \in \{1, ..., |F_a|\}$ and $f_{a,i} R_a f_{a,i+1}$, $x_i = d_{f_{a,i}}(\Pi') - d_{f_{a,i}}(\Pi)$, and we write $\Pi \succ_a \Pi'$.$^{15}$ Conversely, if the first non-zero coordinate of $x_a$ is negative, it prefers $\Pi$ to $\Pi'$. If airline $a$ is indifferent between $\Pi$ and $\Pi'$, we write $\Pi \sim_a \Pi'$; this will happen when (i) both of them are infeasible, and (ii) all coordinates of $x_a$ equal to 0, which will only happen when $\Pi_a = \Pi'_a$. Since airlines only care about their own flights, we will also use $\succ_a$ to compare partial landing schedules for $a$. Let $\succeq_a = (\succ_a)_{a \in A}$ be the preference profile of all airlines.

An instance of an airport slots allocation problem is a 6-tuple $I = (S, A, F, R, e, \Phi^o)$.

---

$^{14}$Note that this delay is respected to $e_f$ but not its original slot $s_f^o$.

$^{15}$Lexicographic preference does not rule out an airline prefers an infeasible landing schedule to a feasible landing schedule.
4 The Mechanism

A lottery mechanism \( \varphi : (R, e) \to \Delta M \) is a mapping that assigns a lottery over the set of landing schedule to every strategy profile \((R, e)\). Let \( \varphi_f(R, e) \) be the outcome for \( f \in F \), and \( \varphi_a(R, e) \equiv \cup_{f \in F_a} \varphi_f(R, e) \) be the outcome for \( a \in A \).

Let \( \varphi^r_s(R, e) \) be a realized landing schedule, where \( R_s \) is the resulted ordering from the compound lottery defined in the mechanism.

Pre-competition Stage (allocation of non-scarce resources):

Set \( F^{0-0} = F \) and \( e = e^{0-0} \).

(i) Use the smallest element of \( e^{0-0} \) to identify the earliest slot \( s \in S \) that is demanded, eliminates all slots that are earlier than \( s \), denote the resulting set \( S^{0-1} \).

(ii) If the earliest slot only can be used by one flight \( f \in F^{0-0} \), assign the slot to this flight. Denote the resulting sets \( S^{0-1} \), \( F^{0-1} \) and \( e^{0-1} \) respectively.

(iii) Repeat (i) and (ii) until the earliest slot is demanded by more than one flights. In general, denote the intermediate sets \( S^{0-t}, F^{0-t} \) and \( e^{0-t} \) for \( t = 2, ..., \).

Denote the resulting sets \( S^1, F^1 \) and \( A^1 \), where \( A^1 \) includes airlines with flights in \( F^1 \). In general, \( S^k, F^k \) and \( A^k \) for \( k = 1, 2, ..., \) only include slots, flights and airlines in step \( k \).

Remove flights \( F \setminus F^1 \) from \( R \) accordingly; denote the resulting ranking profile \( R^1 \). Let \( F^N_a \) be the flights of airline \( a \) that are assigned a slot in this step, and let \( F^1_a \) be the remaining flights of airline \( a \) that are not assigned a slot. In general, \( F^k_a \) for \( k = 1, 2, ..., \) only includes flights of airline \( a \) in step \( k \).

Main Stage (allocation of scarce resources):

According to \( \Phi^o \), construct \( S_a \) for each \( a \in A \). If \( S_a \cap S^1 \) is non-empty, order \( S_a \cap S^1 \) in an increasing order, tentatively assign the earliest slot to \( a \)'s highest ranked flight in \( F^1_a \) that can feasibly use it. Assign the next earliest slot to its remaining highest ranked flight that can feasibly use it. Repeat until there is no more slot.

The price of a lottery ticket is 1 controlled flight. The budgets for each airline \( a \in A^1 \) is \( b^1_a = |F^o_a| - |F^N_a| \).

Run a lottery such that each airline \( a \)'s chance to win is

\[
\frac{b^1_a}{\sum_{a \in A^1} b^1_a}.
\]

Step 1 - Let \( a(i) \) be the \( i \)th important flight of airline \( a \) according to \( R^1 \).\(^{16}\) If \( a \) wins the lottery in step 1, put \( a(1) \) in the 1st place of \( R_S \), where \( R_S \) is the order of picking a slot.

\(^{16}\)We use \( a(i) \) but not \( f_{a,i} \) since \( i \)th important flight in \( R^1_a \) might be different from \( i \)th important flight in \( R_a \).
Assign \(a(1)\) the earliest slot in \(S^1\) such that \(a\)'s highest ranked flight in \(F^1_a\) that can feasibly use it.

(i) If it picks an empty slot \(s'\), go to the next flight in line. If no such flight exists, go to the next step.

(ii) If it is demanding some slot \(s'\) that is endowed to some other airline \(b\), and \(b\) has remaining flight in \(F^1\), insert \(b(1)\) before \(a(1)\), that is, let \(R_S : b(1), a(1)\). If \(b\) has no remaining flight, go to the next step. If a cycle forms, it is formed by the remaining most important flights and slots that owned by their airlines. Each of these flights requests a slot that owned by the airline which is next in the cycle. For each \(a \in A\), let \(s_a\) be some slot in \(S_a \cap S^1\). A cycle in step 1 is an ordered list \((s_a, a(1), s_b, b(1), ..., s_z, z(1))\) of slots and flights where \(a(1)\) demands \(s_a\), ..., \(z(1)\) demands \(s_z\). Remove all flights in the cycle by assigning them the slots they demand. Go to the next step.

(iii) If such flight already has a slot \(s\) endowed to it and its want to use it, insert \(a(2)\) after \(a(1)\), that is, let \(R_S : a(1), a(2)\).

An airline could get at most 1 slot in a cycle, but it might get more from (iii). For each airline \(a\) that gets \(m\) slots in this step, reduce its budget to \(b^2_a = |b^1_a| - m\). Denote the resulting sets \(S^2\), \(F^2\), \(A^2\) and \(F^2_a\) for each \(a \in A\).

**Step** \(n \geq 2\) - run a lottery such that each airline \(a\)'s chance to win is

\[
\frac{b^n_a}{\sum_{a \in A^n} b^n_a}.
\]

Let \(a(n - 1)\) be the last flight of airline \(a\) in \(R_S\). Assign \(a(n)\) the earliest slot in \(S^n\) such that \(a\)'s highest ranked flight in \(F^n_a\) that can feasibly use it.

(i) If it picks an empty slot \(s'\), go to the next flight in line. If no such flight exists, go to the next step.

(ii) If it is demanding some slot that is endowed to some other airline \(b\), and \(b\) has remaining flight in \(F^n\), insert \(b(n)\) before \(a(n)\), that is, let \(R_S : ..., b(n), a(n)\). If \(b\) has no remaining flight, go to the next step. If a cycle forms, a cycle in step \(n\) is an ordered list \((s_a, a(n), s_b, b(n), ..., s_z, z(n))\), where for each \(a \in A\), \(s_a\) is some slot in \(S_a \cap S^n\). Remove all flights in the cycle by assigning them the slots they demand. Go to the next step.

(iii) If such flight already has a slot \(s\) endowed to it and its want to use it, insert \(a(n + 1)\) after \(a(n)\), that is, let \(R_S : ..., a(n), a(n + 1)\).

For each airline \(a\) that gets \(m\) slots in this step, reduce its budget to \(b^{n+1}_a = |b^n_a| - m\). Denote the resulting sets \(S^{n+1}\), \(F^{n+1}\), \(A^{n+1}\) and \(F^{n+1}_a\) for each \(a \in A\).

The main stage terminates when the set of remaining flights \(F^k\) becomes empty.
Supplemental Stage:
Let $V^1$ be the set of vacant slots and $A^1_S$ be the set of airlines with positive budgets $b^S_{a:1}$ for $a \in A^1_S$.

Step $n \geq 1$: If the earliest vacant slot was endowed to some $a$ with positive budget, assign it to $a$. Otherwise, run a lottery such that each airline $a$’s chance to win the earliest vacant slot is

$$\frac{b^S_{a:n}}{\sum_{a \in A^1_S} b^S_{a:n}}.$$

Denote the resulting sets $V^{n+1}$, $A^{n+1}_S$. Update $b^S_{a:n+1} = b^S_{a:n} - 1$ for the winner $a$ and $b^S_{a':n+1} = b^S_{a':n}$ for $a' \in A^{n+1}_S \setminus \{a\}$. Repeat until all budgets are zero.

This algorithm will stop in finite steps. Indeed, it might stop right after step 0. Note that there will always be feasible slots for flights, yet they might be late and outside the GDP time window. The initial budget for airline $a$ to compete scare resources is $b^1_{a}$, which includes the flights that are canceled, so $b^1_{a} \geq |F^1_{a}|$. This will provide no incentive for airline $a$ to hide its cancellations. An airline $a$ will stop participating in the next main stage lottery once it has no remaining flight even if it still has positive remaining budget. This budget, which is the number of $a$’s canceled flights, will become the initial budget in the supplemental step, and it is denoted by $b^S_{a:1}$ in the algorithm. $V^1$ includes vacant slots that are not assigned in both step 0 and step 1.

In a typical YRMH-IGYT mechanism, a cycle is formed by exclusively existing tenants and their houses. By contrast, a cycle in our mechanism forms by flights and slots that are not necessarily endowed to these flights.

5 Properties of the Mechanism

A lottery mechanism $\varphi$ is feasible (non-wasteful) if for any strategy profile $(R, e)$, $\varphi(R, e)$ is feasible (non-wasteful).

A landing schedule $\Pi$ is Pareto efficient if $\not\exists \Pi'$ such that (i) $\forall a \in A$, $\Pi' \succeq_a \Pi$, and (ii) $\exists a \in A$, $\Pi' \succ_a \Pi$. A lottery mechanism $\varphi$ is ex post Pareto efficient if it gives positive weight to only Pareto efficient landing schedule.

A (lottery) mechanism $\varphi$ is strategy-proof if there does not exist $(R, e)$, an airline $a$ with $\widehat{R_a, e_a}$ such that $\varphi_a(\widehat{R_a, e_a}, (R, e)_{-a}) \succeq_a \varphi_a(R, e)$.

Let $\Pi^S_{a'}$ be a partial landing scheduled for $a$ such that $\Pi^S_{a'} \succeq_a \Pi'_a$ for all $\Pi'_a$ given $F_a$ and $S' \subseteq S$.

A landing schedule $\Pi_a$ is individually rational if $\forall a \in A$, $\Pi_a \succeq_a \Pi^S_{a'}$. A lottery mechanism $\varphi$ is ex post individually rational if it gives positive weight to only individual
rational landing schedule.

A landing schedule $\Pi$ is in the core if $\not\exists \Pi'$ and $A' \subseteq A$ such that (i) $\forall f \in \bigcup_{a \in A'} F_a$, $\Pi'(f) \in S_{A'}$, and (ii) $\forall a \in A'$, $\Pi' \succ_a \Pi$. A (lottery) mechanism $\varphi$ is a core-selecting if for any strategy profile $(R, e)$, $\varphi(R, e)$ is in the core.

A (lottery) mechanism $\varphi$ respects property rights if $\forall s \in S_a$, in the first step that $s$ is being assigned, $a$ has the right to (i) use it, (ii) trade it for a better alternative.

**Theorem:** The endogenous trading cycle mechanism is feasible, non-wasteful, ex post individually rational, ex post Pareto efficient, respects property rights, core-selecting, and strategy-proof.

**Proof:** See Appendix.

### 6 Example

We provide an example to illustrate how the proposed mechanism works. Suppose the length of time during which restrictions on access to the airport are in force equals to 10 initial slot durations, $L = 10$. Suppose the FAA requires new arrival rate to be twice smaller than the initial one, that is, $l = 2$, so we have twelve newly created slots twice longer than initial ones. There are three airlines that we refer to as $A, B$ and $C$ and a set of flights for each of them, $F_A = \{f_3, f_4, f_7, f_{10}\}$, $F_B = \{f_2, f_5, f_6\}$ and $F_C = \{f_1, f_8, f_9\}$, respectively.

Flights $f_1$ and $f_2$ were canceled. In addition, for each of the remaining flights there is an earliest feasible arrival time (nominated in units of new slots) $\vec{e}_f = (\times, \times, 1, 1, 1, 2, 2, 0, 4, 6)$ for each flight so that each flight can be feasibly assigned to slot $s_n$ only if $e_f = n \leq s_n$ for $n = 0, 1, 2, ...$. See Table 1 below for the initial schedule and slot assignment.

<table>
<thead>
<tr>
<th>Flight</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>$e_f$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

**Table 1. Initial Slots Allocation.**

### 6.1 Our Algorithm

After the new arrival rate is determined there are 10 newly created slots, $s_0, s_1, ..., s_{11}$, to be allocated. Assume that priorities of each airline are as follows:

$\triangleright_A: f_3, f_7, f_4, f_{10}$;

$\triangleright_B: f_2, f_6, f_5$;

$\triangleright_C: f_1, f_9, f_8$. 

12
According to our mechanism slot $s_1$ becomes endowment of airline $A$; slot $s_2$ belongs to airline $B$:

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$A$</td>
<td>$B$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 0.** Given that slot $s_0$ can be feasibly used only by $f_8$. Thus, slot $s_0$ is permanently assigned to $f_8$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$A$</td>
<td>$B$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There are 9 slots left to be allocated at this point. We define a vector of probabilities that each airline wins a lottery, $\vec{Pr}_{\text{win}} = (Pr(A), Pr(B), Pr(C))$ and combine the outcome of the next step in Table 2 below.

<table>
<thead>
<tr>
<th>Step</th>
<th>$\vec{Pr}_{\text{win}}$</th>
<th>Winner</th>
<th>Slots assigned</th>
<th>Slots left</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\left(\frac{4}{9}, \frac{3}{9}, \frac{2}{9}\right)$</td>
<td>$A$</td>
<td>$f_3$ to $s_1$</td>
<td>$s_2$, $s_3$, $s_4$, $s_5$, $s_6$, $s_7$, $s_8$, $s_9$</td>
</tr>
<tr>
<td>2</td>
<td>$\left(\frac{3}{8}, \frac{3}{8}, \frac{2}{8}\right)$</td>
<td>$A$</td>
<td>$f_6$ to $s_2$, $f_7$ to $s_3$</td>
<td>$s_4$, $s_5$, $s_6$, $s_7$, $s_8$, $s_9$</td>
</tr>
<tr>
<td>3</td>
<td>$\left(\frac{2}{6}, \frac{2}{6}, \frac{2}{6}\right)$</td>
<td>$A$</td>
<td>$f_4$ to $s_4$</td>
<td>$s_5$, $s_6$, $s_7$, $s_8$, $s_9$</td>
</tr>
<tr>
<td>4</td>
<td>$\left(\frac{1}{5}, \frac{2}{5}, \frac{2}{5}\right)$</td>
<td>$C$</td>
<td>$f_9$ to $s_5$</td>
<td>$s_6$, $s_7$, $s_8$, $s_9$</td>
</tr>
<tr>
<td>5</td>
<td>$\left(\frac{1}{3}, \frac{2}{3}, \frac{0}{3}\right)$</td>
<td>$B$</td>
<td>$f_7$ to $s_7$, $f_{11}$ to $s_{10}$</td>
<td>$s_7$, $s_8$, $s_9$</td>
</tr>
<tr>
<td>6</td>
<td>$\left(\frac{1}{1}, \frac{0}{1}, \frac{0}{1}\right)$</td>
<td>$A$</td>
<td>$f_{10}$ to $s_7$</td>
<td>$s_8$, $s_9$</td>
</tr>
<tr>
<td>S:1</td>
<td>$\left(\frac{0}{1}, \frac{1}{2}, \frac{1}{2}\right)$</td>
<td>$B$</td>
<td>$\emptyset$ to $s_8$</td>
<td>$s_9$</td>
</tr>
<tr>
<td>S:2</td>
<td>$\left(\frac{1}{2}, \frac{0}{1}, \frac{1}{2}\right)$</td>
<td>$C$</td>
<td>$\emptyset$ to $s_9$</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Our Algorithm.

We provide clarifying comments for each step in Table 3 below.
Step | Comments
1  | A wins the lottery and points to $s_1$ for its most proffered flight $f_3$.
2  | A wins the lottery and points to $s_2$ for its currently most proffered flight $f_7$. However, $s_2$ belongs to $B$. Thus, $B$ is allowed to point to $s_2$ for its currently most preferred flight $f_6$.
3  | A wins the lottery and points to $s_4$ for its most proffered flight $f_4$.
4  | C wins the lottery and points to $s_5$ for its most proffered flight $f_9$.
5  | B wins the lottery and points to $s_6$ for its currently most proffered flight $f_5$.
6  | A wins the lottery and points to $s_7$ for its currently most proffered flight $f_{10}$.
S:1 | Supplementary round begins. $B$ wins the lottery and points to $s_8$. Note that during this step there are no flights unassigned to slots left. In reality, however, this algorithm might be used consequently several times and these slots might used by some other flights.
S:2 | C wins the lottery and points to $s_9$.

Table 3. Our Algorithm (Comments).

Final allocation of slots becomes:

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_3$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_4$</td>
<td>$f_9$</td>
<td>$f_5$</td>
<td>$f_{10}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

Table 4. Outcome of the Algorithm.

6.2 Comparison with the Compression Algorithm

As we explained before, GDP starts with the RBS procedure that rations slots based on the initial schedule as reflected in Table 5 below.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_5$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_8$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
</tr>
</tbody>
</table>

Table 5. Outcome of the RBS.

The Compression algorithm allows the airlines to exchange slots they cannot use. If an airline has a slot that cannot be used by a flight assigned to this slot during RBS, the airline may assign a later flight to this slot. Crucially, it has to pick the earliest feasible flight according to RBS. For example, Company $C$ that owns slot $s_0$ had to cancel flight $f_1$. According to the Compression algorithm, the airline has to assign not the most preferred
flight but the earliest flight that this company owns that can use it! It is clear that the Compression will not accommodative the preference of the airline as it implicitly assumed the earlier the better regardless of flights. If an airline cannot use a slot by itself, it has to exchange it with another one. In particular, an airline that owns the earliest flight that can feasibly use the slot will be allowed to take it. In exchange, the first airline will get a slot that the flight of the second carrier was assigned to. Thus, in the Compression algorithm, the airline itself does not choose whom to trade with.

For convenience, we define the set of vacant slots as $V$. Initially, $V = \{s_0, s_1\}$. An active slot is $s_0$. Since $f_1$ was canceled company $C$ is allowed to pick the earliest flight that can feasibly use $s_0$ - flight $f_8$. The set of vacant slots $V$ becomes $V = \{s_7, s_1\}$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$C$</td>
<td>$A$</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_5$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_9$</td>
<td>$f_9$</td>
<td>$f_10$</td>
</tr>
</tbody>
</table>

An active slot is now $s_7$. Company $C$ is allowed to pick the earliest flight that can feasibly use $s_7$ - flight $f_9$. The set of vacant slots $V$ becomes $V = \{s_8, s_1\}$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$C$</td>
<td>$A$</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_5$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_9$</td>
<td>$f_9$</td>
<td>$f_10$</td>
</tr>
</tbody>
</table>

An active slot is $s_8$. Company $C$ is allowed to pick the earliest flight that can feasibly use $s_7$, however, there are no such flights that belong to $C$. Thus, the earliest flight that belongs to another company is chosen - flight $f_{10}$ from company $A$. The set of vacant slots $V$ becomes $V = \{s_9, s_1\}$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$A$</td>
<td>$C$</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_5$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_9$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
</tr>
</tbody>
</table>

An active slot is $s_9$. Company $C$ is allowed to pick the earliest flight that can feasibly use $s_7$, however, there are no such flights that belong to $C$. There are no flights assigned to later slots in other companies that can feasibly use $s_9$. Thus, $s_9$ is left empty and the set of vacant slots $V$ becomes $V = \{s_1\}$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$A$</td>
<td>$C$</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_5$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_9$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
</tr>
</tbody>
</table>
An active slot is $s_1$. Company $B$ is allowed to pick the earliest flight that can feasibly use $s_1$ - flight $f_5$. The set of vacant slots $V$ becomes $V = \{s_4\}$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$A$</td>
<td>$C$</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_5$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An active slot is $s_4$. Company $B$ is allowed to pick the earliest flight that can feasibly use $s_4$ - flight $f_6$. The set of vacant slots $V$ becomes $V = \{s_5\}$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$A$</td>
<td>$C$</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_5$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An active slot is $s_5$. Company $B$ is allowed to pick the earliest flight that can feasibly use $s_5$, however, there are no such flights that belong to $B$. Thus, the earliest flight that belongs to another company is chosen - flight $f_7$ from company $A$. The set of vacant slots $V$ becomes $V = \{s_6\}$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$A$</td>
<td>$C$</td>
<td></td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_5$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An active slot is $s_6$. Company $B$ is allowed to pick the earliest flight that can feasibly use $s_5$, however, there are no such flights that belong to $B$. Thus, the earliest flight that belongs to another company is chosen - flight $f_9$ from company $C$. The set of vacant slots $V$ becomes $V = \{s_7\}$.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_5$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An active slot is $s_7$. Company $C$ is allowed to pick the earliest flight that can feasibly use $s_7$, however, there are no such flights that belong to $C$. Thus, the earliest flight that belongs to another company is chosen - flight $f_{10}$ from company $A$. The set of vacant slots $V$ becomes $V = \{\emptyset\}$. In our example, the outcome of the Compression algorithm becomes the following:

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airline</td>
<td>$C$</td>
<td>$B$</td>
<td>$A$</td>
<td>$A$</td>
<td>$B$</td>
<td>$A$</td>
<td>$C$</td>
<td>$A$</td>
<td>$B$</td>
<td>$C$</td>
</tr>
<tr>
<td>Flight</td>
<td>$f_8$</td>
<td>$f_5$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

Table 6. Outcome of Compression.
We compare the outcome of our proposed mechanism and Compression in the Table below.

<table>
<thead>
<tr>
<th>Slot</th>
<th>$s_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
<th>$s_7$</th>
<th>$s_8$</th>
<th>$s_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression</td>
<td>$f_8$</td>
<td>$f_5$</td>
<td>$f_3$</td>
<td>$f_4$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_9$</td>
<td>$f_{10}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>Our Mechanism</td>
<td>$f_8$</td>
<td>$f_3$</td>
<td>$f_6$</td>
<td>$f_7$</td>
<td>$f_4$</td>
<td>$f_9$</td>
<td>$f_5$</td>
<td>$f_{10}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

Table 7. Comparison of Compression and Our Mechanism Outcome.
7 Conclusion

This paper examines the runway slot allocation problem that emerges when inclement weather conditions strike an airport. In such conditions, airports have to modify their initial schedules, as each flight will require more time to land. In addition, some flights may be canceled, and the earliest feasible arrival time may change for others. This creates challenges in matching flights with newly created slots. We argue that the current GDP program does not respect property rights, as it assigns flights to slots based on airlines’ initial schedules. This implies that airlines are not free to use the runway slots they have paid for.

We propose a mechanism that respects property rights. It allows airlines to use a newly created slot if it initially paid for the time interval. First, airlines are required to submit a priority ranking of their flights along with the earliest feasible arrival times for each flight that is not canceled. Second, each slot is assigned through a lottery in which the airlines’ chances of winning depend on the amount of initially owned slots. The winner of a lottery can point to any slot that has not yet been assigned; if the winner prefers a slot owned by another airline, a trading cycle takes place.

This mechanism satisfies factors besides property rights. It is feasible in that it assigns flights only to slots they can use. It is efficient, as no slot is left unassigned when a flight can use it feasibly. It is also strategy-proof and provides an outcome from the core.

To emphasize the importance of property rights, we assume that airlines have lexicographic preferences over flights. Although this simplifies analysis, it does not allow the study of full trade-offs or of sacrificing less costly flights to move more important flights to earlier slots. While this extension is practical and relevant, it is likely to cause certain irregularities, as the literature on many-to-one matching with general preferences has demonstrated. In future research, we plan to conduct a detailed analysis of this problem.
8 Appendix

Proof of the Theorem.

Feasibility. An each step, no flight gets an infeasible slot.

Non-wastefulness. This is by construction of the mechanism. If a flight $\exists f \in F$ such that $s \leq e_f < s$, then it must be the case $\varphi_f(I) < s$.

Ex post individual rationality. In the pre-competition stage, each $f \in F$ that can feasibly use a slot in $S_a \setminus S^1$ will be assigned to such slot; in the main stage, all slots in $S_a \cap S^1$ will be endowed to $a$’s flights in a fashion that favors more important flights. So $\Pi^S_a$ for each $a \in A$ is constructed.

For any realized $R_a$, at each step that an endowed slot is being assigned, an airline can either use the slot on that flight or trade it for a better slot for its remaining most important flight. Let $x_i = d_{f_{a,i}}(\Pi^S_a) - d_{f_{a,i}}(\varphi^{R_S}_a(I))$ for $i \in \{1, \ldots, |F_a|\}$ and $f_{a,i}R_a f_{a,i+1}$. When an airline use an endowed slot, $x_i = 0$ for some $i$, and when the first time an airline (i) trades an endowed slot with some other slot or (ii) pick an empty slot, $x_i > 0$, which would be the first non-zero coordinate of $x_a = (x_1, x_2, \ldots, x_{|F_a|})$. Therefore, $\forall a \in A$, $\varphi^{R_S}_a(I) \succ_a \Pi^S_a$.

Ex post Pareto efficiency. Flights that leave at the pre-competition stage are already getting the earliest slot they can get, and no slot in $S \setminus S^1$ can be used to make flights leave at later steps better off.

Consider the main stage, any flight that leaves at step 1 is assigned its top choice that is available and cannot be made better off. Any flight that leaves at step 2 is assigned its top choice that is available among those slots remaining at Step 2 and since slots are distinct time intervals, it cannot be made better off without hurting some flight who left at Step 1. Proceeding in a similar way, no flight can be made better off without hurting some flight that left at an earlier step.

Moreover, for an airline, a flight left at an earlier step is more important than a flight left later, so it can not make itself better off as well. Therefore, $\varphi$ is ex post Pareto efficient.

Respecting property rights. In each step an endowed slot is being assigned, if it is not assigned to the airline, that means the airline trades it for some better slot. In particular, in the pre-competition stage, the airline trades an infeasible slot for some feasible slot. In the main stage, it trades it for a better slot for its remaining most important flight. In the supplemental stage, an airline will not get its endowed slot only if it has zero budget at that step, which means it traded it for some better slot already.

Core-selecting. Suppose $\exists \Pi'$ and $A' \subseteq A$ such that (i) $\forall f \in \bigcup_{a \in A'} F_a$, $\Pi'(f) \in S_{A'}$, and (ii) $\forall a \in A'$, $\Pi' \succ_a \varphi(I)$. Therefore, $\forall a \in A'$, the first non-zero coordinate of $x_a =$
\((x_1, x_2, \ldots, x_{|F_a|})\) is positive, where for \(i \in \{1, \ldots, |F_a|\}\) and \(f_{a,i}R_afx_{a,i+1}\), \(x_i = d_{f_{a,i}}(\varphi^{R_S}(I)) - d_{f_{a,i}}(\Pi')\) for some \(R_s\).

Consider \(f_{a,i}\) where \(x_i\) is the first non-zero coordinate of \(x_a\) for \(a \in A'\). Note that \(\Pi'(f_{a,i})\) is earlier than \(\varphi^{R_S}_{f_{a,i}}(I)\), \(\Pi'(f_{a,i}) \in S_{A'}\), \(\Pi'(f_{a,i})\) is not available when \(f_{a,i}\) is picking a slot in \(\varphi\). There is a \(\Pi'(f_{a,i})\) for each \(a \in A'\); let \(S_T\) be the collection of \(\Pi'(f_{a,i})\) for all \(a \in A'\). \(S_T\) is the set of slots that make airlines in \(A'\) prefer \(\Pi'\).

(i) If \(a\) is the owner and \(\Pi'(f_{a,i})\) is used by some \(f_{a,j}\) in \(\varphi^{R_S}\), then it must be \(f_{a,j}R_afx_{a,i}\). Since \(x_i\) is the first non-zero coordinate, then \(x_j = 0\), i.e., \(f_{a,j}\) is getting the same slot under \(\Pi'\), a contradiction.

The same argument applies to all airline in \(A'\). Therefore, \(\forall a \in A'\), \(\Pi'(f_{a,i})\) is coming from some airline \(a' \in A'\) with \(a' \neq a\).

Let \(s_a \in S_T \subseteq S_a \cap S^1\) endowed to airline \(a \in A'\) be the first slot being assigned to some \(f\) in \(\varphi^{R_S}\). \(a\) will pick a slot for its highest ranked remaining flight \(f_{a,j}\) before \(s_a\) is assigned. At this step, all slots in \(S_T\) are available (otherwise it contradicts the way we pick \(s_a\)).

(ii) If \(f_{a,i} = f_{a,j}\), \(a\) picks \(\varphi^{R_S}_{f_{a,i}}(I)\) but not \(\Pi'(f_{a,i})\), a contradiction.

(iii) If \(f_{a,i}R_afx_{a,j}\), this means \(\Pi'(f_{a,i})\) is still available after \(f_{a,i}\) picked a slot in \(\varphi^{R_S}\), a contradiction.

(iv) If \(f_{a,j}R_afx_{a,i}\), we have \(\varphi^{R_S}_{f_{a,j}}(I) = \Pi'(f_{a,j}) \in S_{A'}\).

If \(\varphi_{f_{a,j}}(I) \in S_a\), then \(a\) will pick another slot. Let the last flight of \(a\) gets a slot be \(f_{a,k}\). If \(s_a\) is picked by some flight of \(a\) at this step, then this flight must be higher ranked than \(f_{a,i}\) because \(\Pi'(f_{a,i})\) is still available (a result from (i)), \(\Pi'(f_{a,i}) \notin S_a\). Then \(s_a\) will be used by \(a\) under \(\Pi'\) but not another airline, a contradiction.

So \(s_a\) is not used by \(a\) and \(f_{a,k}\) is higher ranked than \(f_{a,i}\). That means airline \(a\) trades \(s_a\) for \(\varphi^{R_S}_{f_{a,k}}(I) \in S_{A'}\) from some airline \(b \in A'\). Let \(\varphi^{R_S}_{f_{b,j}}(I)\) be the slot obtained by \(b\) in this trade. Because all slots in \(S_T\) are still available, the flight \(f_{b,j}\) is higher ranked than \(f_{b,i}\), so \(\varphi^{R_S}_{f_{b,j}}(I) = \Pi'(f_{b,j}) \in S_{A'}\). If \(\varphi^{R_S}_{f_{b,j}}(I) \in S_a\), we have a cycle.

If \(\varphi^{R_S}_{f_{b,j}}(I) \in S_c, c \in A'\) will be the next airline in line of the trade, and the same argument * applies. Because none of the airline in line gets a slot outside \(S_{A'}\) and \(A'\) is finite, there must exist a cycle contains exclusively airlines in \(A'\). Let \(z \in A'\) be the airlines gets \(s_a\) for \(f_{z,j}\), again, since all slots in \(S_T\) are still available, \(f_{z,j}\) is higher ranked than \(f_{z,i}\) and therefore \(\varphi^{R_S}_{f_{z,j}}(I) = \Pi'(f_{z,j}) = s_a\). This contradicts the fact \(s_a\) makes some airline prefer \(\Pi'\) to \(\varphi^{R_S}(I)\).

**Strategyproofness.** We consider each stage to see if an airline \(a\) can manipulate its outcome by misreporting \(R_a\) and \(e_a\) such that the final landing schedule \(\varphi(\widehat{R_a, e_a}, (R, e)_{-a})\) will make \(a\) weakly better off, that is, \(\varphi(\widehat{R_a, e_a}, (R, e)_{-a}) \succ_a \varphi_a(R, e))\).

Let \(x_a = (x_1, x_2, \ldots, x_{|F_a|})\) be a vector such that for \(i \in \{1, \ldots, |F_a|\}\) and \(f_{a,i}R_afx_{a,i+1}\), \(x_i = d_{f_{a,i}}(\varphi^{R_S}(R, e)) - d_{f_{a,i}}(\varphi^{R_S}(\widehat{R_a, e_a}, (R, e)_{-a}))\) for some \(R_s\).
Let $x_j$ be the first non-zero coordinate. If there is no such $x_j$, then $\varphi^R_S(\widehat{R_a}, e_a, (R, e)) \sim_a \varphi^R_S(R, e)$, and we are done. Suppose not. This implies $e_{fa,j} \leq \varphi^R_S(\widehat{R_a}, e_a, (R, e)) < \varphi^R_S(R, e)$.

In the supplemental stage, if $f_{a,j}$ wants a slot that is being assigned in this stage, it will get that slot in the main stage of $\varphi$, a contradiction.

In the pre-competition stage, $R$ is not used. The only way that $f_{a,j}$ can get a slot (that is feasible for $f_{a,j}$ and is not being assigned in this stage under $\varphi$) is when $a$ is able to make $e_{fa,j} \leq s$ can be only used by $f_{a,j}$.

Note that $\varphi_{fa,k}(\widehat{R_a}, e_a, (R, e)_a) = \varphi_{fa,k}(R, e)$ for all $f_{a,k}Raf_{a,j}$. So this can happen only when the earliest slot $s' \in S_1$ is demanded by $a$ with $f_{a,x}', f_{a,x''}, \ldots$ when $s' = \varphi_{fa,x}(R, e)$ for $f_{a,x}'Raf_{a,x}$ and $f_{a,j}Raf_{a,x}$, ... or by $a$ with $f_{a,x}', f_{a,x''}, \ldots$ for $f_{a,j}Raf_{a,x}', f_{a,x''}, \ldots$ and at most one airline $b$ with $f_{b,x}$, then in the the prior case, $a$ can misreport (infeasible or later times) $e_{fa,x'}, e_{fa,x''}, \ldots$ to give that slot to $f_{a,k}$ (otherwise, this slot will not go to $f_{a,k}$ until the main stage), and in the latter case, $a$ can also misreport $e_{fa,x'}, e_{fa,x''}, \ldots$ to give that slot to $f_{b,x}$; again, the next demanded slot is demanded by $a$ with $f_{a,y}', f_{a,y''}, \ldots$ when $s' = \varphi_{fa,y}(R, e)$ for $f_{a,y}'Raf_{a,j}$ and $f_{a,j}Raf_{a,y}$, ... or by $a$ with $f_{a,y}', f_{a,y''}, \ldots$ for $f_{a,j}Raf_{a,y}', f_{a,y''}, \ldots$ and at most one airline $c$ with $f_{c,y}$.

$f_{a,j}$ will get $s$ from $\varphi$’s main stage because the only competitors $f_{b,x}, f_{c,x} \ldots$ will always try to get a slot earlier than $s$ and flights $f_{a,x}', \ldots, f_{a,y}', \ldots$ will not compete with them before $f_{a,j}$ gets a slot. So this contradicts $x_j$ is the first non-zero coordinate.

Now consider the main stage, we want $a(j)$ to pick a slot earlier such that $x_j > 0$. Again, $\varphi^R_S(\widehat{R_a}, e_a, (R, e)_a) = \varphi^R_S(R, e)$ for all $f_{a,k}Raf_{a,j}$ and some $R_S$.

$a(k) > a(j)$ obviously cannot help $a(j)$ to pick a better slot.

The only way $a$ can put $a(j)$ into $R_S$ is to let $a(j - 1)$ pick an endowed slot, but such slot must be the same as $\varphi^R_S(\widehat{R_a}, e_a, (R, e))$, the $a(j)$ will get the slot it can get from $\varphi$, a contradiction. Q.E.D.
9 References


