Airport Slot Allocation Problems*

Ken C. Ho† (Job Market Paper)
Alexander Rodivilov‡

October 1, 2017

(please click here for the latest version)

Abstract

We study airport slot allocation problems during weather-induced congestion. These real-life matching problems are important to airlines as the costs of delays are significant compared to their profits. We introduce a new mechanism, Multiple Trading Cycles (MTC), to allocate landing slots. In contrast to the currently used mechanism, MTC is individually rational, Pareto efficient, strategy-proof, non-manipulable by postponing a flight cancellation, and respects property rights over slots. In addition, it is core-selecting when preferences are strict. The "You Request My House - I Get Your Turn" mechanism (Abdulkadiroğlu and Sönmez, 1999) is a special case of MTC.

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

Keywords: Slot allocation, Strategy-proofness, Mechanism design, Top Trading Cycles

*We are indebted to Quan Wen for his continuous guidance, advice, and support. We thank Al Roth, Jacques Lawarrée, Fahad Khalil, Yoram Barzel, David Miller, and participants at the University of Washington Microeconomics Seminar and the 2017 International Conference on Game Theory (Stony Brook).

†Department of Economics, University of Washington, Seattle, WA 98195, USA. E-mail: kh89@uw.edu.

‡School of Business, Stevens Institute of Technology, 1 Castle Point on Hudson, Hoboken, NJ 07030, USA. E-mail: arodivil@stevens.edu.

1 Introduction

Although landing schedules are made in advance, unpredictable conditions may lead to reallocations of landing slots as the arrival capacity is often reduced below the number of initially scheduled flights.[1] During severe weather such as thunderstorms, low cloud ceilings, or snows, arrival capacity at the affected airport is reduced as it requires more time to land a plane.[2] This is a significant economic problem: Weather-caused flight delays cost billions of dollars every year in the United States.[3] Even though such delays are inevitable, their effect can be mitigated by moving flights into earlier slots that have been vacated by canceled or delayed flights. In the United States, a centralized mechanism performs such reallocations to create a new landing schedule by utilizing relevant flight information reported by the airlines.

Whether a flight is canceled or delayed is privately known by the airline and is not known by the centralized mechanism until the airline reports this information to it. Vacant slots are valuable resources, so airlines' strategic behaviors may affect the efficiency of the reallocations. A well-designed mechanism that provides incentives for airlines to report delays and cancellations promptly is required. In this paper, we formalize the reallocation problem as a generalization of housing allocation with existing tenants problem (Abdulkadiroğlu and Sönmez, 1999) and propose a new mechanism to reallocate landing slots. In contrast to the currently used mechanism, ours gives strong incentives for airlines to report flight delays and cancellations.

Currently, in the United States, the Federal Aviation Administration (FAA) implements Ground Delay Programs to reallocate landing slots. When severe weather is forecasted (typically hours in advance), the FAA declares a ground delay program (GDP) is in effect, where the duration of the GDP is also specified (usually several hours). In a GDP, the FAA assigns new arrival times (landing slots) to aircraft departing airports in the contiguous U.S. and Canada to the affected airport. These aircraft are also assigned new (and usually delayed) departure times at their origin airports (while the aircraft are still on the ground).[4]

[1] A slot at an airport is essentially a time interval that allows an airline to land a plane.

[2] Unpredictable conditions are not limited to severe weather conditions. For instance, runway closures caused by aircraft incidents are also included. Our model can also be applied in this situation, see the last part of footnote 19 and the last part of footnote 24.

[3] The total cost of all flight delays in 2007 was estimated at 31.2 billion dollars, in which 8.3 billion dollars were costs to airlines (Ball et al., 2010). In this particular year, the weather's share of total delay minutes is 43.6% (it varies from 32.8%-49.7% in the period of 2004–2016. See https://www.rita.dot.gov/bts/help/aviation/html/understanding.html).

The costs of weather-caused flight delays are nontrivial to airlines as their profits were just 5 billion dollars (Air Transport Association, 2008) in the same year.

[4] These new departures times are calculated based on the new arrival times. For more details, see Section 17-9-1 of the *Facility Operation and Administration*. This document is available at https://www.faa.gov/air_traffic/publications.

We now describe the relevant details of a current GDP: The FAA adopts a 2-step procedure to carry out the reassignment of landing slots. The first step is to assign slots to flights. The current algorithm is called Ration-by-Schedule (RBS). It orders flights by increasing original scheduled time of arrival and then assigns slots sequentially. That is, the first flight is assigned to the first available GDP slot, the second flight is assigned to the second available GDP slot, etc.[5] RBS may assign slots to flights that have been canceled or delayed by their airlines and consequently cannot feasibly use their assigned slots. The airlines can adjust their schedule by substitutions and cancellations, but flight cancellations and delays may consequently create vacant slots in the landing schedule. The second step is to reassign these newly created vacant slots to airlines that can use them. The mechanism in place now is called Compression.[6] Compression uses information reported by the airlines (cancellations and earliest feasible arrival times of flights) to exchanges slots among them to produce a new landing schedule. Essentially, when an airline cannot use a slot, Compression exchanges it with a later slot that is owned by some airline that can use the original slot.[7] In the current GDP, the first assignment is created by running RBS and possible Compression. But as airlines update their information, Compression might be run multiple times.

Before RBS was adopted, the FAA used a mechanism called Grover-Jack, which assigns slots based on feasible departure times reported by the airlines. This mechanism suffers from a problem called "Double Penalty" that gives incentives for airlines to hide private information.[8] By using the originally scheduled times of arrival instead of the reported feasible departure times to allocate slots, the double penalty problem was resolved. Therefore, it is crucial for any replacement of RBS to avoid the double penalty problem.

We now describe deficiencies of RBS and Compression that motivate our analysis. We show that RBS does not respect a form of property rights *before* a GDP starts. Note that slots of different lengths are different objects. A GDP converts initial slots into GDP slots, but such conversion is just a re-division of time intervals. Under RBS, owning an early initial slot gives the airline an early GDP slot, which is not the same object it had at the beginning, while such time interval (of the GDP slot) might be entirely owned by another airline before the start of a GDP. A mechanism that respects property rights before a GDP starts would

[5]The OAG schedule is considered to be the initial schedule in the industry.

[6]So the currently used mechanism is a combination of RBS and Compression.

[7]Compression moves flights up in the schedule to fill those vacant slots. For more details, see Schummer and Vohra (2013) or Vossen and Ball (2006a).

[8]Feasible departure times are private information of the airlines. Suppose a flight needs to delay its departure for 1 hour because of some mechanical issues. If a ground delay program (with Grover-Jack) is implemented in which the flight would be delayed for another hour, then its total delay would be 2 hours. This is known as the "Double Penalty" problem. If the airline had withheld the information, this flight would have been assigned a slot 1 hour earlier, which it could feasibly use in this example. As a result, airlines may intentionally withhold information to avoid double penalties.

endow a GDP slot to the airline that owns the entire time interval of this GDP slot before a GDP starts. RBS might lead to outcomes that are not individually rational and thus outside the core in our model. We show that Compression does not respect a form of property rights *after* a GDP starts (in reassignments).[9] Also, Compression produces outcomes that might be outside the core and is manipulable by postponing a flight cancellation (Schummer and Vohra, 2013).[10] Moreover, it is not strategy-proof (Schummer and Abizada, 2017).[11] We show that these negative results for Compression also hold in our preference domain. In addition, we show that Compression is not individually rational and not Pareto efficient.[12]

We propose a new mechanism, Multiple Trading Cycles (MTC), that will overcome the aforementioned problems. In our model, airlines have lexicographic preferences, in which each airline has an importance ranking over its flights. MTC solicits importance rankings and earliest feasible arrival times of flights from airlines. If the time interval of a slot is entirely owned by some airline, then this slot is considered to be owned by the airline in MTC. There are three stages in MTC. In the first stage, each slot that is demanded by only 1 airline (in a sense we make precise below) will be identified and assigned. In the second stage, all slots being assigned are demanded by more than 1 airline. According to an ordering determined by MTC, an airline picks a slot for its most important remaining flight or picks a slot pro forma (if it has no remaining flight). If the airline picks a slot that is owned by another airline, then the latter can pick a slot for its most important remaining flight or pick a slot pro forma. If a cycle forms, each airline in the cycle picks a slot for its most important remaining flight or pick a slot pro forma. This stage finishes when each non-canceled flight has been assigned a slot. In last stage, each airline that has canceled flights will get the same number of slots. (The last stage is consistent with RBS. MTC can also be used to perform reassignments, and the slots assigned in the last stage might be valuable in a subsequent reassignment.)

We now highlight some desirable properties of MTC. MTC avoids the double penalty problem as it does not use reported feasible departure times to allocate slots. In contrast to RBS and Compression, MTC respects a form of property rights before and after a GDP starts, produces outcomes that are individually rational, Pareto efficient and in the core (in this problem, the core might not be a subset of the Pareto set), is strategy-proof and non-manipulable by postponing a flight cancellation. By truth-telling, MTC minimizes the

[9] Schummer and Vohra (2013) define property rights based on core allocations from an initial endowment of landing slots and claim that Compression does not respect such a form of property rights.

[10] Indeed, they show Compression fails a condition that is weaker than non-manipulable by postponing a flight cancellation.

[11] Schummer and Vohra (2013) show that Compression is strategy-proof in their preference domain.

[12] Schummer and Vohra (2013) show that Compression is Pareto efficient in their preference domain.

expected delays for each airline lexicographically. That is, for each airline, it minimizes the expected delay for the most important flight then minimizes the expected delay for the second most important flight, and so on.[13]

We provide two algorithms to find the outcome of MTC. We also provide a modified version of MTC called Multiple Trading Cycles-2. MTC-2 has all the desirable properties of MTC while it possibly favors some airlines. Finally, we extend our model to allow indifferences in preferences. In that extended model, a modified version of MTC with tiebreaking inherits most of the desirable properties while it possibly produces outcomes outside the core as agents might be in more than one cycle under their true preferences, where one of the cycles might lead to a better outcome for the agents (this does not happen in models where agents have unit demand and non-strict preferences).

Our mechanism might be useful in other applications. For example, when a set of objects (or tasks) is being distributed to several teams and team members have heterogeneous preferences, given each team has an internal ranking over its members, a slight modification of MTC can be used in this environment.[14]

2 Related Literature

The two papers most related to ours are Schummer and Vohra (2013) and Schummer and Abizada (2017). Importantly, both papers take RBS outcomes as initial endowments and focus on the reassignment step. In Schummer and Vohra (2013), preferences are incomplete as not every pair of feasible landing schedules is comparable; they propose a mechanism called Trading Cycle (TC).[15] In Schummer and Abizada (2017), the preference domain is larger

[13]RBS minimizes the delay for each flight in a lexicographic order. It is easy to see it minimizes the delay for the first flight then minimizes the delay for the second flight, and so on. Vossen and Ball (2006a) show RBS lexicographically minimize the maximum delay with respect to the original schedule. Their formulation is different, but the intuition is similar.

[14]Two more applications are also related to our model (they are also mentioned in Schummer and Vohra (2013)). One is when agents have multiple jobs/tests/orders that need to be processed sequentially by a number of servers/laboratories/warehouse. There might be constraints on submission times if these jobs cannot be prepared for processing before a certain date. Also, the rankings can vary, e.g., when a customer (of some agent) pays an expedite fee or cancel order. The other application is related to geographic fairness in deceased organ donation. Suppose under some fair policy, each region receives a fixed percentage of deceased organs. Agents are regions with queued patients. Instead of time constraints, there might be other considerations here (for example, tissue rejection or blood-type incompatibility). Regions' rankings over patients can vary as well, e.g., when some patient's condition deteriorates quickly or some patient dies.

[15]They assume an airline is made better off only if it moves a flight up in the schedule while no others move down, so airlines' preferences are induced by feasible arrival times and the current landing schedule. The current landing schedule is preference-incomparable with another landing schedule that puts some of airline a's flights in earlier positions and some in later positions.

In this paper and Schummer and Abizada (2017), an airline might be made better off even if a flight is moved down.

than ours since they allow airlines to put arbitrary weights on flights, while lexicographic preference assumes that the weight of a flight is infinitesimal compared to the weight of a more important flight. They separately consider airlines' incentives to report flights' feasible arrival times, relative delay costs (weights), and cancellations. They propose a mechanism called Deferred Acceptance with Self Optimization (DASO). DASO is not Pareto efficient (this is because DASO does not use weights, which is necessary to determine Pareto efficient outcomes). DASO is non-manipulable via weights and non-manipulable by postponing a flight cancellation, though it is still manipulable by intentional flight delay. By contrast, our mechanism achieves full strategy-proofness (non-manipulable via feasible arrival times and rankings) and Pareto efficiency in a smaller preference domain. Furthermore, MTC is also non-manipulable by postponing a flight cancellation.

Another related paper is Abdulkadiroğlu and Sönmez (1999). Indeed, when (i) no airline owns a canceled flight, (ii) each airline owns exactly one non-canceled flight, and (iii) each airline owns at most one GDP slot, our model degenerates to the housing allocation with existing tenants model, and MTC reduces to YGMH-IGYT (with random ordering).[16] Given (ii) and (iv) no airline owns a GDP slot, our model reduces to a house allocation problem (Hylland and Zeckhauser, 1979), and MTC reduces to random serial dictatorship. Given (i), (ii), (v) each airline owns exactly one GDP slot, and (vi) each slot is owned by some airline, our model reduces to a housing market (Shapley and Scarf, 1974), and MTC reduces to the core mechanism.[17]

Kurino (2014); Kennes et al. (2014); Pereyra (2013) study dynamic object allocation problems with overlapping generations in house assignments, daycare assignments, and teacher assignments, respectively. They also propose mechanisms that respect the property rights over the objects induced by the allocation in the previous period.

Konishi et al. (2001) also generalize the housing market. In their model, multiple types of indivisible goods are traded. They show that the core may be empty and there is no Pareto efficient, individually rational, and strategy-proof (deterministic) mechanism. In our context, we obtain positive results on these properties for MTC (which is stochastic). Chun and Park (2017) study a slot allocation problem assuming monetary transfers are feasible.

[16]With unit demand, agents' preferences are trivially lexicographic.

(i) is non-trivial when some airline owns a GDP slot. Without (i), when some airline owns a GDP slot, the airline has the ability to remove it from the set of available GDP slots (see Section 6.1 for more details), and such feature is absent in the housing allocation with existing tenants model.

This mechanism is called Top Trading Cycle mechanism in Abdulkadiroğlu and Sönmez (1999); Sönmez and Ünver (2005) and YGMH-IGYT mechanism in Sönmez and Ünver (2010).

[17]In a housing market with strict preferences, there is a unique matching in the core (Roth and Postlewaite, 1977), and Gale's top trading cycles algorithm (attributed to David Gale by Shapley and Scarf, 1974) can be used to find the outcome of the core mechanism.

(i), (ii), (v) and (vi) imply that the number of flights equals the number of slots.

By contrast, we assume monetary transfers are infeasible.

In the transportation literature on GDP, optimization is the main focus.[18] Vossen (2002) proposes a proportional random assignment method, in which each flight is entitled to an equal share of each slot it can use. Balakrishnan (2007) uses the housing market model by treating flights as agents. These two papers do not take airlines' incentives into account. Ball et al. (2010) propose an algorithm called Ration-by-Distance that assigns slots to flights based on distance. They show that Ration-by-Distance minimizes total expected delay, while MTC minimizes the expected delays for each airline lexicographically.

The rest of the paper is organized as follows: Section 3 introduces the model; Section 4 illustrates the mechanism; Section 5 shows properties of the mechanism; Section 6 discusses extensions; and Section 7 concludes the paper. All proofs are in Appendix A, a summary of properties is in Appendix B, and two examples for Compression are in Appendix C.

3 Model

There is a finite set of airlines A and a finite set of flights $F^o = \cup_{a \in A} F^o_a$, where F^o_a is the set of flights owned by airline a. Some flights might be canceled during the GDP or before the GDP starts; we use $F \subseteq F^o$ to denote the set of non-canceled flights and $F_a \subseteq F^o_a$ to denote the set of flights owned by airline a that are not canceled. There is a set of initial slots $S^o = \{s^o_1, s^o_2, ..., s^o_{|L|}\}$, where the length of each slot is normalized to one unit of time. Note that $|F^o|$ of the $|L|$ initial slots were owned by some airlines. Let the set of available GDP slots be $S = \{s_1, s_2, ...\}$, where the length of each slot is l units of time ($l > 1$).[19]

There is an earliest feasible arrival time $e_f \in S$ for each flight $f \in F$, and f can be feasibly assigned to a slot s_n only if $e_f \leq s_n$.[20] With a slight abuse of notation, we use n instead of s_n to express e_f in our examples. Let $e = (e_f)_{f \in F}$ be the vector of all earliest feasible arrival times and $e_a = (e_f)_{f \in F_a}$ be the vector of airline a's earliest feasible arrival times. A **landing schedule** is an injective function $\Pi : F \to S$ that assigns each flight to a

[18]See Vossen and Ball (2006a,b); Bard and Mohan (2008); Ball et al. (2009); Glover and Ball (2013).

[19]Note that available GDP slots do not have to be adjacent since there are exempted flights in a GDP (for example, international and airborne flights are exempted). There will always be feasible slots for flights, yet they might be late and outside the GDP time window. The number of arrivals a runway can accept (per 60 minutes) is called maximum runway arrival capacity. If 1 unit of time is 1 minute and the maximum runway arrival capacity (MRAC) during a GDP is 30, then $l = \frac{60}{MRAC \times 1\, unit\, of\, time} = \frac{60}{30 \times 1} = 2$.

There is a single runway in our model. See Section 6.2 for an extended model with multiple runways, in which there are multiple slots available at a time.

When a GDP is implemented because of runway closures caused by aircraft incidents but not severe weather, $l = 1$ is possible. However, all results in this paper hold for $l = 1$.

[20]For $n = 1, 2, ...,$ slot s_n is the time interval $[1 + (n-1)l, 1 + nl]$, where "1" represents the starting time of s^o_1. We use s_n to represent $1 + (n-1)l$ on the time line. Following the literature, we call e_f the earliest feasible arrival *time* of f, but strictly speaking, e_f is the earliest feasible arrival *slot* of f.

landing slot. Let $\mathcal{M}$ be the set of all landing schedules. A landing schedule Π is **feasible** if $\Pi(f) \geq e_f$ for all $f \in F$. A landing schedule Π is **non-wasteful** if $\nexists f \in F$ and $s \in S$ such that $\Pi^{-1}(s) = \emptyset$ and $e_f \leq s < \Pi(f)$.

An initial landing schedule is an injective function $\Pi^o : F \to S^o$. Given some initial landing schedule Π^o, one can infer slot $\Pi^o(f)$ is initially endowed to airline a if $f \in F_a^o$. Let $\hat{S} \in \{S^o, S\}$. When a subset of initial slots or a subset of GDP slots is owned by some airline, we use the following function to describe. $\Phi(A, \hat{S}) : A \to 2^{\hat{S}}$ is a **slot ownership function** such that $a \neq a' \implies \Phi(a, \hat{S}) \cap \Phi(a', \hat{S}) = \emptyset$, where $\Phi(a, \hat{S})$ is the subset of slots (in $\hat{S}$) owned by airline a. $\Phi(A, \hat{S})$ is consistent with a landing schedule Π when occupation (under Π) implies ownership (under $\Phi(A, \hat{S})$): $\forall a \in A$, $\forall f \in F_a$, $\Pi(f) \in \Phi(a, \hat{S})$. A pair $(\Pi, \Phi(A, \hat{S}))$ that satisfies this consistency condition is called an **assignment**. Given an initial assignment $(\Pi^o, \Phi(A, S^o))$, the set of initial slots owned by airline a is $\Phi(a, S^o)$.

3.1 Preferences

An airline's preference over landing schedules might be induced by numbers passengers on flights, future needs of aircraft, flights' operating costs, deadlines for crews timing out, etc. As pointed out in Schummer and Abizada (2017), it may be impractical for airlines to evaluate and report their full preferences over landing schedules as such complex information is unique to every GDP. To simplify the problem, we assume airlines have lexicographic preferences, so each airline $a \in A$ has an importance ranking over its flights.[21] Formally, let R_a be a strict total order over F_a. If $f \in F_a$ is more important than $f' \in F_a$, we write $f R_a f'$. Let $R = (R_a)_{a \in A}$ be the importance ranking profile.

All else being equal, airline a prefers flight $f \in F_a$ to land as early as possible (but not earlier than e_f). Given a landing schedule $\Pi \in \mathcal{M}$, we define the **delay** for each flight f by $d_f(\Pi) = \Pi(f) - e_f$, where $\Pi(f)$ is the slot assigned to f in Π.[22]

Airline a's preference over *feasible* landing schedules is induced by R_a and e_a. For any feasible landing schedules Π and Π', airline a (lexicographically) prefers Π to Π' if and only if the first non-zero coordinate of $x_a = (x_1, x_2, ..., x_{|F_a|})$ is positive, where $x_i = d_{f_{a,i}}(\Pi') - d_{f_{a,i}}(\Pi)$ for $i \in \{1, ..., |F_a|\}$ and $f_{a,i} R_a f_{a,i+1}$, and we write $\Pi \succ_a \Pi'$. Conversely, if the first non-zero coordinate of x_a is negative, Π' is preferred to Π. If airline a is indifferent between Π and Π',

[21] In the two-sided many-to-one matching literature, Dutta and Massó (1997) study a model where the one side agents have lexicographic preferences, and Abizada and Dur (2017) study a model with complementarities where the many side agents have lexicographic preferences. Schulman and Vazirani (2012) and Saban and Sethuraman (2014) study allocation of divisible goods under lexicographic preferences. Fujita et al. (2015) study exchange with multiple indivisible goods under lexicographic preferences. Ehlers (2002) and Ehlers (2003) study locating multiple public goods when agents have lexicographic preferences.

[22] Note that the delay of a flight f is with respect to e_f but not its original slot $\Pi^o(f)$.

we write $\Pi \sim_a \Pi'$; this happens when all coordinates of x_a equal to 0. Let $\Pi_a : F_a \to \Phi(a, S)$ denote a **landing schedule for** a. $\Pi \sim_a \Pi'$ implies $\Pi_a = \Pi'_a$.[23] Since airlines only care about their own flights, we will also use $\succsim_a$ to compare landing schedules for a.

A **schedule lottery** is a probability distribution over the set of all landing schedules $\mathcal{M}$. Let $\Delta\mathcal{M}$ denote the set of all schedule lotteries. We denote a schedule lottery by $\mathcal{L} = \sum p_\Pi \cdot \Pi$ where $p_\Pi \in [0, 1]$ is the probability weight of landing schedule Π and $\sum_\Pi p_\Pi = 1$. We now extend an airline's preference to allow it to compare schedule lotteries. Given a schedule lottery $\mathcal{L} \in \Delta\mathcal{M}$, the **expected delay** for f is $d_f(\mathcal{L}) = \sum_\Pi p_\Pi \cdot (\Pi(f) - e_f)$. For any schedule lotteries $\mathcal{L}$ and $\mathcal{L}'$, $\mathcal{L} \succ_a \mathcal{L}'$ if and only if the first non-zero coordinate of $x_a = (x_1, x_2, ..., x_{|F_a|})$ is positive, where $x_i = d_{f_{a,i}}(\mathcal{L}') - d_{f_{a,i}}(\mathcal{L})$ for $i \in \{1, ..., |F_a|\}$ and $f_{a,i} R_a f_{a,i+1}$; other cases are the same as above.

Let $\succsim = (\succsim_a)_{a \in A}$ be the preference profile of all airlines. An **instance** of an Airport Slot Allocation Problem is a tuple $I = (S, A, F^o, R, e, \Phi(A, \hat{S}))$. An instance is not equivalent to an airport slot allocation problem for two reasons. Firstly, there can be multiple instances in an airport slot allocation problem (Section 4.3). Secondly, an airport slot allocation problem includes contents that are outside an instance (Section 6.1). A GDP is an airport slot allocation problem but the opposite is not true because airport slot allocation problems subsume problems that are different from GDPs (as discussed in Section 2).

3.2 Mechanisms and Their Properties

In an instance I, parameters other than R and e are fixed. R and e will be reported by the airlines. A *(direct) schedule mechanism* $\varphi : (\boldsymbol{R}, \boldsymbol{e}) \to \mathcal{M}$ is a mapping that selects a landing schedule for every strategy profile $(\boldsymbol{R}, \boldsymbol{e})$. Let $\varphi_f(\boldsymbol{R}, \boldsymbol{e})$ be the slot that is assigned to f in $\varphi(\boldsymbol{R}, \boldsymbol{e})$, and $\varphi_a(\boldsymbol{R}, \boldsymbol{e})$ be the landing schedule for a in $\varphi(\boldsymbol{R}, \boldsymbol{e})$. A *(direct) lottery mechanism* $\phi : (\boldsymbol{R}, \boldsymbol{e}) \to \Delta\mathcal{M}$ is a mapping that selects a schedule lottery for every strategy profile $(\boldsymbol{R}, \boldsymbol{e})$. $\phi_f(\boldsymbol{R}, \boldsymbol{e})$ and $\phi_a(\boldsymbol{R}, \boldsymbol{e})$ are defined analogously. The strategy space for airline a is $\mathcal{R}_a \times S^{|F_a|}$, where $\mathcal{R}_a$ is the set of strict total orders over F_a and $S^{|F_a|}$ is the vector space of airline a's earliest feasible arrival times.

A schedule mechanism φ is *regular* if for any strategy profile $(\boldsymbol{R}, \boldsymbol{e})$, the induced ownership function $\Phi^{\varphi(\boldsymbol{R}, \boldsymbol{e})}(A, S)$ is consistent with $\varphi(\boldsymbol{R}, \boldsymbol{e})$. Let $\phi\utilde{(\boldsymbol{R}, \boldsymbol{e})}$ be a realized landing schedule of the schedule lottery $\phi(\boldsymbol{R}, \boldsymbol{e})$. A lottery mechanism ϕ is regular if for any strategy profile $(\boldsymbol{R}, \boldsymbol{e})$ and any realization $\phi\utilde{(\boldsymbol{R}, \boldsymbol{e})}$, the induced ownership function $\Phi^{\phi\utilde{(\boldsymbol{R}, \boldsymbol{e})}}(A, S)$ is consistent with $\phi\utilde{(\boldsymbol{R}, \boldsymbol{e})}$. A schedule mechanism φ is *feasible (non-wasteful)* if for any strategy profile $(\boldsymbol{R}, \boldsymbol{e})$, $\varphi(\boldsymbol{R}, \boldsymbol{e})$ is feasible (non-wasteful).

[23] When there are multiple runways (as in Section 6.2), $\Pi \sim_a \Pi'$ implies Π_a is effectively the same as Π'_a.

A schedule mechanism or lottery mechanism is **strategy-proof** if truth-telling is a dominant strategy in its induced preference revelation game. A landing schedule Π is **Pareto efficient** if $\nexists \Pi'$ such that (i) $\forall a \in A$, $\Pi' \succsim_a \Pi$, and (ii) $\exists a \in A$, $\Pi' \succ_a \Pi$. The set of Pareto efficient landing schedules is the **Pareto set**. A schedule mechanism φ is *Pareto efficient* if for any strategy profile $(\boldsymbol{R}, \boldsymbol{e})$, $\varphi(\boldsymbol{R}, \boldsymbol{e})$ is Pareto efficient.

Let $\Phi^{ex\,ante}(a, S)$ be the set of available GDP slots that their time intervals are entirely owned by airline a before the GDP starts.[24] For $A' \subseteq A$, let $\Phi^{ex\,ante}(A', S) \equiv \cup_{a \in A'} \Phi^{ex\,ante}(a, S)$ and $\Phi(A', S) \equiv \cup_{a \in A'} \Phi(a, S)$. Let $S_{A'} \in \{\Phi^{ex\,ante}(A', S), \Phi(A', S)\}$ (if $A' = A$ or $A' = \{a\}$, we write S_A and S_a, respectively).

An airline has the right to swap its own flights within its own set of slots.[25] A landing schedule for a $\Pi_a^{S_a}$ is **self-optimized** (with respects to S_a) if $\forall f \in F_a$, $\Pi_a^{S_a}(f), \Pi'_a(f) \in S_a$, $\Pi_a^{S_a} \succsim_a \Pi'_a$. Since the preference of airline $a \in A$ is strict, $\Pi_a^{S_a}$ is necessarily unique. To construct $\Pi_a^{S_a}$, order slots in S_a in ascending order, assign a's most important flight to the earliest slot that it can feasibly use, then assign a's second most important flight to the earliest slot (among those remaining) that it can feasibly use, and so on until there is no more slot or no more flight.

A landing schedule Π is **individually rational** (with respect to S_A) if $\forall a \in A$, $\Pi_a \succsim_a \Pi_a^{S_a}$. A schedule mechanism φ is *individually rational* if for any strategy profile $(\boldsymbol{R}, \boldsymbol{e})$, $\varphi(\boldsymbol{R}, \boldsymbol{e})$ is individually rational. A landing schedule Π is in the core if no subgroup of airlines could reallocate their slots to each other and make themselves better off than in Π. Formally, a landing schedule Π is in the **core** if $\nexists \Pi'$ and $A' \subseteq A$ such that (i) $\forall f \in \cup_{a \in A'} F_a$, $\Pi'(f) \in S_{A'}$, and (ii) $\forall a \in A'$, $\Pi' \succ_a \Pi$.[26] A schedule mechanism φ is **core-selecting** if for any strategy profile $(\boldsymbol{R}, \boldsymbol{e})$, $\varphi(\boldsymbol{R}, \boldsymbol{e})$ is in the core. If S_A is empty, then any feasible mechanism is individually rational and core-selecting.[27]

Schummer and Vohra (2013) consider that Compression does not respect property rights because it might produce outcomes outside the core. We propose the following explicit definition for mechanism instead.[28] A schedule mechanism or lottery mechanism **respects**

[24] Formally, $\Phi^{ex\,ante}(a, S) = \{s_n \in S | [1 + (n-1)l, 1 + nl] \subseteq \cup_{s_n^o \in \Phi(a, S^o)} [n, n+1]\}$. See footnote 20 for the definition of s_n.

In the case that there are runway closures caused by aircraft incidents and only one runway left, the construction of $\Phi^{ex\,ante}(a, S)$ should follow the description in Section 6.2.2 rather than the description here.

[25] See Section 17-9-5 of the *Facility Operation and Administration*. The link to download this document is in footnote 4.

[26] The core defined by weak domination might be empty. For example, when an airline with a GDP slot cancels all of its flights and such GDP slot is demanded by more than 1 airline, the core defined by weak domination is empty.

[27] Note that this is the counterpart of any acceptable mechanism is individually rational and core-selecting in a house allocation problem.

[28] When there are indifferences in preferences, a mechanism that respects property rights might produce outcomes outside the core. More details can be found in Section 6.2.

property rights over S_a if for each slot $s \in S_a$, a can use the slot by itself or trade it for a better slot if there is any. A slot s' is *better* than s for a if (i) s' can be used by a flight $f \in F_a$ that has not been assigned a slot or is currently assigned a slot later than s', (ii) s cannot be used by a flight $f' \in F_a$ that is more important than f, or s can be used by f' but f' is currently assigned a slot earlier than s. We say a mechanism respects property rights *before* a GDP starts if it respects property rights over $\Phi^{ex\,ante}(a, S)$, and we say a mechanism respects property rights *after* a GDP starts if it respects property rights over $\Phi(a, S)$.[29] So a mechanism respects property rights before and after a GDP starts if it respects property rights over S_a.

A lottery mechanism is *ex post individually rational* if it only gives positive probability to landing schedules that are individually rational. A lottery mechanism is *ex post Pareto efficient* if it only gives positive probability to landing schedules that are Pareto efficient. Other *ex post properties* for a lottery mechanism are defined analogously.

4 The Mechanism

Given a set $\mathcal{X}$, a **(priority) ordering** (of its elements) is a bijective function $z(\mathcal{X}) : \{1, 2, ..., |\mathcal{X}|\} \to \mathcal{X}$. Let $Z(\mathcal{X})$ be the set of orderings ($|Z(\mathcal{X})| = |\mathcal{X}|!$).

We define the **Multiple Trading Cycles** mechanism to be a mechanism that produces a landing schedule for each input using the following algorithm.

Algorithm 1:

Pre-competition Stage (Identification and Allocation of Non-scarce Resources):

(a) Order flights in F in increasing order of e_f (break ties arbitrarily).

(b) Assign flights sequentially to the earliest slot in S that each flight can feasibly use (there might be gaps between occupied slots). Denote the tentative landing schedule by $\hat{\Pi}$ and the set of occupied slots by S^{0-0}.

(c) Let $F^{0-0} = F$. Find the earliest $s \in S^{0-0}$ such that

(c-i) $s = \hat{\Pi}^{-1}(f)$ for some $f \in F_a^{0-0}$ and $e_f > s(-1)$, where $s(-1)$ is the last slot before s in S^{0-0} (f occupies s in $\hat{\Pi}$ and will not compete for slots earlier than s. Since in $\hat{\Pi}$, all flights that arrive strictly earlier than f will get a slot strictly earlier than s, this condition also implies that $\forall f' \in F^{0-0}$ with $e_{f'} \leq s(-1)$, $\hat{\Pi}^{-1}(f') \leq s(-1)$. Therefore, there are sufficient slots to accommodate flights that arrive earlier than f, so they will not compete for s);

(c-ii) If f' has $e_f \leq e_{f'} \leq s$, then $f' \in F_a^{0-0}$ (all flights that want s belong to a).

(We say s is *demanded* only by airline a (not can be used only by a) if (c-i) and (c-ii) hold

[29]TC in Schummer and Vohra (2013) and DASO in Schummer and Abizada (2017) indeed respect property rights after a GDP starts.

simultaneously (note that (c-i) is trivially satisfied if s is the earliest occupied slot) and s is demanded by more than 1 airline if either (c-i) or (c-ii) fails (such a slot might be assigned to different airlines in different feasible and non-wasteful mechanisms). For each slot in S^{0-t} (see below, $t = 0, 1, ...$), it is a *non-scarce* resource if it is demanded only by 1 airline, and it is a *scarce* resource if it is demanded by more than 1 airline. We can further categorize the slots in S^{0-t} into four groups: type 1 slots satisfy (c-i) but not (c-ii), type 2 slots do not satisfy both, type 3 slots satisfy (c-ii) but not (c-i), and type 4 slots satisfy both. Only type 4 slots are non-scarce resources.

Example 1:

GDP slot	s_1	s_2	s_3	s_4
$\hat{\Pi}$	$f_{a,1}$	$f_{b,1}$	$f_{c,1}$	$f_{d,1}$
e_f	1	1	2	4

The number below each flight is its earliest feasible arrival time (we use n to mean s_n for e_f as mentioned, so $e_{f_{a,1}} = 1$ means $e_{f_{a,1}} = s_1$). In this example, $S^{0-0} = \{s_1, s_2, s_3, s_4\}$ and $\hat{\Pi}$ is shown in the table above. s_1 is a type 1 slot since (c-i) is trivially satisfied as s_1 is the earliest occupied slot but (c-ii) fails as $f_{b,1}$ also wants s_1. s_2 is a type 2 slot since (c-i) fails as $f_{b,1}$ wants s_1 and (c-ii) fails as $e_{f_{a,1}} \leq e_{f_{b,1}} \leq 2$ but $f_{b,1} \notin F_a^{0-t}$. s_3 is a type 3 slot since (c-i) fails as $f_{c,1}$ wants s_2 but (c-ii) is trivially satisfied (there is no such f'). s_4 is a type 4 slot as (c-i) and (c-ii) hold simultaneously. It is easy to see that $f_{d,1}$ will not compete for slots earlier than s_4 and there are sufficient slots to accommodate flights that arrive earlier than $f_{d,1}$, so those flights will not compete for s_4. Also, there is no other flight arrives at the same time or later. Therefore, any feasible and non-wasteful mechanism will assign s_4 to $f_{d,1}$. The pre-competition stage identifies and allocates type 4 slots to avoid strategic issues that we will discuss later.)

Assign s to $f_{a,i}$, where $f_{a,i} \in F_a^{0-0}$ is the most important flight with $e_f \leq e_{f_{a,i}} \leq s$. Remove $f_{a,i}$ from F^{0-0} and s from S^{0-0}. If $f \neq f_{a,i}$, modify $\hat{\Pi}$ in the following way: Start from f, move each flight to the next slot in S^{0-0} until $\hat{\Pi}(f_{a,i})$ is filled. For $t = 0, 1, ...$, update F^{0-t} to $F^{0-(t+1)}$ and S^{0-t} to $S^{0-(t+1)}$, respectively (a type 3 slot in S^{0-t} can become a type 4 slot in $S^{0-(t+1)}$).

(d) Repeat (c) until all slots are demanded by more than 1 airline.

Denote the resulting sets S^{main} and F^{main}.

Main Stage (Allocation of Scarce Resources):

If this is the first assignment in a GDP, for each $a \in A$, construct $S_a = \Phi^{ex\,ante}(a, S)$ according to the initial slot ownership function $\Phi(A, S^o)$ (see footnote 24). Otherwise, let $S_a = \Phi(a, S)$ (from the last assignment). Creates $|F_a^o|$ surrogates of a for each $a \in A$, name them $a(1), a(2), ..., a(|F_a^o|)$. Denote the set of surrogates by $\mathcal{A}$ ($|\mathcal{A}| = |F^o|$). Randomly select an ordering $z(\mathcal{A})$ with uniform distribution over $Z(\mathcal{A})$. For each a, rearrange its surrogates to their positions in the ordering such that they are in the order of $a(1), a(2), ..., a(|F_a^o|)$ (alternatively, one can draw a surrogate at a time (without replacement) and let the first surrogate of $a \in A$ be $a(1)$, the second be $a(2)$, and so on). Denote the resulting ordering z.

Let $a(i) \in \{a(1), ...a(|F_a^{main}|)\}$ represent the i-th important flight in F_a^{main} (according to R_a); we call it a remaining flight of a. Let $a(i) \in a(|F_a^{main}|+1), ..., a(|F_a|)$ represent the i-th important flight in $F_a \setminus F_a^{main}$ (according to R_a); we call it a duplicate flight of a (each of these flights will be *assigned pro forma* the slot it was assigned in the pre-competition stage. They are here to reward airlines that gave them slots and so property rights over those slots are respected). Let $a(i) \in a(|F_a|+1), ..., a(|F_a^o|)$ represent a canceled flight of a; we call it a dummy flight of a (to be consistent with RBS, airline with canceled flights will get the same number of of slots).

Let $S^1 = S^{0-0}$ and $F^1 = F$.

Step 1 - Without lost of generality, let $a(1)$ be the first flight in z.

(i) If $a(1)$ is a dummy flight, remove it and skip to the next flight in line.

If $a(1)$ is a remaining flight, let $a(1)$ pick the earliest feasible slot in $S^1 \cap S^{main}$. If $a(1)$ is a duplicate flight, let $a(1)$ pick the earliest feasible slot in $S^1 \setminus S^{main}$.

(ii) If $a(1)$ picks a slot in (a) $S^1 \cap S_a$, (b) $S^1 \setminus S_A$ or $S^1 \cap S_b$ but b has no remaining/duplicate flight in F^1, assign this slot to $a(1)$, go to the next step.

(iii) If $a(1)$ picks a slot $s \in S^1 \cap S_b$ and b has a remaining/duplicate flight in F^1, modify z by inserting $b(1)$ in front of $a(1)$.

(iii-i) If $b(1)$ picks a slot in $S^1 \setminus S_A$ or a slot in $S^1 \cap S_c$ but c has no remaining/duplicate flight in F^1, assign this slot to $b(1)$ and assign s to $a(1)$, go to the next step.

Let $F^{1-0} = F^1$ and $S^{1-0} = S^1$.

(iii-ii) If $b(1)$ picks s, assign s to $b(1)$. Let $S^{1-1} = S^{1-0} \setminus \{s\}$ and $F^{1-1} = F^{1-0} \setminus \{b(1)\}$. Let $a(1)$ pick the next available slot in S^{1-1}.

When (iii-ii), (iii-iii) or (iii-iv)-(a) are repeated, for $t = 1, ...$, update S^{1-t} to $S^{1-(t+1)}$ and F^{1-t} to $F^{1-(t+1)}$, respectively.

(iii-iii) If $b(1)$ picks a slot s' in $S^1 \cap S_b \setminus \{s\}$, assign this slot to $b(1)$, modify z by inserting $b(2)$ behind $b(1)$. If there is no $b(2)$ or $b(2)$ is a dummy flight, then it will be in case (ii)-(b); otherwise, apply (iii) to $b(2)$ with $S^{1-1} = S^{1-0} \setminus \{s'\}$ in place of S^1 and $F^{1-1} = F^{1-0} \setminus \{b(1)\}$ in place of F^1.

(iii-iv) If $b(1)$ picks a slot $s'' \in S^1 \cap S_c$ and c has a remaining/duplicate flight in F^1, modify z by inserting $c(1)$ in front of $b(1)$, apply (iii)$'$ to $c(1)$ with $b(1)$ in place of $a(1)$, where (iii)$'$ is a generalization of (iii): For (iii-i), replace "assign this slot to..." by "then it will be in case (iii-iv)-(b).” For (iii-iii), replace “then it will be in case (ii)-(b); otherwise, apply (iii)” by “then it will be in case (iii-iv)-(b); otherwise, apply (iii)′.”

For each airline $a \in A$, let s_a be some slot in $S^{1-t} \cap S_a$ for some $t \in \mathbb{N}$. If there is (a) a cycle $(x(k), s_y, y(\cdot), ..., s_z, z(\cdot), s_x)$ of slots and most important remaining/duplicate flights such that $x(\cdot)$ picks s_y,..., $z(\cdot)$ picks s_x ($a(1)$ is not in the cycle), remove all flights in the cycle by assigning them the slots they pick. Let $F^{1-(t+1)} = F^{1-t} \setminus \{x(k), y(\cdot), ..., z(\cdot)\}$ and $S^{1-(t+1)} = S^{1-t} \setminus \{s_y, ..., s_z, s_x\}$.

If s_x is demanded by the flight that inserted $x(k)$, let it pick the next available slot (if $x(k)$ is inserted by $x(k-1)$, then check if s_x is demanded by the flight that inserted $x(k-1)$, and so on. Denote the flight of x that was inserted by another airline by $x(\cdot)$); otherwise, modify z by inserting $x(k+1)$ behind $x(k)$. If there is no $x(k+1)$ or $x(k+1)$ is a dummy flight, then it will be in case (iii-iv)-(b). Otherwise, apply (iii)′ to $x(k+1)$ with $S^{1-(t+1)}$ in place of S^1, $F^{1-(t+1)}$ in place of F^1, and the flight that inserted $x(\cdot)$ in place of $a(1)$.

After possible repetitions of (iii-ii), (iii-iii) and (iii-iv)-(a), at the end, there must be (b) a chain $(a(1), s_d, d(\cdot), ..., s_g, g(\cdot), s''')$ with s''' in $S^{1-T} \setminus S_A$ or $S^{1-T} \cap S_w$ but w has no remaining/duplicate flight in F^{1-T}, or (c) a cycle $(a(1), s_d, d(\cdot), ..., s_g, g(\cdot), s_a)$. Remove all flights in the chain/cycle by assigning them the slots they pick. Go to the next step. (In both cases, d does not have to be b, and even $d = b$, s_d does not have to be s. $a(1)$ would pick the next available slot if b uses s or there is a cycle that contains s but not $a(1)$, in which b trades s for a slot owned by another airline.)

Denote the resulting sets S^2 and F^2.

Step $n \geq 2$ - Without lost of generality, let $a(i)$ be the next flight in line.

(i) If $a(i)$ is a dummy flight, remove it and skip to the next flight in line.

If $a(i)$ is a remaining flight, let $a(i)$ pick the earliest feasible slot in $S^n \cap S^{main}$. If $a(i)$ is a duplicate flight, let $a(i)$ pick the earliest feasible slot in $S^n \setminus S^{main}$.

(ii) If $a(i)$ picks a slot in (a) $S^n \cap S_a$, (b) $S^n \setminus S_A$ or $S^n \cap S_b$ but b has no remaining/duplicate flight in F^n, assign this slot to $a(i)$, go to the next step.

(iii) If $a(i)$ picks a slot $s \in S^n \cap S_b$ and b has a remaining/duplicate flight in F^n, modify z by inserting $b(j)$ in front of $a(i)$, where $b(j)$ is b's most important remaining/duplicate flight in F^n.

(iii-i) If $b(j)$ picks a slot in $S^n \setminus S_A$ or a slot in $S^n \cap S_c$ but c has no remaining/duplicate flight in F^n, assign this slot to $b(j)$ and assign s to $a(i)$, go to the next step.

Let $F^{n-0} = F^n$ and $S^{n-0} = S^n$.

(iii-ii) If $b(j)$ picks s, assign s to $b(j)$. Let $S^{n-1} = S^{n-0} \setminus \{s\}$ and $F^{n-1} = F^{n-0} \setminus \{b(j)\}$. Let $a(i)$ pick the next available slot in S^{n-1}.

When (iii-ii), (iii-iii) or (iii-iv)-(a) are repeated, for $t = 1, ...$, update S^{n-t} to $S^{n-(t+1)}$ and

F^{n-t} to $F^{n-(t+1)}$, respectively.

(iii-iii) If $b(j)$ picks a slot s' in $S^n \cap S_b \setminus \{s\}$, assign this slot to $b(j)$, modify z by inserting $b(j+1)$ behind $b(j)$. If there is no $b(j+1)$ or $b(j+1)$ is a dummy flight, then it will be in case (ii)-(b); otherwise, apply (iii) to $b(j+1)$ with $S^{n-1} = S^{n-0} \setminus \{s'\}$ in place of S^n and $F^{n-1} = F^{n-0} \setminus \{b(j)\}$ in place of F^n.

(iii-iv) If $b(j)$ picks a slot $s'' \in S^n \cap S_c$ and c has a remaining/duplicate flight in F^n, modify z by inserting $c(\cdot)$ in front of $b(j)$, where $c(\cdot)$ is c's most important remaining/duplicate flight in F^n. Apply (iii)$'$ to $c(\cdot)$ with $b(j)$ in place of $a(i)$ (see the description for (iii)$'$ in Step 1).

For each airline $a \in A$, let s_a be some slot in $S^{n-t} \cap S_a$ for some $t \in \mathbb{N}$. If there is (a) a cycle $(x(k), s_y, y(\cdot), ..., s_z, z(\cdot), s_x)$, remove all flights in the cycle by assigning them the slots they pick. Let $F^{n-(t+1)} = F^{n-t} \setminus \{x(k), y(\cdot), ..., z(\cdot)\}$ and $S^{n-(t+1)} = S^{n-t} \setminus \{s_y, ..., s_z, s_x\}$.

If s_x is demanded by the flight that inserted $x(k)$, let it pick the next available slot (if $x(k)$ is inserted by $x(k-1)$, then check if s_x is demanded by the flight that inserted $x(k-1)$, and so on. Denote the flight of x that was inserted by another airline by $x(\cdot)$); otherwise, modify z by inserting $x(k+1)$ behind $x(k)$. If there is no $x(k+1)$ or $x(k+1)$ is a dummy flight, then it will be in case (iii-iv)-(b). Otherwise, apply (iii)$'$ to $x(k+1)$ with $S^{n-(t+1)}$ in place of S^n and $F^{n-(t+1)}$ in place of F^n, and the flight that inserted $x(\cdot)$ in place of $a(i)$.

After possible repetitions of (iii-ii), (iii-iii) and (iii-iv)-(a), at the end, there must be (b) a chain $(a(i), s_d, d(\cdot), ..., s_g, g(\cdot), s''')$ with s''' in $S^{n-T} \setminus S_A$ or $S^{n-T} \cap S_w$ but w has no remaining/duplicate flight in F^{n-T}, or (c) a cycle $(a(i), s_d, d(\cdot), ..., s_g, g(\cdot), s_a)$. Remove all flights in the chain/cycle by assigning them the slots they pick. Go to the next step.

Denote the resulting sets S^{n+1} and F^{n+1}.

The main stage stops when $F^k = \emptyset$ for some $k \geq 1$.

Supplemental Stage:

Let $V = S \setminus S^{0-0}$ be the set of remaining vacant slots. Start from the earliest slot in $V \cap S_A$, if a slot is in some S_a and a has a dummy flight, assign it to a and remove a dummy flight of a. Repeat until there is no more slot can be assigned by the above procedure. Denote the resulting set V^1. Assign the earliest slot in V^1 to the dummy flight with the highest order in z. Repeat until there is no more dummy flight.

4.1 An Alternative Algorithm

Abdulkadiroğlu and Sönmez (1999) provide 2 algorithms to find the outcome of the YGMH-IGYT mechanism. One is YGMH-IGYT algorithm, and another is top trading cycles algorithm. Algorithm 1 generalizes the YGMH-IGYT algorithm, and Algorithm 2 below generalizes the top trading cycles algorithm. Note that (iii-i) to (iii-iv) in Algorithm 1, which provide a complete picture of all cycles that are triggered by $a(i)$, are not explicitly described in the YGMH-IGYT algorithm. Comparing these two algorithms, Algorithm 1 is more transparent on how cycles form, while Algorithm 2 is more transparent on how it works.

We change the main stage from "Step 1...."

Algorithm 2:

In general, at Step h:

- Each remaining flight in F^h points to the earliest feasible slot in $S^h \cap S^{main}$;
- Each duplicate flight in F^h points to the earliest feasible slot in $S^h \setminus S^{main}$;
- Each slot in $S^h \setminus S_A$ or $S^h \cap S_a$ ($\forall a \in A$) but a has no remaining/duplicate flight in F^h points to the remaining/duplicate flight in F^h with the highest priority in z (if such flight is a dummy flight, remove it and skip to the next flight in line); and
- Each slot in $S^h \cap S_a$ ($\forall a \in A$) and a has a remaining/duplicate flight in F^h points to the most important flight in $F^h \cap F_a$.

Since $|F|$ and $|S^{0-0}|$ are finite, there is at least one cycle. Each airline can be in at most 1 cycle in each step. Every flight in a cycle is assigned (or assigned pro forma) the slot that it points to and removed with such slot. Whenever there is a slot in $S^h \setminus S_A$ or $S^h \cap S_a$ but a has no remaining/duplicate flight in F^h in a cycle, the remaining/duplicate flight in F^h with the highest priority in z is also in the cycle. The set of slots that are not removed at the end of Step h is denoted by S^{h+1}. The set of flights that are not removed at the end of Step h is denoted by F^{h+1}. The main stage stops when $F^h = \emptyset$ for some $h \geq 1$.

Theorem 1: For a given ordering z, Algorithm 1 and Algorithm 2 produce the same outcome.

Note that in Algorithm 2, a cycle that is not removed at any step remains a cycle at the next step (as the earliest feasible slots for the flights in a cycle still remain). Therefore, removing one cycle (instead of multiple) at a time will not change its outcome. Algorithm 1 removes cycles in Algorithm 2 one at a time but possibly multiple at a step. Recall (iii-ii) in Algorithm 1: If $b(j)$ picks $s \in S^n \cap S_b$ (the slot demanded by $a(i)$), assign s to $b(j)$, and let $a(i)$ pick the next available slot. In this case, s leaves earlier than $a(i)$ in Algorithm 2, so $a(i)$ cannot pick s. Recall (iii-iii) in Algorithm 1: If $b(j)$ picks a slot s' in $S^n \cap S_b \setminus \{s\}$, assign this slot to $b(j)$, modify z by inserting $b(j+1)$ behind $b(j)$. In this case, $b(j)$ leaves earlier than $a(i)$ in Algorithm 2, so $b(j+1)$ should be in place of $b(j)$. In general, for any given ordering z, any flights that are inserted in front of $a(i)$ but not in a cycle/chain that contains $a(i)$ in Algorithm 1 leave earlier than $a(i)$ in Algorithm 2, and any flights that are inserted in front of $a(i)$ and in a cycle/chain that contains $a(i)$ in Algorithm 1 are in the same cycle in Algorithm 2.

4.2 Some Observations

Both algorithms stop in at most $|F^o|$ steps. The following claim implies that S^{0-0} is assigned in any feasible and non-wasteful mechanism.

Claim 1: S^{0-0} is assigned in any feasible and non-wasteful landing schedule Π.

There is a pattern for scarce resources. Scarce resources are sequences of adjacent slots in S^{main} such that each **sequence** starts with a type 1 slot and ends with a type 3 slot. Each sequence contains one type 1 slot, some type 3 slots, and possibly some type 2 slots. In any feasible and non-wasteful landing schedule, a flight that gets a slot in a sequence in $\hat{\Pi}$ will *always* get a slot in the same sequence (it is infeasible to get a slot earlier than the sequence as (c-i) holds for the first slot in the sequence. It is wasteful if the flight gets a slot later than the sequence—by feasibility, the number of flights that can feasibly use the slots in the sequence is fixed, the flight gets a slot later than the sequence implies there exists some slot in S^{0-0} that is empty); a flight that gets a slot outside a sequence in $\hat{\Pi}$ will *never* get a slot in that sequence (it is infeasible to get a slot in a sequence earlier than the slot it gets in $\hat{\Pi}$ as (c-i) holds for the type 1 or type 4 slot that locates right after the sequence in S^{main}. It is wasteful if the flight gets a slot in a sequence later than the slot it gets in $\hat{\Pi}$—by feasibility, the number of flights that can feasibly use the slots earlier than this sequence is fixed, the flight gets a slot in this sequence implies there exists some slot earlier than this sequence in S^{0-0} is empty). A type 1 slot can be followed by a type 3 slot. Also, there might be more than one type 3 slot.

Example 2:

GDP slot	s_1	s_2	s_3
$\hat{\Pi}$	-	$f_{a,1}$	$f_{b,1}$
e_f	-	1	2

GDP slot	s_1	s_2	s_3	s_4
$\hat{\Pi}$	-	$f_{a,1}$	$f_{b,1}$	$f_{b,2}$
e_f	-	1	2	2

In the left table above, $s_1 \notin S^{0-t}$. It is easy to see that s_1 is a type 1 slot and s_3 is a type 3 slot in S^{0-t}. In the right table above, s_4 is also a type 3 slot.

Airline a's probability of getting the first position in z is $\frac{|F_a^o|}{|F^o|}$. Given it gets the first position, the probability of getting the second position declines to $\frac{|F_a^o|-1}{|F^o|-1}$. Airline a "pays" positions $a(|F_a^1|+1), ..., a(|F_a|)$ (possibly to other airlines) for the slots it obtains in the pre-competition stage. These positions have zero values to a (with respect to $\succ_a$), but they might be valuable to other airlines.

Recall that when (i) no airline owns a canceled flight, (ii) each airline owns exactly one non-canceled flight, and (iii) each airline owns at most one GDP slot, MTC reduces to YGMH-IGYT (with random ordering) as the pre-competition stage and the supplemental stage become redundant. However, one can modify YGMH-IGYT by using the pre-competition stage: Fix an ordering. If a house is acceptable to only one agent i and the house is his top choice, assign this house to i. If the house is owned by some agent j, let j take i's position if it is earlier than j's position (in the ordering) and i's house if i owns a house. Repeat this procedure until there is no more such house. Then run YGMH-IGYT with the reduced ordering (with these i's eliminated). The outcome does not change because j would be inserted in front of i when it is i's turn or i has been inserted as some agent demands i's house. Each agent that is assigned a house here always obtains the same house (in any acceptable and non-wasteful mechanism). In Section 6.1, we discuss another modification that is also meaningful to MTC but insignificant to YGMH-IGYT.

4.3 Subsequent Reassignments

In the current GDP, the first assignment is created by running RBS and possible Compression. But as airlines update their information, Compression might be run multiple times. MTC can also be used to perform reassignments.

A new instance is a tuple $\tilde{I} = (\tilde{S}, \tilde{A}, \tilde{F}^o, \tilde{R}, \tilde{e}, \Phi(\tilde{A}, \tilde{S}))$, where $\Phi(\tilde{A}, \tilde{S})$ is the slot ownership function from the last assignment restricted to $\tilde{A}$ and $\tilde{S}$.[30] If an airline freezes a flight in a slot $s \in S_a$ ("airlines will also have the capability to freeze flights they don't want moved up through the submission of an earliest time of arrival" (Wambsganss, 1996)), then $s \notin \tilde{S}$. If $\tilde{F}_a = \emptyset$ for some airline a (all flights in F_a^o are canceled or frozen in some slots), then $a \notin \tilde{A}$ and $F_a^o \nsubseteq \tilde{F}^o$. For $a \in \tilde{A}$, now its preference over feasible landing schedules, $\tilde{\succ}_a$, is induced by $\tilde{R}_a$ and $\tilde{e}_a$. Vacant slots obtained from the last supplemental stage might become valuable (in the sense that airlines can use them or trade them for better slots) in this new instance.[31]

[30] In this paper, we assume l is constant across instances. But if l changes, to construct $\Phi(\tilde{A}, \tilde{S})$, one can either modify the formula in footnote 24 or ignore the change in l.

[31] The supplemental stage of MTC assigns the earliest slots in $V \cap S_a$ to a if a has dummy flights but not some random slots in V. Suppose when a has a dummy flight, we assign the earliest slot in $V \cap S_a$, s, to $b \in A$ and assign a random slot s' to a. Then if a wants s but not s' in the next instance, it has to trade with b (this potentially hurts some less important flights of a). Here, s cannot be compared with s' in the former instance but s is better than s' in the latter instance. Since a's preference might change, it seems that assigning s to a in the former instance is more appropriate.

4.4 Examples

Since MTC is a rather complex mechanism, a giant example that shows all of its features would be quite involved. Thus, we use two examples to demonstrate MTC.

Example 3:

Initial slot	s^o_1	s^o_2	s^o_3	s^o_4	s^o_5	s^o_6	s^o_7	s^o_8	s^o_9	s^o_{10}	s^o_{11}	s^o_{12}	s^o_{13}	s^o_{14}
Flight	$f_{c,3}$	$f_{b,(1)}$	$f_{a,1}$	$f_{a,2}$	$f_{a,3}$	$f_{b,2}$	$f_{c,2}$	$f_{c,1}$	$f_{b,1}$	$f_{a,4}$	$f_{a,5}$	$f_{a,6}$	$f_{b,3}$	$f_{a,7}$
$\Phi(\cdot, S^o)$	c	b	a	a	a	b	c	c	b	a	a	a	b	a
e_f	1	-	2	3	4	2	5	6	8	7	10	10	12	10
R	3	-	1	2	3	2	2	1	1	4	5	6	3	7
GDP slot ($l = 2$)	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}
$\hat{\Pi}$	$f_{c,3}$	$f_{a,1}$	$f_{b,2}$	$f_{a,2}$	$f_{a,3}$	$f_{c,2}$	$f_{c,1}$	$f_{a,4}$	$f_{b,1}$	$f_{a,6}$	$f_{a,5}$	$f_{a,7}$	$f_{b,3}$	
$\Phi^{ex\,ante}(\cdot, S)$		a		c		a								
RBS+Comp	$f_{c,3}$	$f_{b,2}$	$f_{a,1}$	$f_{a,2}$	$f_{a,3}$	$f_{c,2}$	$f_{c,1}$	$f_{b,1}$	$f_{a,4}$	$f_{a,5}$	$f_{a,6}$	$f_{a,7}$	$f_{b,3}$	b
MTC	$f_{c,3}$	$f_{a,1}$	$f_{a,2}$	$f_{a,3}$	$f_{b,2}$	$f_{c,1}$	$f_{a,4}$	$f_{b,1}$	$f_{c,2}$	$f_{a,5}$	$f_{a,6}$	$f_{b,3}$	$f_{a,7}$	b

Flight $f_{b,(1)}$ is a canceled flight, where the subscript $b,(1)$ indicates it is a canceled flight of b with index (1) (we use this notation occasionally and might simply use "-" to indicate a flight is canceled). R is the importance ranking profile. In this example, $s_2 \in \Phi^{ex\,ante}(a, S)$ (s^o_3 and $s^o_4 \in \Phi(a, S^o)$ cover the entire time interval of s_2) but a's most important flight $f_{a,1}$ with $e_{f_{a,1}} = 2$ cannot use it because of RBS. This shows RBS does not respect property rights before a GDP starts. Since s_2 would be assigned to $f_{a,1}$ in any individually rational landing schedule, RBS might lead to outcome that is not individually rational and thus not in the core (any individually rational outcome dominates the RBS+Comp outcome via subgroup $\{a\}$). Also, the RBS+Comp outcome is not Pareto efficient: $f_{c,1} R_c f_{c,2}$ but $e_{f_{c,1}} \leq \Pi(f_{c,2}) < \Pi(f_{c,1})$.

Putting e with $\Phi^{RBS+Comp}(A, S)$ (the induced ownership function of RBS+Comp outcome) back into Compression will give the same outcome, and that implies Compression is not Pareto efficient, not individually rational and thus not core-selecting.[32] The violation of individual rationality is not because s_2 has not been assigned to $f_{a,1}$ (RBS+Compression produces an assignment with $s_2 \in \Phi^{RBS+Comp}(b, S)$) but because the landing schedule is not self-optimized yet (recall we define individual rationality through self-optimization).

Airline a's preference in this instance is the following:

$$\succsim_a\colon (s_2, s_3, s_4, s_7, s_{10}, s_{10}, s_{10}), (s_2, s_3, s_4, s_7, s_{10}, s_{10}, s_{11}), \ldots,$$
$$(s_2, s_3, s_4, s_7, s_{10}, s_{11}, s_{12}), \ldots, (s_3, s_3, s_4, s_7, s_{10}, s_{10}, s_{10}), \ldots,$$

[32]Example 5 below shows that Compression is not core-selecting but not because of violations of individual rationality.

where each element is of the form $(\Pi(f_{a,1}), \Pi(f_{a,2}), ..., \Pi(f_{a,7}))$ and the best feasible landing schedule for a is $(s_2, s_3, s_4, s_7, s_{10}, s_{11}, s_{12})$. Airline b and c's preferences can be expressed similarly. Now we run MTC.

Pre-competition stage: A tentative landing schedule $\hat{\Pi}$ is created as shown in the second table above and the set of occupied slot S^{0-0} is $\{s_{1,}s_2, ..., s_{13}\}$.

The first type 4 slot is s_1 as it is demanded only by c with $f_{c,3}$, so $f_{c,3}$ is assigned this slot. The resulting sets are S^{0-1} and F^{0-1}.

The second type 4 slot is s_{10} as it is demanded only by a with $f_{a,5}, f_{a,6}, f_{a,7}$, so the most important flight among these three, $f_{a,5}$, is assigned this slot. Since $\hat{\Pi}(f_{a,5}) = s_{11}$, s_{11} is empty now. Update $\hat{\Pi}$ by moving $f_{a,6}$ into s_{11}. The resulting sets are S^{0-2} and F^{0-2}.

The third type 4 slot is s_{11} as it is demanded only by a with $f_{a,6}, f_{a,7}$, so the most important flight among these two, $f_{a,6}$, is assigned this slot. The resulting sets are S^{0-3} and F^{0-3}. (Note that $f_{a,5} \notin F^{0-2}$ and $s_{10} \notin S^{0-2}$. Without the previous iteration, (c-i) does not hold for s_{11}. In S^{0-2}, $s(-1)$ of s_{11} is s_9.)

Now all slots are demanded by more than 1 airline (s_{12} is demanded by a with $f_{a,6}$ and b with $f_{b,3}$). $S^{main} = S^{0-3} = \{s_2, s_{3,}..., s_9, s_{12}, s_{13}\}$ and

$$F^{main} = F^{0-3} = \{f_{a,1}, f_{a,2}, f_{a,3}, f_{a,4}, f_{a,7}, f_{b,1}, f_{b,2}, f_{b,3}, f_{c,2}, f_{c,3}\}.$$

Main stage: For each a, $S_a = \Phi^{ex\,ante}(a, S)$ is constructed as shown in the second table above.

$$z = (a(1), a(2), a(3), b(1), c(1), a(4), b(2), b(3), b(4), a(5), a(6), a(7), c(2), c(3)).$$

$a(1), ..., a(5)$ represent $f_{a,1}, f_{a,2,}, f_{a,3}, f_{a,4}, f_{a,7}$ (remaining flights), respectively. $a(6)$ and $a(7)$ represent $f_{a,5}$ and $f_{a,6}$ (duplicate flights), respectively. $b(1), ..., b(3)$ represent $f_{b,1}, ..., f_{b,3}$, respectively. $b(4)$ represents $f_{b,(1)}$ (dummy flight). $c(1)$ and $c(2)$ represent $f_{c,1}$ and $f_{c,2}$, respectively. Lastly, $c(3)$ represents $f_{c,3}$. $S^1 = S^{0-0}$ and $F^1 = F$.

Step 1: $a(1)$ picks $s_2 \in S^1 \cap S_a$ for $f_{a,1}$. $f_{a,1}$ is assigned this slot. The resulting sets are S^2 and F^2. ((ii)-(a))

Step 2: $a(2)$ picks $s_3 \in S^2 \setminus S_A$ for $f_{a,2}$. $f_{a,2}$ is assigned this slot. The resulting sets are S^3 and F^3. ((ii)-(b))

Step 3: $a(3)$ picks $s_4 \in S^3 \cap S_c$ for $f_{a,3}$ and c has a remaining/duplicate fight in F^3, modify z by inserting $c(1)$, c's most important remaining/duplicate fight in F^3, in front of $a(3)$. $c(1)$ picks $s_6 \in S^3 \cap S_a$ for $f_{c,1}$. $a(3)$ and $c(1)$ form a cycle. $f_{c,1}$ is assigned s_6 and $f_{a,3}$ is assigned s_4. The resulting sets are S^4 and F^4. ((iii-vi)-(c))

Step 4: $b(1)$ picks $s_8 \in S^4 \setminus S_A$ for $f_{b,1}$. $f_{b,1}$ is assigned this slot. The resulting sets are

S^5 and F^5.

Step 5: Note that $c(1)$ has been inserted in front. Now $a(4)$ picks $s_7 \in S^5 \setminus S_A$. $f_{a,4}$ is assigned this slot. The resulting sets are S^6 and F^6.

Step 6: $b(2)$ picks $s_5 \in S^6 \setminus S_A$ for $f_{b,2}$ (s_2, s_3 and s_4 have already been assigned). $f_{b,2}$ is assigned this slot. The resulting sets are S^7 and F^7.

Step 7: $b(3)$ picks $s_{12} \in S^7 \setminus S_A$ for $f_{b,3}$. $f_{b,3}$ is assigned this slot. The resulting sets are S^8 and F^8.

Step 8: $b(4)$ is a dummy flight, remove it and skip to $a(5)$. ((i))

$a(5)$ picks $s_{13} \in S^8 \setminus S_A$ for $f_{a,7}$ ($a(5)$ is a remaining flight, so it picks the earliest feasible slot in $S^8 \cap S^{main}$. s_{10} and s_{11} are in $S^8 \setminus S^{main}$, and s_{12} has been assigned before). $f_{a,7}$ is assigned this slot. The resulting sets are S^9 and F^9.

Step 9: $a(6)$ picks $s_{10} \in S^9 \setminus S_A$ for $f_{a,5}$ ($a(6)$ is a duplicate flight, so it picks the earliest feasible slot in $S^9 \setminus S^{main}$. Note that s_{10} is the slot $f_{a,5}$ was assigned in the pre-competition stage). $f_{a,5}$ is assigned this slot pro forma. The resulting sets are S^{10} and F^{10}.

Step 10: $a(7)$ picks $s_{11} \in S^{10} \setminus S_A$ for $f_{a,6}$. $f_{a,6}$ is assigned this slot pro forma. The resulting sets are S^{11} and F^{11}.

Step 11: $c(2)$ picks $s_9 \in S^{11} \setminus S_A$ for $f_{c,2}$ (s_5, s_6, s_7 and s_8 have already been assigned). $f_{c,2}$ is assigned this slot. The resulting sets are S^{12} and F^{12}.

Step 12: $c(3)$ picks $s_1 \in S^{12} \setminus S_A$ for $f_{c,3}$. $f_{c,3}$ is assigned this slot pro forma. The resulting sets are S^{13} and F^{13}. Note that $F^{13} = \emptyset$. The main stage stops here.

Supplemental stage:

$V = S \setminus S^{0-0} = \{s_{14,} s_{15}, ...\}$. The is no slot in $V \cap S_a$ for some a, so $V^1 = V$. Assign the earliest slot s_{14} to b.

Example 4:

GDP slot	s_1	s_2	s_3	s_4	s_5	s_6
Flight	$f_{c,1}$	$f_{a,1}$	$f_{a,2}$	$f_{a,3}$	$f_{a,(1)}$	$f_{b,1}$
$\Phi(\cdot, S)$	c	a	a	a	a	b
$\tilde{e}_f$	5	2	4	1	-	2
R	1	1	2	3	-	1
$\hat{\Pi}$	$f_{a,3}$	$f_{b,1}$	$f_{a,1}$	$f_{a,2}$	$f_{c,1}$	
MTC	$f_{a,3}$	$f_{a,1}$	$f_{b,1}$	$f_{a,2}$	$f_{c,1}$	a

In this example, $f_{c,1}$ has been delayed and cannot use the slot it was assigned in the last assignment. The same for $f_{a,2}$.

Pre-competition stage:

A tentative landing schedule $\hat{\Pi}$ is created as shown in the table above, and the set of occupied slot S^{0-0} is $\{s_{1,} s_2, ..., s_5\}$.

The first type 4 slot is s_1 as it is demanded only by a with $f_{a,3}$, so $f_{a,3}$ is assigned this slot. The resulting sets are S^{0-1} and F^{0-1}.

The second type 4 slot is s_4 as it is demanded only by a with $f_{a,2}$, so $f_{a,2}$ is assigned this slot. The resulting sets are S^{0-2} and F^{0-2}.

The third type 4 slot is s_5 as it is demanded only by c with $f_{c,1}$, so $f_{c,1}$ is assigned this slot. The resulting sets are S^{0-3} and F^{0-3}.

Now all slots are demanded by more than 1 airline. $S^{main} = S^{0-3} = \{s_2, s_3\}$ and $F^{main} = F^{0-3} = \{f_{a,1}, f_{b,1}\}$.

Main stage:

For each a, let $S_a = \Phi(a, S)$ from the last assignment (in the table above).

$$z = (b(1), a(1), a(2), c(1), a(3), a(4)).$$

$a(1)$ represents $f_{a,1}$. $a(2)$ and $a(3)$ represent $f_{a,2}$ and $f_{a,3}$, respectively. $a(4)$ represents $f_{a,(1)}$. $b(1)$ represents $f_{b,1}$. $c(1)$ represents $f_{c,1}$. $S^1 = S^{0-0}$ and $F^1 = F$.

Step 1: $b(1)$ picks $s_2 \in S^1 \cap S_a$ for $f_{b,1}$ and a has a remaining/duplicate fight in F^1, modify z by inserting $a(1)$ in front of $b(1)$. $a(1)$ picks $s_2 \in S^1 \cap S_a$ for $f_{a,1}$. Assign s_2 to $f_{a,1}$. The resulting sets are S^{1-1} and F^{1-1}. ((iii)-(ii))

$b(1)$ picks the next available slot $s_3 \in S^{1-1} \cap S_a$ and a has a remaining/duplicate fight in F^{1-1}, modify z by inserting $a(2)$ in front of $b(1)$.

$a(2)$ picks $s_4 \in S^{1-1} \cap S_a$. Assign s_4 pro forma to $f_{a,2}$ and modify z by inserting $a(3)$ behind $a(2)$. The resulting sets are S^{1-2} and F^{1-2}. ((iii)-(iii))

$a(3)$ picks $s_1 \in S^{1-2} \cap S_c$ for $f_{a,3}$ and c has a remaining/duplicate fight in F^{1-2}, modify z by inserting $c(1)$ in front of $a(3)$.

$c(1)$ picks $s_5 \in S^{1-2} \cap S_a$ for $f_{c,1}$. $a(3)$ and $c(1)$ form a cycle. $f_{c,1}$ is assigned pro forma s_5 and $f_{a,3}$ is assigned pro forma s_1. The resulting sets are S^{1-3} and F^{1-3}. ((iii-vi)-(a))

s_5 is not demanded by $b(1)$ (s_5 is some s_x in Algorithm 1, and $b(1)$ is the flight that inserted $x(k-1)$, which is $a(2)$ here), so $a(4)$ should be inserted behind $a(3)$. But $a(4)$ is a dummy flight, which means a has no remaining/duplicate fight in F^{1-3}. So $f_{b,1}$ is assigned s_3. The resulting sets are S^{1-4} and F^{1-4}. ((iii-vi)-(b))

Note that $F^{1-4} = \emptyset$. The main stage stops here.

Supplemental stage:

$V = S \setminus S^{0-0} = \{s_{6,\ldots}\}$. $s_6 \in V \cap S_b$ but b has no dummy flight. So $V^1 = V$. Assign the earliest slot s_6 to a.

5 Properties of the Mechanism

Proposition 1: The multiple trading cycles mechanism ϕ is regular, ex post feasible, ex post non-wasteful, and respects property rights over S_a.

Example 5:

GDP slot	s_1	s_2	s_3	s_4	s_5
Flight	-	-	$f_{a,1}$	$f_{c,1}$	$f_{b,1}$
$\Phi(\cdot, S)$	a	b	a	c	b
e_f	-	-	2	1	1
Compression	$f_{c,1}$	$f_{b,1}$	$f_{a,1}$	b	a
MTC	$f_{b,1}$	$f_{a,1}$	$f_{c,1}$	a	b

We have shown that RBS does not respect property rights before a GDP starts by Example 3. Example 5 shows that Compression does not respect property rights after a GDP starts and is not core-selecting but not because of violations of individual rationality. For a, s_2 is better than s_1 and s_3, while s_4 is not better than s_1 or s_3. Yet in the first step of Compression, c obtains s_1 for $f_{c,1}$ while a obtains s_4.[33] By contrast, in MTC, a trades s_1 for s_2. The Compression outcome is not in the core because it is dominated by the MTC outcome via subgroup $\{a, b\}$.

Proposition 2: The multiple trading cycles mechanism ϕ is ex post individually rational (with respect to S_A).

If an airline a only uses its own slots in S_a under any ordering z, $\phi_a(\boldsymbol{R}, \boldsymbol{e}) = \Pi_a^{S_a}$ with probability 1. But if it uses some other slots under some ordering z, $\phi_a(\boldsymbol{R}, \boldsymbol{e})$ would be preferred to $\Pi_a^{S_a}$.

Proposition 3: The multiple trading cycles mechanism ϕ is ex post Pareto efficient.

Suppose $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$ is a realized landing schedule of MTC. $\Phi^{\varphi^z(\boldsymbol{R},\boldsymbol{e})}(A, S)$ is its induced ownership function. Proposition 2 and 3 imply that putting $(\boldsymbol{R}, \boldsymbol{e})$ with $\Phi^{\varphi^z(\boldsymbol{R},\boldsymbol{e})}(A, S)$ back into MTC would not change the outcome because there will be no trading cycle for any ordering z'. If $a(i)$ picks the slot of $b(j)$, then $b(k)$ will be inserted in front of $a(i)$ where $b(k)$ is the most important remaining flight of b ($b(k)$ and $b(j)$ will not be duplicate flights as $a(i)$ demands the slot of $b(j)$). Procedure continues in a similar way until some flight f picks its top choice. Then the flight that inserted f picks its own slot. The same for the next flight that inserted it, and so on. At the end, $a(i)$ picks the next slot and the same argument applies until its gets the slot it was assigned in $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$.

In a housing market with strict preferences, the core is a subset of the Pareto set.[34] But

[33] Then b obtains s_4 for $f_{b,1}$ while a obtains s_5. Finally, $f_{b,1}$ is moved to s_2.

[34] In that problem, the core is equivalent to the core defined by weak domination, and the core defined by weak domination is a subset of the Pareto set.

in an airport slot allocation problem with strict preferences, the core might not be a subset of the Pareto set since $S_A \neq S$.[35] A landing schedule in the core is not necessary Pareto efficient if some airline can benefit by having a slot that is not owned by some airline. The following example shows a Pareto efficient landing schedule might not be in the core.

Example 6:

Flight	$f_{a,1}$	$f_{a,2}$	$f_{b,1}$
e_f	1	1	1
R	1	2	1

GDP slot	s_1	s_2	s_3
$\Phi(\cdot, S)$	a		a
Π	$f_{b,1}$	$f_{a,1}$	$f_{a,2}$
Π'	$f_{a,1}$	$f_{b,1}$	$f_{a,2}$

Π is Pareto efficient, but it is not in the core since a can use slots only in $\Phi(a, S)$ and be better off (as in Π'). Proposition 3 and Theorem 2 below imply that MTC selects landing schedules from the intersection of the core and the Pareto Set.

Theorem 2: The multiple trading cycles mechanism ϕ is ex post core-selecting.

Theorem 3: The multiple trading cycles mechanism ϕ is strategy-proof.

There are two sources of strategy-proofness. The first one is the the randomness introduced in the main stage together with a feature of lexicographic preference that it does not sacrifice the benefit of a flight for the benefit of a less important flight. By truth-telling, MTC minimizes the expected delays for each airline lexicographically. But if an airline deviates, it might be able to reduce the expected delays for some of its flights, but the expected delay for a more important flight will increase.[36] The next example will show the importance of randomness.

Recall that when (ii) each airline owns exactly one non-canceled flight and (iv) no airline owns a GDP slot, MTC reduces to random serial dictatorship; if we fix an ordering z, it further reduces to serial dictatorship. However, when (iv) holds but not (ii), MTC with fixed ordering is different from serial dictatorship. In this context, airlines are agents, so serial dictatorship would allow airline a to pick all slots it wants, then allow airline b to pick all slots it wants (among those remaining), etc. It is well-known that serial dictatorship is strategy-proof, and the reason is that an airline does not need to manipulate its report to get the best set of available slots, but MTC with fixed ordering does not have this feature. We illustrate this point by the following example.

Example 7: (MTC with fixed ordering is not strategy-proof)

---

[35] Indeed, this is also true for a house allocation with existing tenants problem with strict preferences. A similar but distinct result can be found in Roth and Sotomayor (1992): In a two-sided college admissions problem, the college-optimal stable matching does not need to be Pareto efficient for the colleges (Theorem 5.10), but this matching is in the core defined by weak domination (Theorem 5.36).

[36] If the lexicographic preference assumption is relaxed, the same result might still be obtained if the size of market goes to infinite.

Consider a case where (iv) holds but not (ii). There is no non-scarce resource in this example, so we can skip the pre-competition stage. Fix an ordering z, where $z = (a(1), b(1), a(2), b(2))$. $a(1)$ represents the most important flight according to R_a and $a(2)$ represents the other. $b(1)$ represents the most important flight according to R_b and $b(2)$ represents the other.

Flight	$f_{a,1}$	$f_{a,2}$	$f_{b,1}$	$f_{b,2}$
e_f	1	2	2	1
R	1	2	1	2
$\widehat{e_f}$	2	1	2	1
$\widehat{R}$	2	1	1	2

GDP slot	s_1	s_2	s_3	s_4
$\varphi^z(\boldsymbol{R}, \boldsymbol{e})$	$f_{a,1}$	$f_{b,1}$	$f_{a,2}$	$f_{b,2}$
$\varphi^z(\widehat{\boldsymbol{R}_a}, \boldsymbol{e}_a, (\boldsymbol{R}, \boldsymbol{e})_{-a})$	$f_{a,1}$	$f_{a,2}$	$f_{b,1}$	$f_{b,2}$
$\varphi^z(\boldsymbol{R}_a, \widehat{\boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a})$	$f_{a,2}$	$f_{a,1}$	$f_{b,1}$	$f_{b,2}$

In this example, by either misreporting its importance ranking or earliest feasible arrival times (as in the left table above), a can gain by having s_2 (see the right table above. In the bottom case, it can swap slots for $f_{a,1}$ and $f_{a,2}$).

The second source of strategy-proofness is the design of the pre-competition stage. If an airline knows one of the slots will be used by one of its flights only, say the most important one, then it will have the incentive to misreport its ranking such that this flight is the least important one; alternatively, it can misreport its earliest feasible arrival times such that each of its flights picks a slot for the next most important flight and the least important flight picks a slot for the most important flight. By doing either of these, all of its remaining flights would be weakly better off if one runs MTC without the pre-competition stage.

Example 8: (MTC without the pre-competition stage is not strategy-proof)

Flight	$f_{a,1}$	$f_{a,2}$	$f_{b,1}$
e_f	3	1	1
R	1	2	1
$\widehat{e_f}$	1	3	1
$\widehat{R}$	2	1	1

GDP slot	s_1	s_2	s_3
φ^{z^1} or $\widehat{\varphi^{z^1}}$ or $\widehat{\varphi^{z^2}}$	$f_{a,2}$	$f_{b,1}$	$f_{a,1}$
φ^{z^2} or φ^{z^3} or $\widehat{\varphi^{z^3}}$	$f_{b,1}$	$f_{a,2}$	$f_{a,1}$

We drop the arguments for $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$ and write φ^z. In this example, given a can swap slots for $f_{a,1}$ and $f_{a,2}$ whenever necessary, reporting either $\widehat{e_a}$ or $\widehat{\boldsymbol{R}_a}$ will give a the same outcome, but reporting both of them together will give a the outcome of $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$. We use $\widehat{\varphi^z}$ to represent $\varphi^z(\widehat{\boldsymbol{R}_a}, \boldsymbol{e}_a, (\boldsymbol{R}, \boldsymbol{e})_{-a})$ and $\varphi^z(\boldsymbol{R}_a, \widehat{\boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a})$ after necessary self-optimization. There are three possible orderings: $z^1 = (a(1), a(2), b(1))$, $z^2 = (a(1), b(1), a(2))$, and $z^3 = (b(1), a(1), a(2))$. Let $a(i)$ represent $f_{a,i}$ for $i \in \{1, 2\}$ and $b(1)$ represent $f_{b,1}$. By reporting either $\widehat{e_a}$ or $\widehat{\boldsymbol{R}_a}$, a can strictly gain if z^2 is realized and lose nothing under z^1 and z^3.

Note that s_3 would be assigned in the pre-competition stage. MTC avoids the aforementioned manipulations by assigning non-scarce resources (type 4 slots) in the pre-competition stage without asking airlines to give up anything meaningful to them in the main stage. For completeness, we provide an example in Appendix C to show Compression is not strategy-proof in our preference domain.

6 Extensions

6.1 Outside an Instance

An airline can freeze a flight $f \in F_a^o$ in a slot $s \in S_a$ and effectively remove s from an instance. This is not a strategy in the induced preference revelation game of a mechanism, but rather a way to change the game. Schummer and Abizada (2017) show that DASO is non-manipulable by postponing a flight cancellation (an airline cannot gain by freezing a canceled flight in a slot $s \in S_a$ and reusing it later), while Schummer and Vohra (2013) show that both Compression and TC fail an even weaker condition.[37] For completeness, we provide an example in Appendix C to show Compression is manipulable by postponing a flight cancellation in our preference domain.

The above non-manipulable condition is defined for deterministic mechanisms. We define the corresponding condition for stochastic mechanisms. A lottery mechanism ϕ is **manipulable by postponing a flight cancellation** if there are instances $I = (S, A, F^o, R, e, \Phi(A, \hat{S}))$ and $I' = (S \setminus \{s\}, A, F^o \setminus \{f\}, R, e, \Phi(A, \hat{S}))$, airline $a \in A$, $f \in F_a^o \setminus F_a$ and slot $s \in S_a$ such that $\exists \mathcal{L} \succ_a \phi^I(R, e)$, where $\mathcal{L} = \sum p_\Pi \cdot \Pi$ and each Π is some landing schedule that contains landing schedule for a $\Pi_a^{\overbrace{\phi_a^{I'}(R,e)\cup\{s\}}}$. In words, a lottery mechanism is manipulable by postponing a flight cancellation if an airline can gain by freezing a canceled flight in a slot $s \in S_a$ and then self-optimize using the slots in $\phi_a^{I'}(R, e) \cup s$.

Thereon 4-1: The multiple trading cycles mechanism ϕ is non-manipulable by postponing a flight cancellation.

The proof is based on the following ideas. Suppose airline a freezes a canceled flight $f \in F_a^o \setminus F_a$ in a slot $s \in S_a$, so both s and f are removed from the instance. First, the probabilities of getting better positions (for remaining flights in the main stage) in z are higher if f is not removed. For instance, the probability of $a(1)$ being the first flight in z is $\frac{|F_a^o - 1|}{|F^o - 1|}$ if f is removed, and such probability increases to $\frac{|F_a^o|}{|F^o|}$ if f is not removed. Second, removing s means a does not use s to trade, but MTC is ex post individually rational (with respect to S_A), putting s back into the instance would only make a weakly better off.

It is easy to see that from Claim 1 and Proposition 1, if a slot $s \in S^{0-0}$ is removed from

[37] Compression and TC fail a condition called non-manipulable via slot destruction—an airline cannot gain by freezing a canceled flight in a slot $s \in S_a$.

an instance, then each flight with $e_f \leq s$ would get a slot that is weakly later than otherwise, and if a flight $f \in F$ is removed from an instance, then each flight in $F \setminus \{f\}$ would get a weakly earlier slot in MTC (for any given z with f removed).

Can an airline gain by freezing a non-canceled flight $f_{a,i} \in F_a$ in a slot $s \in S_a$? The answer is maybe.

Theorem 4-2: Suppose $f_{a,i}$ is the most important flight of a in an instance I and MTC is used. If the earliest feasible available slot for $f_{a,i}$, s, is in S_a and s is a scarce resource, then a can weakly gain by freezing $f_{a,i}$ in s.

Putting s and $f_{a,i}$ into the instance would make a "pay" the position of $a(1)$ in any ordering z to get s even though s is in S_a (in this case, if a removes s and f, then the theorem applies to the next most important flight of a). However, if such s is a non-scarce resource, freezing $f_{a,i}$ in s makes a weakly worse off (this is by Theorem 4-1 because $f_{a,i}$ in this situation is effectively the same as a canceled flight to a in MTC). We illustrate these points by the following example.

Example 9:

Flight	$f_{a,1}$	$f_{a,2}$	$f_{b,1}$
e_f	1	1	1
$\underline{e_f}$	3	1	1
R	1	2	1
$\tilde{e_f}$	-	1	1
$\tilde{R}$	-	1	1

GDP slot	s_1	s_2	s_3
$\Phi(\cdot, S)$	a		
φ^{z^1} or φ^{z^4}	$f_{a,1}$	$f_{a,2}$	$f_{b,2}$
φ^{z^2} or φ^{z^3} or φ^{z^5}	$f_{a,1}$	$f_{b,1}$	$f_{a,2}$
$\underline{\Phi(\cdot, S)}$			a
$\underline{\varphi^{z^1}}$ or $\underline{\varphi^{z^2}}$ or $\underline{\varphi^{z^4}}$	$f_{a,2}$	$f_{b,1}$	$f_{a,1}$
$\underline{\varphi^{z^3}}$ or $\underline{\varphi^{z^5}}$	$f_{b,1}$	$f_{a,2}$	$f_{a,1}$

Let φ^z represent $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$ and $\underline{\varphi^z}$ represent $\varphi^z(\boldsymbol{R}, \underline{\boldsymbol{e}})$. Let $b(1)$ represent $f_{b,1}$.

Case 1: $s_1 \in \Phi(a, S)$ is a scarce resource ($e_{f_{a,1}} = 1$). There are three possible orderings: $z^1 = (a(1), a(2), b(1))$, $z^2 = (a(1), b(1), a(2))$, and $z^3 = (b(1), a(1), a(2))$. Let $a(i)$ represent $f_{a,i}$ for $i \in \{1, 2\}$. Each ordering realizes with probability $\frac{1}{3}$. While $f_{a,1}$ always get s_1 (under z^3, $a(1)$ would be inserted in front of $b(1)$), $f_{a,2}$ gets s_2 with probability $\frac{1}{3}$ (only under z^1). Consider if a freezes $f_{a,1}$ in s_1, so both $f_{a,1}$ and s_1 are removed from the instance. There are two possible orderings: $z^4 = (a(1), b(1))$ and $z^5 = (b(1), a(1))$. Let $a(1)$ represent $f_{a,2}$. Each ordering realizes with probability $\frac{1}{2}$. Therefore, $f_{a,2}$ gets s_2 with probability $\frac{1}{2}$, so its expected delay is lower.

Case 2: $s_3 \in \underline{\Phi(a, S)}$ is a non-scarce resource ($\underline{e_{f_{a,1}}} = 3$). s_3 is assigned to $f_{a,1}$ in the pre-competition stage. There are also three possible orderings z^1, z^2 and z^3 as in Case 1. Let $a(1)$ represent $f_{a,2}$ and $a(2)$ represent $f_{a,1}$ ($f_{a,1}$ is a duplicate flight). $f_{a,2}$ gets s_2 with probability $\frac{2}{3}$ (under z^1 and z^2). Consider if a freezes $f_{a,1}$ in s_3, so both $f_{a,1}$ and s_3 are removed from the instance. Again, there are also two possible orderings z^4 and z^5 as in Case

1. Let $a(1)$ represent $f_{a,2}$. Each ordering realizes with probability $\frac{1}{2}$. Therefore, $f_{a,2}$ gets s_1 with probability $\frac{1}{2}$, so its expected delay is higher.

Case 3: Consider when $f_{a,1}$ is canceled ($(\tilde{R}, \tilde{e})$ is reported). $s_3 \in \underline{\Phi(a, S)}$ is not in S^{0-0}. $a(2)$ now represents a dummy flight $f_{a,1}$. Everything else is the same as in Case 2.

The following mechanism is inspired by Theorem 4-2. We call the MTC that uses the following modified main stage **Multiple Trading Cycles-2** (MTC-2). All results of MTC hold for MTC-2. MTC and MTC-2 are not the same mechanism as MTC-2 possibly favors some airlines that fit the description of Theorem 4-2.

Modified Main Stage:

If this is the first assignment in a GDP, for each $a \in A$, construct $S_a = \Phi^{ex\,ante}(a, S)$ according to the initial slot ownership function $\Phi(A, S^o)$. Otherwise, let $S_a = \Phi(a, S)$ (from the last assignment). Start from the earliest slot in $S^{main} \cap S_A$, if a slot in S_a is the earliest feasible available slot in S^{main} to the most important flight of a in F^{main}, assign it to this flight and update S^{main} to $S^{(1)}$ and F^{main} to $F^{(1)}$ (in general, update $S^{(t)}$ to $S^{(t+1)}$ and $F^{(t)}$ to $F^{(t+1)}$). Then start from the earliest slot in $S^{(t)} \cap S_A$ for $t = 1, 2, ...$, repeat the above procedure until there is no more such slot. Denote the set of slots being assigned here by S^{top} and the set of flights that are assigned a slot here by F^{top}.

Creates $|F_a^o| - |F_a^{top}|$ surrogates of a for each $a \in A$, name them $a(1), a(2), ..., a(|F_a^o| - |F_a^{top}|)$. Denote the set of surrogates by $\mathcal{A}$ ($|\mathcal{A}| = |F^o| - |F^{top}|$). Randomly select an ordering $z(\mathcal{A})$ with uniform distribution over $Z(\mathcal{A})$. For each a, rearrange its surrogates to their positions in the ordering such that they are in the order of $a(1), a(2), ..., a(|F_a^o| - |F_a^{top}|)$. Denote the resulting ordering z.

Let $a(i) \in \{a(1), ...a(|F_a^{main}| - |F_a^{top}|)\}$ represent the i-th important flight in $F_a^{main} \setminus F_a^{top}$ (according to R_a); we call it a remaining flight of a. Let $a(i) \in a(|F_a^{main}| - |F_a^{top}| + 1), ..., a(|F_a| - |F_a^{top}|)$ represent the i-th important flight in $F_a \setminus F_a^{main}$ (according to R_a); we call it a duplicate flight of a. Let $a(i) \in a(|F_a| - |F_a^{top}| + 1), ..., a(|F_a^o| - |F_a^{top}|)$ represent a canceled flight of a; we call it a dummy flight of a.

Let $S^1 = S^{0-0} \setminus S^{top}$ and $F^1 = F \setminus F^{top}$. (We only change the procedure before Step 1.)

In Case 1 of Example 9, MTC-2 assigns s_1 to $f_{a,1}$ and then selects z^4 and z^5 randomly, while MTC selects z^1, z^2, and z^3 randomly. One can modify YGMH-IGYT by using the modified main stage: Fix an ordering. If a house is owned by an agent and the house is his top choice, assign this house to this agent. Repeat this procedure until there is no more such house. Then run YGMH-IGYT with the reduced ordering (with these agents eliminated). The outcome does not change because the positions of these agents in the original ordering are independent to the final outcome. Each agent that is assigned a house here always obtains the same house (in any individually rational mechanism). This modification is compatible with the previous one.

6.2 Multiple Runways

6.2.1 An Extended Model

When there are multiple runways, there will be multiple slots available at a time. Let m be the number of runways, so the set of available GDP slots is $S^m = \{s_{1,1}, s_{1,2}, ..., s_{1,m}, s_{2,1}, ...\}$. We assume airlines are indifferent between slots of the same time.[38] Since we need strict preferences in MTC, we can use tiebreaking rules to eliminate these indifferences.

Tiebreaking rule-1: Given S_A and a preference profile $\succsim$ induced by (R, e) with $e_f \in S$ for each flight f, construct a strict preference profile $\succ^{main}$ with $e_f \in S^m$ for each flight f as follows: For any airline a, given two slots of the same time,

(1) if both slots are in S_a or S_b (for some $b \in A$) or $S \setminus S_A$, then the slot with the lower index is strictly better,

(2) if one slot is in S_a and another is in S_b, then the one in S_a is strictly better,

(3) if one slot is in S_b and another is in S_c, then the one ________ is strictly better (see discussion below), and

(4) if one slot is in S_A and another is in $S \setminus S_A$, then the one in $S \setminus S_A$ is strictly better.

Anything (fixed or random) based on some exogenous parameters that does not create indifferences can be filled in the blank. Abdulkadiroğlu and Sönmez (1999) propose a tie-breaking rule where in the situation of (3), a slot owned by a higher ranked owner in z is preferred.[39] The intuition behind (2) and (4) is that trading with another airline (for a slot in S^{main}) is not free (potentially make some slots in S_a unavailable to itself), so an airline might want to avoid trading whenever possible. Under Tiebreaking rule-1, all else being equal, for any airline a, slots in $S \setminus S_A$ are the best, slots in S_a are in the middle, and slot in some S_b are the worst.

Tiebreaking rule-2: Given S_A and a preference profile $\succsim$ induced by (R, e) with $e_f \in S$ for each flight f, construct a strict preference profile $\succ^{pre-competition}$ with $e_f \in S^m$ for each flight f as follows: For any airline a, given two slots of the same time,

(1′) if both slots are in S_a or $S \setminus S_a$, then the slot with the lower index is strictly better, and

(2′) if one slot is in S_a and another is in $S \setminus S_a$, then the one in $S \setminus S_a$ is strictly better.

[38] The are many papers study allocation problems with non-strict preferences. See Quint and Wako (2004); Ergin (2008); Abdulkadiroğlu et al. (2009); Alcalde-Unzu and Molis (2011); Jaramillo and Manjunath (2012); Ehlers (2014); Erdil and Ergin (2017).

[39] Under that tiebreaking rule, in situation (4), the one in S_A is strictly better.

The intuition behind ($2'$) is that trading a slot $s \in S_a$ for a slot in $S \setminus S^{main}$ eliminates the potential gain from s in the main stage. Under Tiebreaking rule-2, all else being equal, for any airline a, slots in $S \setminus S_a$ are the best, slots in S_a are the worst.

6.2.2 Modified MTC

For the first assignment, we need to construct $\Phi^{ex\, ante}(a, S^m)$ for each $a \in A$. Let $S^{o,m} = \{s^o_{1,1}, s^o_{1,2}, ..., s^o_{1,m}, s^o_{2,1}, ...\}$. The indices of runways are not important in the construction of $\Phi^{ex\, ante}(a, S^m)$ as we treat $s^o_{1,r}$ for $r \in \{1, ..., m\}$ as s^o_1. We now construct $\Phi^{ex\, ante}(a, S^m)$ according to the initial slot ownership function $\Phi(A, S^{o,m})$: Select an arbitrary airline $a \in A$, if the time interval of a GDP slot s_n is entirely owned by airline a before the GDP starts, endow $s_{n,1}$ to a for each n.[40] Remove one copy of each initial slot that covers these s_n's. If s_n is still being covered (by the remaining set of initial slots), endow $s_{n,2}$ to a for each n. Procedure continues in a similar way until no more s_n is covered. Then select an arbitrary airline $b \in A$...(if a is endowed $s_{n,r}$ for some n, then other airlines might be endowed $s_{n,r+1}$, and so on. If in a situation where t slots at time s_n are being endowed but only r slots are available ($r < t$, there can be exempted flights, crossing runways, etc.), then remove all s_n's from $\Phi^{ex\, ante}(A, S^m)$).

Modified Pre-competition Stage:

(a) Order flights in F in increasing order of e_f (break ties arbitrarily).

(b) Assign flights sequentially to the earliest slot (start from the one with the lowest index) in S that each flight can feasibly use. Denote the tentative landing schedule by $\hat{\Pi}$ and the set of occupied slots by S^{0-0}.

(c) Let $F^{0-0} = F$. Find an earliest $s \in S^{0-0}$ such that

(c-i) $s = \hat{\Pi}^{-1}(f)$ for some $f \in F^{0-0}_a$ and $e_f > s(-1)$, where $s(-1)$ is some last slot before s in S^{0-0} (f occupies s in $\hat{\Pi}$ and will not compete for slots earlier than s. Since in $\hat{\Pi}$, all flights that arrive strictly earlier than f will get a slot *no later* (compare to "strictly earlier" in the single runway problem) than s, this condition also implies that $\forall f' \in F^{0-0}$ with $e_{f'} \leq s(-1)$, $\hat{\Pi}^{-1}(f') \leq s$. Therefore, there are sufficient slots to accommodate flights that arrive earlier than f, so they will not compete for s);

(c-ii) (a) If f' has $e_f \leq e_{f'} \leq s$, then $f' \in F^{0-0}_a$; or (b)$\nexists f'$ with $e_f \leq e_{f'} \leq s$ such that $s(+1) = \hat{\Pi}^{-1}(f')$, where $s(+1)$ is some next slot after s in S^{0-0} ((a) says all flights that want s belong to a, and (b) says each flight that wants s get a slot no later than s).

Assign s *tentatively* to $f_{a,i}$ if (c-ii)-(a) is satisfied, where $f_{a,i} \in F^{0-0}_a$ is the most important flight with $e_f \leq e_{f_{a,i}} \leq s$. Remove $f_{a,i}$ from F^{0-0} and s from S^{0-0}. If $f \neq f_{a,i}$, modify $\hat{\Pi}$

[40] Formally, $s_n = [1 + (n-1)l, 1 + nl] \subseteq \cup_{s^o_{n,r} \in \Phi(a, S^{o,m})} [n, n+1]$.

in the following way: Start from f, move each flight to the next slot in S^{0-0} until $\hat{\Pi}(f_{a,i})$ is filled.

Assign s *tentatively* to f if (c-ii)-(b) is satisfied (or both (c-ii)-(a) and (c-ii)-(b) are satisfied).

For $t = 0, 1, ...$, update F^{0-t} to $F^{0-(t+1)}$ and S^{0-t} to $S^{0-(t+1)}$, respectively.

(d) Repeat (c) until all slots are demanded by more than 1 airline.

Denote the resulting set of flights F^{main}.

(Since both main stage and modified main stage use S^{main} as an input. We need to construct S^{main}. Unlike the single runway problem, S^{main} might be different from the last S^{0-t} (see Example 13).)

Construct S_a for each $a \in A$ as in MTC. Creates $|F_a \setminus F_a^{main}|$ surrogates of a for each $a \in A$, name them $a(1), a(2), ..., a(|F_a \setminus F_a^{main}|)$. Denote the set of surrogates by $\mathcal{A}^0$ ($|\mathcal{A}^0| = | \cup_{a \in A} F_a \setminus F_a^{main}|$). Randomly select an ordering $z(\mathcal{A}^0)$ with uniform distribution over $Z(\mathcal{A}^0)$. For each a, rearrange its surrogates to their positions in the ordering such that they are in the order of $a(1), a(2), ..., a(|F_a \setminus F_a^{main}|)$. Denote the resulting ordering z^0. Let $a(i) \in a(1), ..., a(|F_a \setminus F_a^{main}|)$ represent the i-th important flight in $F_a \setminus F_a^{main}$ (according to R_a). According to Tiebreaking rule-2, tentatively assign the first flight in z^0 the slot it wants, then tentatively assign the second flight in z^0 the slot it wants (among those remaining), and so on.

Denote the resulting set of slots S^{main}.

(Each slot that is tentatively assigned will be assigned to one of the flights that obtains a slot of the same time here in the main stage. Assigning slots tentatively allows us to apply Tiebreaking rule-1 for all flights in the main stage (Tiebreaking rule-2 has fulfilled its mission already); otherwise, we would need to apply Tiebreaking rule-2 for duplicate flights. The two other stages are the same except the S_a for each $a \in A$ is constructed earlier and a duplicate flight will be assigned pro forma the slot it was assigned in the pre-competition stage or a slot of the same time. The final assignment depends on the realized ordering in the main stage and Tiebreaking rule-1.

We now say s is demanded only by *some* airline a if (c-i) and (c-ii) hold simultaneously and s is demanded by more than 1 airline if either (c-i) or (c-ii) fails. Recall the 4 types of slots: Type 1 slots satisfy (c-i) but not (c-ii), type 2 slots do not satisfy both, type 3 slots satisfy (c-ii) but not (c-i), and type 4 slots satisfy both.)

Example 10:

Flight	$f_{a,1}$	$f_{b,1}$	$f_{c,1}$	$f_{d,1}$	$f_{e,1}$
e_f	1	1	2	4	4

GDP slot (Runway 1)	$s_{1,1}$	$s_{2,1}$	$s_{3,1}$	$s_{4,1}$
GDP slot (Runway 2)				$s_{4,2}$
$\hat{\Pi}$	$f_{a,1}$	$f_{b,1}$	$f_{c,1}$	$f_{d,1}$
				$f_{e,1}$

This example is modified from Example 1. We only need to check the new condition (c-ii)-(b). $s_{1,1}$ is a type 1 slot since (c-i) is satisfied, (c-ii)-(a) fails, and (c-ii)-(b) fails as $\hat{\Pi}(f_{b,1}) = s(+1) = s_{2,1}$ with $e_{f_{a,1}} \leq e_{f_{b,1}} \leq s_{1,1}$. $s_{2,1}$ is a type 2 slot since (c-i) fails, (c-ii)-(a) fails, and (c-ii)-(b) fails as $\hat{\Pi}(f_{c,1}) = s(+1) = s_{3,1}$ with $e_{f_{b,1}} \leq e_{f_{c,1}} \leq s_{2,1}$. $s_{3,1}$ is a type 3 slot since (c-i) fails, (c-ii)-(a) is satisfied, and (c-ii)-(b) is satisfied (there is no such f'). $s_{4,1}$ is a type 4 slot as (c-i) and (c-ii)-(b) hold simultaneously ((c-ii)-(a) fails because $f_{d,1} \leq f_{e,1} \leq s_{4,1}$ but $f_{e,1} \notin F_d^{0-t}$). $s_{4,2}$ is also a type 4 slot by similar reasoning. Now consider the following modified Example.

Example 11:

Flight	$f_{a,1}$	$f_{b,1}$	$f_{c,1}$	$f_{d,1}$	$f_{d,2}$	$f_{d,3}$
e_f	1	1	2	4	4	4

GDP slot (Runway 1)	$s_{1,1}$	$s_{2,1}$	$s_{3,1}$	$s_{4,1}$	$s_{5,1}$
GDP slot (Runway 2)				$s_{4,2}$	
$\hat{\Pi}$	$f_{a,1}$	$f_{b,1}$	$f_{c,1}$	$f_{d,1}$	$f_{d,3}$
				$f_{d,2}$	

$s_{4,1}$ and $s_{4,2}$ is a type 4 slot as (c-i) and (c-ii)-(a) hold simultaneously (for $s_{4,1}$, (c-ii)-(b) fails as $\hat{\Pi}(f_{d,3}) = s(+1) = s_{5,1}$ with $f_{d,1} \leq f_{d,3} \leq s_{4,1}$. The same for $s_{4,2}$). $s_{5,1}$ is also a type 4 slot in $S^{0-(t+2)}$ (after the removals of $s_{4,1}$ and $s_{4,2}$) as (c-i), (c-ii)-(a) and (c-ii)-(b) hold simultaneously.

Remark: If a slot $s_{n,1}$ does not satisfy (c-ii) (some flight of other airlines also wants this slot but has been assigned a later slot in $\hat{\Pi}$), then each $s_{n,r}$ also fails (c-ii). But if a slot $s_{n,1}$ does not satisfy (c-i), some $s_{n,r}$ might satisfy (c-i). Therefore, type 1 and type 2 slots may coexist in some $(s_{n,1}, s_{n,2}, ..., s_{n,m})$ (by construction, type 2 slots have lower indices). Similarly, type 3 and type 4 slots may coexist in some $(s_{n,1}, s_{n,2}, ..., s_{n,m})$ (a slot is type 3 or type 4 is determined in the pre-competition stage. Also, there can be slots in $S \setminus S^{0-0}$ here as well).

Example 12:

Flight	$f_{a,1}$	$f_{b,1}$	$f_{c,1}$	$f_{d,1}$
e_f	1	1	2	2

GDP slot (Runway 1)	$s_{1,1}$	$s_{2,1}$	$s_{3,1}$
GDP slot (Runway 2)		$s_{2,2}$	
$\hat{\Pi}$	$f_{a,1}$	$f_{b,1}$	$f_{d,1}$
		$f_{c,1}$	

In Example 12, $s_{2,1}$ is a type 2 slots and $s_{2,2}$ is a type 1 slot. Note that $s_{1,1}$ is also a type 1 slot. $s_{3,1}$ is a type 3 slot since (c-i) fails but (c-ii)-(b) is satisfied.

Example 13:

Flight	$f_{a,1}$	$f_{b,1}$	$f_{c,1}$
e_f	1	1	2

GDP slot (Runway 1)	$s_{1,1}$	$s_{2,1}$
GDP slot (Runway 2)		$s_{2,2}$
$\hat{\Pi}$	$f_{a,1}$	$f_{b,1}$
		$f_{c,1}$

In Example 13, $s_{2,1}$ is a type 3 slots and $s_{2,2}$ is a type 4 slot. Suppose now $s_{2,2} \in \Phi(c, S)$. In the modified pre-competition stage, $s_{2,2}$ is tentatively assigned to $f_{c,1}$, so $S^{0-1} = \{s_{1,1}, s_{2,1}\}$. However, according to Tiebreaking rule-2, $f_{c,1}$ will pick $s_{2,1}$. Therefore, $S^{main} = \{s_{1,1}, s_{2,2}\} \neq S^{0-1}$. Now $s_{2,2}$ is a type 3 slots and $s_{2,1}$ is a type 4 slot. Note that if there is a $s_{2,3}$, then it is in $S \setminus S^{0-0}$.

We call $(s_{n,1}, s_{n,2}, ..., s_{n,m})$ a **slot group**. We say a slot group it a *type x slot group* if all slots in the slot group are type x slots. Scarce resources in the extended model are sequences of adjacent *slot groups* in S^{main} such that each **sequence** starts with a type 1 slot group and ends with type 3 slots. Note that $s_{2,2}$ in Example 13 (without the supposition) is not in any sequence as it is not a scarce resource. Suppose there is no gap in S^{main} and there are always m slots at time s_n, a sequence is of the form

$$((s_{n,1}, s_{n,2}, ..., s_{n,m}), (s_{n+1}, s_{n+1,2}, ..., s_{n+1,m})..., (s_{n+t}, s_{n+t,2}, ..., s_{n+t,m})),$$

where $(s_{n,1}, s_{n,2}, ..., s_{n,m})$ is a type 1 slot group and $(s_{n+t,1}, s_{n+t,2}, ..., s_{n+t,m})$ contains some type 3 slots. Each sequence contains one type 1 slot group and possibly more type 1 slots (as in Example 12), some type 3 slots, and possibly some type 2 slots. As before, in any feasible and non-wasteful landing schedule, a flight that gets a slot in a sequence in $\hat{\Pi}$ will *always* get a slot in the same sequence and a flight that gets a slot outside a sequence in $\hat{\Pi}$ will *never* get a slot in that sequence.

Since airlines are indifferent between slots of the same time, there might be multiple individually rational landing schedules for some airline a. But a is indifferent between any of these landing schedules as each of its flights would get a slot of the same time only. The next proposition is the main result for this extended model. We call the MTC that uses the modified pre-competition stage and Tiebreaking rule-1 in the main stage **modified MTC with tiebreaking** (by replacing MTC with MTC-2, we can define *modified MTC-2 with tiebreaking.* Again, all results are the same).

Proposition 4: The results of Claim 1, Proposition 1, 2, 3 and Theorem 1, 3, 4-1, 4-2 hold for the modified MTC with tie-breaking in the extended model.

This result is for the true preference profile $\succsim$.[41] Example 14 shows the modified MTC with tie-breaking might produce outcomes outside the core.

Example 14:

Flight	$f_{a,1}$	$f_{a,2}$	$f_{b,1}$	$f_{c,1}$	$f_{d,1}$	$f_{e,1}$
e_f	1	2	3	2	1	1

GDP slot (Runway 1)	$s_{1,1}$	$s_{2,1}$	$s_{3,1}$	$s_{4,1}$	$s_{5,1}$
GDP slot (Runway 2)	$s_{1,2}$				
$\Phi(\cdot, S)$	b	a	a		
	c				
Case 1	$f_{a,1}$	$f_{a,2}$	$f_{b,1}$	$f_{c,1}$	$f_{e,1}$
	$f_{d,1}$				
Case 2	$f_{a,1}$	$f_{c,1}$	$f_{b,1}$	$f_{a,2}$	$f_{e,1}$
	$f_{d,1}$				

In situation (3) of Tiebreaking rule-1, break ties based on alphabetical order. That is, slots in S_a are better than slots in S_b, etc. There is no non-scarce resource in this example, so we can skip the pre-competition stage.

$$z = (a(1), a(2), c(1), b(1), d(1), e(1)).$$

Let $a(i)$ represent $f_{a,i}$ for $i = \{1, 2\}$ and others represent their only flights.

Case 1: $a(1)$ picks $s_{1,1} \in S_b$ for $f_{a,1}$, then $b(1)$ will be inserted in front of $a(1)$. $b(1)$ picks $s_{3,1} \in S_a$ for $f_{b,1}$. $a(1)$ and $b(1)$ form a cycle. $f_{a,1}$ is assigned $s_{1,1}$ and $f_{b,1}$ is assigned $s_{3,1}$. Then $s_{2,1}$ goes to $a(2)$, $s_{4,1}$ goes to $c(1)$, $s_{1,2}$ goes to $d(1)$, and $s_{5,1}$ goes to $e(1)$.

Now suppose MTC breaks ties based on reverse alphabetical order.

Case 2: $a(1)$ picks $s_{1,2} \in S_c$ for $f_{a,1}$, then $c(1)$ will be inserted in front of $a(1)$. $c(1)$ picks $s_{2,1} \in S_a$ for $f_{c,1}$. $a(1)$ and $c(1)$ form a cycle. $f_{a,1}$ is assigned $s_{2,1}$ and $f_{c,1}$ is assigned $s_{2,1}$. Then $s_{3,1}$ goes to $a(2)$, $s_{4,1}$ goes to $b(1)$, $s_{1,2}$ goes to $d(1)$, and $s_{5,1}$ goes to $e(1)$.

Note that both landing schedules are Pareto efficient and individually rational but the Case 2 landing schedule is not in the core because it is dominated by the Case 1 landing schedule via subgroup $\{a, b\}$. Yet MTC respects property rights over S_a because the definition only requires that a trades a slot in S_a for a better slot and $s_{1,2}$ is better than $s_{2,1}$ and $s_{3,1}$. When $f_{a,1}$ is picking a slot, a is in two cycles under its true preference, where one of the cycles leads to a better outcome for a.[42] This does not happen in models where agents have unit demand and non-strict preferences.[43]

[41] Indeed, Proposition 4 also holds for $\succ^{main}$.

[42] Indeed, condition (iv) in the Proof of Theorem 2 no longer holds in the extended model because now $f_{a,j}$ does not have to pick in $S_{A'}$. In Example 14, $f_{a,1}$ can pick $s_{1,2} \notin S_{A'} = S_a \cup S_b$.

[43] In such models, an agent could be in more than one cycle under his true preference, but none of the cycles would lead to a better outcome for the agent.

7 Conclusion

This paper studies airport slot allocation problems. When inclement weather strikes a heavily used airport, its landing schedule must be reconfigured as now it requires more time to land a plane. Some flights have to be postponed, but such postponements may be too costly to airlines, and so the airlines might cancel those flights. Cancellations and delays create vacant slots in the landing schedule, which are new resources to be reallocated.

We argue that the currently used mechanism does not respect property rights before and after a GDP starts. The mechanism we proposed solicits private information such as earliest feasible arrival times and importance rankings from the airlines. Based on this information, the mechanism produces outcomes that are individually rational, Pareto efficient and in the core. Our mechanism also respects property rights before and after a GDP starts, is strategy-proof and non-manipulable by postponing a flight cancellation.

In the extended model with multiple runways, a modified version of our mechanism with tiebreaking inherits most of the aforementioned properties but might produce outcomes outside the core. It remains an open question whether there exists some endogenous tie-breaking rules that can resolve this problem.

References

Abdulkadiroğlu, A., P. A. Pathak, and A. E. Roth (2009). Strategy-proofness versus efficiency in matching with indifferences: Redesigning the nyc high school match. *The American Economic Review 99*(5), 1954–1978. 38

Abdulkadiroğlu, A. and T. Sönmez (1999). House allocation with existing tenants. *Journal of Economic Theory 88*(2), 233–260. (document), 1, 2, 16, 4.1, 6.2.1

Abizada, A. and U. M. Dur (2017). College admissions with complementarities. 21

Air Transport Association (2008). 2008 economic report. Technical report. 3

Alcalde-Unzu, J. and E. Molis (2011). Exchange of indivisible goods and indifferences: The top trading absorbing sets mechanisms. *Games and Economic Behavior 73*(1), 1–16. 38

Balakrishnan, H. (2007). Techniques for reallocating airport resources during adverse weather. In *Decision and Control, 2007 46th IEEE Conference*, pp. 2949–2956. IEEE. 2

Ball, M., C. Barnhart, M. Dresner, M. Hansen, K. Neels, A. Odoni, E. Peterson, L. Sherry, A. Trani, and B. Zou (2010). Total delay impact study: A comprehensive assessment of the costs and impacts of flight delay in the united states. Technical report, NEXTOR Report Prepared for the Federal Aviation Administration. 3

Ball, M., G. Dahl, and T. Vossen (2009). Matchings in connection with ground delay program planning. *Networks 53*(3), 293–306. 18

Ball, M. O., R. Hoffman, and A. Mukherjee (2010). Ground delay program planning under uncertainty based on the ration-by-distance principle. *Transportation Science 44*(1), 1–14. 2

Bard, J. F. and D. N. Mohan (2008). Reallocating arrival slots during a ground delay program. *Transportation Research Part B: Methodological 42*(2), 113–134. 18

Chun, Y. and B. Park (2017, Jan). A graph theoretic approach to the slot allocation problem. *Social Choice and Welfare 48*(1), 133–152. 2

Dutta, B. and J. Massó (1997). Stability of matchings when individuals have preferences over colleagues. *Journal of Economic Theory 75*(2), 464–475. 21

Ehlers, L. (2002). Multiple public goods and lexicographic preferences: replacement principle. *Journal of Mathematical Economics 37*(1), 1–15. 21

Ehlers, L. (2003). Multiple public goods, lexicographic preferences, and single-plateaued preference rules. *Games and Economic Behavior 43*(1), 1–27. 21

Ehlers, L. (2014). Top trading with fixed tie-breaking in markets with indivisible goods. *Journal of Economic Theory 151*, 64–87. 38

Erdil, A. and H. Ergin (2017). Two-sided matching with indifferences. *Journal of Economic Theory 171*, 268–292. 38

Ergin, H. (2008). What's the matter with tie-breaking? improving efficiency in school choice. *The American Economic Review 98*(3), 669–689. 38

Fujita, E., J. Lesca, A. Sonoda, T. Todo, and M. Yokoo (2015). A complexity approach for core-selecting exchange with multiple indivisible goods under lexicographic preferences. In *AAAI*, pp. 907–913. 21

Glover, C. N. and M. O. Ball (2013). Stochastic optimization models for ground delay program planning with equity–efficiency tradeoffs. *Transportation Research Part C: Emerging Technologies 33*, 196–202. 18

Hylland, A. and R. Zeckhauser (1979). The efficient allocation of individuals to positions. *Journal of Political economy 87*(2), 293–314. 2

Jaramillo, P. and V. Manjunath (2012). The difference indifference makes in strategy-proof allocation of objects. *Journal of Economic Theory 147*(5), 1913–1946. 38

Kennes, J., D. Monte, and N. Tumennasan (2014). The day care assignment: A dynamic matching problem. *American Economic Journal: Microeconomics 6*(4), 362–406. 2

Konishi, H., T. Quint, and J. Wako (2001). On the shapley–scarf economy: the case of multiple types of indivisible goods. *Journal of mathematical economics 35*(1), 1–15. 2

Kurino, M. (2014). House allocation with overlapping generations. *American Economic Journal: Microeconomics 6*(1), 258–289. 2

Pereyra, J. S. (2013). A dynamic school choice model. *Games and economic behavior 80*, 100–114. 2

Quint, T. and J. Wako (2004). On houseswapping, the strict core, segmentation, and linear programming. *Mathematics of Operations Research 29*(4), 861–877. 38

Roth, A. E. and A. Postlewaite (1977). Weak versus strong domination in a market with indivisible goods. *Journal of Mathematical Economics 4*(2), 131–137. 17

Roth, A. E. and M. Sotomayor (1992). Two-sided matching. *Handbook of game theory with economic applications 1*, 485–541. 35

Saban, D. and J. Sethuraman (2014). A note on object allocation under lexicographic preferences. *Journal of Mathematical Economics 50*, 283–289. 21

Schulman, L. J. and V. V. Vazirani (2012). Allocation of divisible goods under lexicographic preferences. *arXiv preprint arXiv:1206.4366*. 21

Schummer, J. and A. Abizada (2017). Incentives in landing slot problems. *Journal of Economic Theory 170*, 29–55. 1, 2, 15, 3.1, 29, 6.1, C

Schummer, J. and R. V. Vohra (2013). Assignment of arrival slots. *American Economic Journal: Microeconomics 5*(2), 164–185. 7, 1, 9, 11, 12, 2, 14, 3.2, 29, 6.1, B, C

Shapley, L. and H. Scarf (1974). On cores and indivisibility. *Journal of mathematical economics 1*(1), 23–37. 2, 17

Sönmez, T. and M. U. Ünver (2005). House allocation with existing tenants: an equivalence. *Games and Economic Behavior 52*(1), 153–185. 16

Sönmez, T. and M. U. Ünver (2010). House allocation with existing tenants: A characterization. *Games and Economic Behavior 69*(2), 425–445. 16

Vossen, T. (2002). *Fair allocation concepts in air traffic management*. Ph. D. thesis, University of Maryland, College Park. 2

Vossen, T. and M. Ball (2006a). Optimization and mediated bartering models for ground delay programs. *Naval Research Logistics (NRL) 53*(1), 75–90. 7, 13, 18

Vossen, T. W. and M. O. Ball (2006b). Slot trading opportunities in collaborative ground delay programs. *Transportation Science 40*(1), 29–43. 18

Wambsganss, M. (1996). Collaborative decision making through dynamic information transfer. *Air Traffic Control Quarterly 4*(2), 109–125. 4.3

A Omitted Proofs

(Proofs are in the language of Algorithm 1 unless otherwise noted.)

Proof of Theorem 1: The two algorithms only differ in their main stages. For any set of slot $S' \subset S$ and set of flights $F' \subseteq F$, the main stage of Algorithm 1 assigns (or assigns pro forma) the next series of slots in one of two possible ways.

Case 1: There is a chain $(a(i), s_b, b(j), s_c, .., s_d, d(k), s)$ (which may consist of a single flight $a(i)$ and a single slot s), where $a(i), b(j), ..., d(k)$ are most important remaining/duplicate flights, and $a(i)$ has the highest priority in z and demands s_b, $b(j)$ demands s_c,..., $d(k)$ demands s in $S' \setminus S_A$ or $S' \cap S_e$ but e has no remaining/duplicate flight in F'. Every flight in the chain is assigned (or assigned pro forma) the slot that it demands. Note that this is a cycle in Algorithm 2 for (S', F'). (Case 1 includes (ii)-(b), (iii-i), and (iii-iv)-(b).)

Case 2: There is a cycle $(a(i), s_b, b(j), ..., s_d, d(k), s_a)$ (which may consist of a single flight $a(i)$ and a single slot s_a), and every flight in the cycle is assigned (or assigned pro forma) the slot that it demands. This is also a cycle in Algorithm 2 for (S', F'). (Case 2 includes (ii)-(a), (iii-ii), (iii-iii), (iii-iv)-(a), and (iii-iv)-(c).)

Hence Algorithm 1 finds a cycle in Algorithm 2 and implements the associates trades for any set of slots and set of flights. In Algorithm 2, a cycle that is not removed at any step remains a cycle at the next step, so for any given ordering z, the main stages of Algorithm 1 and Algorithm 2 produce the same outcome. ■

Proof of Claim 1: Suppose not. $\exists s \in S^{0-0}$ such that $\Pi^{-1}(s) = \emptyset$, where Π is some feasible and non-wasteful landing schedule. By feasibility, the number of slots earlier than s that can be occupied is fixed. $\Pi^{-1}(s) = \emptyset$ but $s \in S^{0-0}$ then implies $\exists f \in F$ such that $e_f \leq s < \Pi(f)$. Contradicts to the non-wastefulness of Π. ■

Proof of Proposition 1: Regularity: This is by construction of the mechanism. Ex post feasibility: For any ordering z, at each stage and each step, no flight gets an infeasible slot. Ex post non-wastefulness: This is also by construction of the mechanism. For any ordering z, let φ^z be the induced schedule mechanism. If $\exists f \in F$ such that $s \in V$ with $e_f < s$, then it must be the case that $\varphi^z_f(\boldsymbol{R}, \boldsymbol{e}) < s$.

Respects property rights over S_a: Without loss of generality, we can focus on the main stage. At each step a slot in some S_a is being assigned (or assigned pro forma), there are three possibilities: (i) It is assigned to the most important remaining/duplicate flight of a; (ii) a trades it for a better slot for its most important remaining/duplicate flight; (iii) a has no more remaining/duplicate flight and this slot is assigned to some airline in the main stage (or the supplemental stage). In (iii), there is no better slot for a. ■

Proof of Proposition 2: For any ordering z, let φ^z be the induced schedule mechanism.

Let $x_i = d_{f_{a,i}}(\Pi_a^{S_a}) - d_{f_{a,i}}(\varphi_a^z(\boldsymbol{R}, \boldsymbol{e}))$ for $i \in \{1, ..., |F_a|\}$ and $f_{a,i} R_a f_{a,i+1}$.

If $x_i = 0$ for all $i \in \{1, ..., |F_a|\}$, then $\Pi_a^{S_a} = \varphi_a^z(\boldsymbol{R}, \boldsymbol{e})$. Otherwise, let x_j be the first non-zero coordinate of $x_a = (x_1, x_2, ..., x_{|F_a|})$. x_j will always be positive since airline a picks $\varphi_{f_{a,j}}^z(\boldsymbol{R}, \boldsymbol{e})$ instead of $\Pi_a^{S_a}(f_{a,j})$, which means $e_{f_{a,i}} \leq \varphi_{f_{a,j}}^z(\boldsymbol{R}, \boldsymbol{e}) < \Pi_a^{S_a}(f_{a,j})$. Hence,$\forall a \in A$, $\varphi_a^z(\boldsymbol{R}, \boldsymbol{e}) \succsim_a \Pi_a^{S_a}$. ■

Proof of Proposition 3: (We use the language of Algorithm 2 for this proof.) Fights that leave in the pre-competition stage are already getting the earliest slot they can get without hurting any flights that are more important than them within their airlines, and no slot in $S \setminus S^{main}$ can be used to make flights leave in the main stage better off.

Consider the main stage, for any ordering z, 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 remaining) and cannot be made better off without hurting some flight who left at Step 1. Proceeding in a similar fashion, 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 cannot make itself better off as well. Therefore, ϕ is ex post Pareto efficient. ■

Proof of Theorem 2: For any ordering z, let φ^z be the induced schedule mechanism. Suppose $\exists \Pi'$ and $A' \subseteq A$ such that (i) $\forall f \in \cup_{a \in A'} F_a$, $\Pi'(f) \in S_{A'}$, and (ii) $\forall a \in A'$, $\Pi' \succ_a \varphi^z(R, e)$. Therefore, $\forall a \in A'$, the first non-zero coordinate of $x_a = (x_1, x_2, ..., x_{|F_a|})$ is positive where $x_i = d_{f_{a,i}}(\varphi_a^z(\boldsymbol{R}, \boldsymbol{e})) - d_{f_{a,i}}(\Pi')$ for $i \in \{1, ..., |F_a|\}$ and $f_{a,i} R_a f_{a,i+1}$.

Consider $f_{a,i}$ where x_i is the first non-zero coordinate of x_a for $a \in A'$. Note that $\Pi'(f_{a,i}) \in S_{A'}$ is better than $\varphi_{f_{a,i}}^z(\boldsymbol{R}, \boldsymbol{e})$, and $\Pi'(f_{a,i})$ is not available when $f_{a,i}$ is picking a slot in $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$, so $\Pi'(f_{a,i})$ is not assigned in the supplemental stage. 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 makes airlines in A' prefer Π'.

(i) If $\Pi'(f_{a,i})$ is used by some $f_{a,j}$ in $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$, then it must be $f_{a,j} R_a f_{a,i}$. Since x_i is the first non-zero coordinate, $x_j = 0$, i.e., $f_{a,j}$ is getting the same slot in Π', a contradiction.

The same argument applies to all airlines in A'. Therefore, $\forall a \in A'$, $\Pi'(f_{a,i})$ is used by some other airline $a' \in A'$ in $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$.

The fact that $\Pi'(f_{a,i})$ is not available when $f_{a,i}$ is picking a slot in $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$ together with (i) implies $\Pi'(f_{a,i})$ is not assigned in the pre-competition stage, so it must be assigned in the main stage. Let $s_a \in S_T \cap S_a$ $(a \in A')$ be the first slot in S_T that is being assigned to some $f \in F$ in $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$. a will pick a slot for its most important remaining flight $f_{a,j}$ before s_a is assigned (either it is $f_{a,j}$'s turn or $f_{a,j}$ has been inserted; if a has no remaining flight, then it will be in case (iii) below). At this point, 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 $\varphi^z_{f_{a,i}}(\boldsymbol{R}, \boldsymbol{e})$ but not $\Pi'(f_{a,i})$, a contradiction.

(iii) If $f_{a,i} R_a f_{a,j}$, this means $\Pi'(f_{a,i})$ is still available after $f_{a,i}$ picked a slot in $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$, a contradiction.

(iv) If $f_{a,j} R_a f_{a,i}$, we have $\varphi^z_{f_{a,j}}(\boldsymbol{R}, \boldsymbol{e}) = \Pi'(f_{a,j}) \in S_{A'}$.

By (iv), $\varphi^z_{f_{a,j}}(\boldsymbol{R}, \boldsymbol{e}) \neq s_a$ otherwise it contradicts the supposition that s_a makes some airline in A' prefer Π'. Therefore, $f_{a,j}$ picks some slot other than s_a. That means airline a trades s_a for $\varphi^z_{f_{a,j}}(\boldsymbol{R}, \boldsymbol{e}) \in S_b$ and $b \in A'$.

(♠) Let $\varphi^z_{f_{b,j}}(\boldsymbol{R}, \boldsymbol{e})$ be the slot obtained by b in this trade. Because all slots in S_T are still available, the flight $f_{b,j}$ is more important than $f_{b,i}$, so $\varphi^z_{f_{b,j}}(\boldsymbol{R}, \boldsymbol{e}) = \Pi'(f_{b,j}) \in S_{A'}$. If $\varphi^z_{f_{b,j}}(\boldsymbol{R}, \boldsymbol{e}) \in S_a$ we have a cycle.

If $\varphi^z_{f_{b,j}}(\boldsymbol{R}, \boldsymbol{e}) \in S_c$, $c \in A'$ will be the next airline in this trade, and the argument (♠) applies. Because none of the airlines in this trade gets a slot outside $S_{A'}$ and A' is finite, there must exist a cycle contains exclusively airlines in A'. Let $y \in A'$ be the airlines gets s_a for $f_{y,j}$. Recall $\Pi'(f_{y,i})$ is not available when $f_{y,i}$ is picking a slot in $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$. Since all slots in S_T are still available, $f_{y,j}$ is more important than $f_{y,i}$, and therefore $\varphi^z_{f_{y,j}}(\boldsymbol{R}, \boldsymbol{e}) = \Pi'(f_{y,j}) = s_a$. This contradicts the supposition that s_a makes some airline in A' prefer Π'. ■

Proof of Theorem 3: For any ordering z, let φ^z be the induced schedule mechanism. We inspect each stage to see if an airline a can be better off by misreporting R_a or e_a, that is, $\varphi^z(\widehat{\boldsymbol{R}_a, \boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a}) \succ_a \varphi^z(\boldsymbol{R}, \boldsymbol{e})$. Let $x_a = (x_1, x_2, ..., x_{|F_a|})$ be a vector where $x_i = d_{f_{a,i}}(\varphi^z(\boldsymbol{R}, \boldsymbol{e})) - d_{f_{a,i}}(\varphi^z(\widehat{\boldsymbol{R}_a, \boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a}))$ for $i \in \{1, ..., |F_a|\}$ and $f_{a,i} R_a f_{a,i+1}$. Let x_j be the first non-zero coordinate of x_a. If there is no such x_j, then $\varphi^z(\widehat{\boldsymbol{R}_a, \boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a}) \sim_a \varphi^z(\boldsymbol{R}, \boldsymbol{e}))$, and we are done. Suppose not. $\varphi^z(\widehat{\boldsymbol{R}_a, \boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a}) \succ_a \varphi^z(\boldsymbol{R}, \boldsymbol{e})$ implies that x_j is positive, and so there exists some slot $s \in S$ such that $e_{f_{a,j}} \leq s = \varphi^z_{f_{a,j}}(\widehat{\boldsymbol{R}_a, \boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a}) < \varphi^z_{f_{a,j}}(\boldsymbol{R}, \boldsymbol{e})$. Note that s is not used by a flight of a that is more important than $f_{a,j}$ since that would contradict x_j is the first non-zero coordinate of x_a.

In the pre-competition stage, the only possible way for airline a to get a slot that is feasible for $f_{a,j}$ but not being assigned in this stage is to make it a non-scarce resource (type 4 slot) such that it will be assigned to $f_{a,j}$. For contradiction, suppose there is a slot s', which might be a type 1, type 2, or type 3 slot in the sequence that contains s, that can be converted to a type 4 slot and assigned to $f_{a,j}$ in the pre-competition stage. It is sufficient to show that $s \neq s'$ for all such s'.

Case 1: Suppose s' is the type 1 slot of the sequence. a can make s' a type 4 slot only when s' violates (c-ii) because of a (with $f_{a,x}, f_{a,x'}, ...$) and at most one airline b. Then a can misreport $\widehat{e_{f_{a,x}}, e_{f_{a,x'}}}, ...$ (some infeasible or later times) to give that slot to airline b. Since a cannot have such slot assigned to $f_{a,j}$, $s \neq s'$. (Note that this procedure (of misreporting $\widehat{e_{f_{a,x}}, e_{f_{a,x'}}} ...$) makes the type 1 slot satisfy (c-ii). It also makes the next slot a type 1 slot if it was a type 2 slot; moreover, it makes the next slot a type 4 slot if it was a type 3 slot. In other words, this procedure makes the next slot satisfy (c-i).)

Case 2: Suppose s' is some type 2 slot of the sequence. a can make s' a type 4 slot only by making it a type 1 slot first and then converting it to a type 4 slot (all by repeating the procedure in Case 1, if not impossible). But, again, a cannot have such slot assigned to $f_{a,j}$, so $s \neq s'$. (Observe that a can convert a type 2 slot to a type 3 slot only when the slot violates (c-ii) because of a (with $f_{a,x}, f_{a,x'}, ...$) and at most one airline b. Then a can misreport $\widehat{e_{f_{a,x}}, e_{f_{a,x'}}}, ...$ to make it a type 3 slot (with (c-ii) satisfied). But there is no way for a to convert it to a type 4 slot after that.)

Case 3: Suppose s' is some type 3 slot of the sequence. a can make s' a type 4 slot only when s' violates (c-i) because of a. Then a can change $e_{f_{a,j}}$ to s' $(> e_{f_{a,j}})$ and have this slot assigned to $f_{a,j}$ (if there is more than 1 such type 3 slots. Let flights that are assigned type 3 slots in $\hat{\Pi}$ be $f_{a,x}, f_{a,x'}, ...$ a also needs to misreport $\widehat{e_{f_{a,x}}, e_{f_{a,x'}}}, ...$ But this even requires $f_{a,j}$ to be the most important flight among $f_{a,j}, f_{a,x}, f_{a,x'}, ...$ since a is only willing to sacrifice flights that are less important than $f_{a,j}$ to help it). However, $f_{a,j}$ can always get a weakly earlier slot in the main stage. $s = s'$ would contradict x_j is positive $(\varphi^z_{f_{a,j}}(\boldsymbol{R}, \boldsymbol{e}) \leq \varphi^z_{f_{a,j}}(\widehat{\boldsymbol{R}_a, \boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a}) = s' = s)$, so $s \neq s'$.

Suppose a wants to manipulate the pre-competition stage to help $f_{a,j}$ to get a slot in the main stage. The only thing it might be able to change is S^{main}. By sacrificing some flights $f_{a,x}, f_{a,x'}, ...$ that are less important than $f_{a,j}$, it might shrink the size of S^{main} (as described in Case 1 to 3 above), or it might enlarge the size of S^{main} by reporting $\widehat{e_{f_{a,x}}, e_{f_{a,x'}}}, ...$ so that some slots $\varphi^z_{f_{b,x}}(\boldsymbol{R}, \boldsymbol{e}) = \widehat{e_{f_{a,x}}}, \varphi^z_{f_{g,x}}(\boldsymbol{R}, \boldsymbol{e}) = \widehat{e_{f_{a,x'}}}, ...$ become scarce resources. But z is fixed, so those manipulations do not work: Suppose a wants to get $s = \varphi^z_{f_{c,x}}(\boldsymbol{R}, \boldsymbol{e})$ for $f_{a,j}$, so it shrinks or enlarges S^{main} by misreporting earliest feasible arrival times of flights that are less important than $f_{a,j}$. Note that $f_{c,x}$ (with $e_{f_{c,x}} \leq s$) ranks higher than $f_{a,j}$ in z. If s is going to $f_{a,j}$, it means $f_{c,x}$ must be getting an earlier slot s' (shrinking S^{main} by giving (type 1 or type 2) slots away will only help but not hurt flights of other airlines. Shrinking S^{main} by removing type 3 slots would only make $f_{a,j}$ worse off (if a removes type 3 slots in some earlier sequences such that flights in earlier sequences are forced to pick slots in the sequence that contains s and $f_{c,x}$ gets a later slot because of that, then $f_{a,j}$ would get an even later slot. If a removes type 3 slots in the sequence that contains s, this would not help. $f_{a,j}$ wants $\varphi^z_{f_{c,x}}(\boldsymbol{R}, \boldsymbol{e})$ instead of $\varphi^z_{f_{a,j}}(\boldsymbol{R}, \boldsymbol{e})$ but it cannot get it under $\varphi^z(\boldsymbol{R}, \boldsymbol{e})$. This implies $\varphi^z_{f_{c,x}}(\boldsymbol{R}, \boldsymbol{e})$ is an earlier slot that cannot be removed by a. If a removes a type 3 slot in a later sequence, nothing will change for $f_{a,j}$). Enlarging S^{main} will not force $f_{c,x}$ to get a later slot too since $f_{a,x}, f_{a,x'}, ...$ (with $\widehat{e_{f_{a,x}}, e_{f_{a,x'}}}, ...$ reported) will not compete with $f_{b,x}, f_{g,x}, ...$ before $f_{a,j}$ picks a slot); if s' is going to $f_{c,x}$, it means a flight $f_{d,x}$, which ranks higher than $f_{c,x}$ and picks s' in $\varphi^z(\boldsymbol{R},\boldsymbol{e})$, must be getting an earlier slot s'', and so on. But this contradicts the finiteness of $|F|$.

In the main stage, if a wants to manipulate the outcome, it can misreport (i) R_a; (ii) e_a; or (iii) both.

There is no way to change the ranking of a single flight, and changing a flight's earliest feasible arrival time cannot help the flight itself (recall a wants to get s such that $e_{f_{a,j}} \leq s < \varphi^z_{f_{a,j}}(\boldsymbol{R},\boldsymbol{e})$, but by misreporting $e_{f_{a,j}}$, $f_{a,j}$ either gets a slot strictly earlier than $e_{f_{a,j}}$, or a slot weakly later than $\varphi^z_{f_{a,j}}(\boldsymbol{R},\boldsymbol{e})$). Therefore, a must use a subset of its flights to help another subset.

Note that there is no way to use less important flights to help a_j as they always pick later than a_j. Eventually, there is only 1 channel to improve a's outcome: use flights to help flights that are less important than them; that is, use $f_{a,i}, f_{a,i}, ... \in F_{a,I}$ to help $f_{a,j}, f_{a,j'}, ... \in F_{a,J}$, where $f_{a,i}R_af_{a,i'}...R_af_{a,j}R_af_{a,j'}....$

(i)$'$ If a misreports R_a, in $\varphi^z(\widehat{\boldsymbol{R}_a,\boldsymbol{e}_a},(\boldsymbol{R},\boldsymbol{e})_{-a})$, $f_{a,i}$ will pick later than itself in $\varphi^z(\boldsymbol{R},\boldsymbol{e})$. Let $f_{a,x}$ be the flight that takes $f_{a,i}$'s position. It is without loss of generality to assume $e_{f_{a,i}} \neq e_{f_{a,x}}$. Otherwise, we can exclude $f_{a,i}$ from $F_{a,I}$ as it effectively picks a slot for itself (a can always self-optimize at the end).

(ii)$'$ If a misreports e_a, $f_{a,i}$ will pick some slot for a less important flight and some less important flight will pick a slot for it, then a can self-optimize given the slots it gets by misreporting.

(iii)$'$ If a misreports both, one of the two cases above must happen.

Note that in all circumstances, $\varphi^z_{a,i}(\boldsymbol{R},\boldsymbol{e})$ is the best possible slot $f_{a,i}$ can get given z. In (i)$'$, (ii)$'$, and (iii)$'$, a will pick a slot for $f_{a,i}$ strictly later than it would in $\varphi^z(\boldsymbol{R},\boldsymbol{e})$. In the main stage, each slot is demanded by more than 1 airline. Since z is arbitrary, that means there exists some realization such that $\varphi^z_{a,i}(\boldsymbol{R},\boldsymbol{e})$ would be picked by some other airline.

Sum up, the probability for $f_{a,i}$ to get an earlier slot is 0, while the probability of getting a later slot is bounded away from 0. Let $\mathcal{L}(\phi(\boldsymbol{R},\boldsymbol{e}))$ be the schedule lottery induced by MTC if $(\boldsymbol{R},\boldsymbol{e})$ is being reported. We have $d_{f_{a,i}}(\mathcal{L}(\phi(\widehat{\boldsymbol{R}_a,\boldsymbol{e}_a},(\boldsymbol{R},\boldsymbol{e})_{-a}))) - d_{f_{a,i}}(\mathcal{L}(\phi(\boldsymbol{R},\boldsymbol{e}))) > 0$, and this means a prefers to report R_a and e_a truthfully.

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

Proof of Theorem 4-1 and 4-2: (4-1) Suppose airline a freezes a canceled flight $f \in F^o_a \setminus F_a$ in a slot $s \in S_a$. $I = (S, A, F^o, R, e, \Phi(A,\hat{S}))$ and $I' = (S \setminus \{s\}, A, F^o \setminus \{f\}, R, e, \Phi(A,\hat{S}))$. First suppose s cannot be used to trade (not demanded by another airline, so it can be in S^{0-0} only if a uses it).

Denote the probability of $a(i)$ in instance I is drawn before the t-th flight is being drawn by $q^i_{I,t}$. Given $a(i-1)$ is drawn before, when the t-th flight is being drawn but $a(i)$ has not been drawn yet, the probability for $a(i)$ to get this position t in z in instance I is denoted by $p^i_{I,t}$. We list $p^i_{I,t}$'s for $a(1)$'s and $a(2)$'s in the table below.

z	1	2	3	$\cdots$	$\lvert F^o\rvert - \lvert F^o_a\rvert$	$\lvert F^o\rvert - \lvert F^o_a\rvert + 1$	$\lvert F^o\rvert - \lvert F_a\rvert + 2$
$a(1)$ in I	$\frac{\lvert F^o_a\rvert}{\lvert F^o\rvert}$	$\frac{\lvert F^o_a\rvert}{\lvert F^o\rvert-1}$	$\frac{\lvert F^o_a\rvert}{\lvert F^o\rvert-2}$	$\cdots$	$\frac{\lvert F^o_a\rvert}{\lvert F^o_a\rvert+1}$	1	0
$a(1)$ in I'	$\frac{\lvert F^o_a\rvert-1}{\lvert F^o\rvert-1}$	$\frac{\lvert F^o_a\rvert-1}{\lvert F^o\rvert-2}$	$\frac{\lvert F^o_a\rvert-1}{\lvert F^o\rvert-3}$	$\cdots$	$\frac{\lvert F^o_a\rvert-1}{\lvert F^o_a\rvert}$	1	0
$a(2)$ in I	0	$\frac{\lvert F^o_a\rvert-1}{\lvert F^o\rvert-1}$	$\frac{\lvert F^o_a\rvert-1}{\lvert F^o\rvert-2}$	$\cdots$	$\frac{\lvert F^o_a\rvert-1}{\lvert F^o_a\rvert+1}$	$\frac{\lvert F^o_a\rvert-1}{\lvert F^o_a\rvert}$	1
$a(2)$ in I'	0	$\frac{\lvert F^o_a\rvert-2}{\lvert F^o\rvert-2}$	$\frac{\lvert F^o_a\rvert-2}{\lvert F^o\rvert-3}$	$\cdots$	$\frac{\lvert F^o_a\rvert-2}{\lvert F^o_a\rvert}$	$\frac{\lvert F^o_a\rvert-2}{\lvert F^o_a\rvert-1}$	1

Position $|F^o|-|F+a|+1$ is the worst position $a(1)$ can get in *both* instances. Each initial ordering z in MTC with $a(1)$ in position $|F^o|-|F^o_a|+1$ induces a landing schedule, and there are $(|F^o|-|F^o_a|)!$ of them. Let d^1_{max} denote the expected delay for $a(1)$ in these landing schedules (in MTC, the is the average delay for $a(1)$ in these landing schedule because z has a uniform distribution). Similarly, let $d^1_{|F^o|-|F^o_a|}$ denote the expected delay for $a(1)$ in landing schedules that are induced by initial ordering z's with $a(1)$ in position $|F^o|-|F^o_a|$. In general, let d^i_t denote the expected delay for flight $a(i)$ in landing schedules that are induced by initial ordering z's with $a(i)$ in position t. Since getting an earlier position does not hurt $a(i)$, $d^i_t \le d^i_{t+1}$ for all i and t (note that $i \le t \le |F^o|-|F+a|+i$. In general, $d^i_{max} \equiv d^i_{|F^o|-|F^o_a|+i}$. For brevity, we omit the domains of t's below).

The expected delay for $a(1)$ in instance I when the $|F^o|-|F^o_a|$-th flight is being drawn but $a(1)$ has not been drawn yet is

$$D^{1,I}_{|F^o|-|F^o_a|} = \frac{|F^o_a|}{|F^o_a|+1} \cdot d^1_{|F^o|-|F^o_a|} + (1-\frac{|F^o_a|}{|F^o_a|+1}) \cdot d^1_{max},$$

and the expected delay for $a(1)$ in instance I when the $|F^o|-|F^o_a|-1$-th flight is being drawn but $a(1)$ has not been drawn yet is

$$D^{1,I}_{|F^o|-|F^o_a|-1} = \frac{|F^o_a|}{|F^o_a|+2} \cdot d^1_{|F^o|-|F^o_a|-1} + (1-\frac{|F^o_a|}{|F^o_a|+2}) \cdot D^{1,I}_{|F^o|-|F^o_a|}.$$

In general, $D^{1,I}_t = p^1_{I,t} \cdot d^1_t + (1-p^1_{I,t})D^{1,I}_{t+1}$. We can calculate $D^{1,I}_t$'s recursively and eventually get

$$D^{1,I}_1 = \frac{|F^o_a|}{|F^o|} \cdot d^1_1 + (1-\frac{|F^o_a|}{|F^o|}) \cdot D^{1,I}_2.$$

For $a(1)$ in I',

$$D^{1,I'}_{|F^o|-|F^o_a|} = \frac{|F^o_a|-1}{|F^o_a|} \cdot d^1_{|F^o|-|F^o_a|} + (1-\frac{|F^o_a|-1}{|F^o_a|}) \cdot d^1_{max}.$$

Because $d^1_{|F^o|-|F^o_a|} \leq d^1_{max}$ and $\frac{|F^o_a|}{|F^o_a|+1} > \frac{|F^o_a|-1}{|F^o_a|}$, we have $D^{1,I}_{|F^o|-|F^o_a|} \leq D^{1,I'}_{|F^o|-|F^o_a|}$. Since $p^1_{I,t} > p^1_{I',t}$ for all t, we have $D^{1,I}_t \leq D^{1,I'}_t$ for all t. ($D^{1,I}_1$ is indeed the expected delay for $a(1)$ in $\phi^I(R,e)$, that is, $d_{a(1)}(\phi^I(R,e))$.)

In general, the expected delay for $a(i)$ in instance I when the t-th flight is being drawn but $a(i)$ has not been drawn yet is $D^{i,I}_t$ ($D^{i,I}_t$ for $t \leq i$ are the same as $D^{i,I}_i$, and $D^{i,I}_1 = D^{i,I}_i$ is the expected delay for $a(i)$ in $\phi^I(R,e)$, that is, $d_{a(i)}(\phi^I(R,e))$). If $D^{1,I}_1 = D^{1,I'}_1$, then we need to show $D^{2,I}_1 \leq D^{2,I'}_1$.

Recall that $q^i_{I,t}$ is the probability of $a(i)$ in instance I is drawn before the t-th flight is being drawn. Observe that $q^1_{I,t}(p^1_{I,1}, p^1_{I,2}, ..., p^1_{I,t-1})$ is monotone in each argument. Since $p^1_{I,t} > p^1_{I',t}$ for all t, we have $q^1_{I,t} \geq q^1_{I',t}$ for all t (when the t-th flight is being drawn but $a(1)$ has not been drawn yet, the probability that $a(1)$ is not drawn is $(1-p^1_{I,t})$ for all t, and so $q^1_{I,t} = 1 - \prod_{t-1}(1-p^1_{I,t})$). In other words, $q^1_{I,t}$ first order stochastic dominates $q^1_{I',t}$. Observe that $q^2_{I,t}(q^1_{I,t-1}, p^2_{I,2}, ..., p^2_{I,t-1})$ is monotone in each argument. Since $p^2_{I,t} > p^2_{I',t}$ for all t, we have $q^2_{I,t} \geq q^2_{I',t}$ for all t (when the t-th flight is being drawn but $a(2)$ has not been drawn yet and $a(1)$ is drawn before (this has probability $q^1_{I,t}$), the probability that $a(2)$ is not drawn is $(1-q^1_{I,t}\cdot p^2_{I,t})$ for all t; therefore, $q^2_{I,t} = 1-\prod_{t-1}(1-q^1_{I,t}\cdot p^2_{I,t})$). In general, $q^i_{I,t}(q^{i-1}_{I,t-1}, p^i_{I,i}, ..., p^i_{I,t-1})$ is monotone in each argument. Since $p^i_{I,t} > p^i_{I',t}$ for all i and t, we have $q^i_{I,t} \geq q^i_{I',t}$ for all i and t ($q^i_{I,t} = 1-\prod_{t-1}(1-q^{i-1}_{I,t}\cdot p^i_{I,t})$). Therefore, we have $q^{i-1}_{I,t}\cdot p^i_{I,t} \geq q^{i-1}_{I',t}\cdot p^i_{I',t}$ for all i and t ($q^0_{I,t} = 1$ for all t). Note that

$$D^{i,I}_t = q^{i-1}_t \cdot p^i_t \cdot d^i_t + (1-q^{i-1}_t \cdot p^i_t)D^{i,I}_{t+1}$$

($D^{1,I}_t = p^i_t \cdot d^i_t + (1-p^i_t)D^{1,I}_{t+1}$ because $q^0_{I,t} = 1$ for all t). By the same recursive argument above, we have $D^{2,I}_1 \leq D^{2,I'}_1$ and in general, $D^{i,I}_1 \leq D^{i,I}_1$ for all $a(i)$.

Self optimization after obtaining $\underbrace{\phi^{I'}_a(R,e)}$: If s cannot be used by any flight of a, we are done. Otherwise, a can give s to some of its flights and obtain some self-optimized landing schedule for a $\Pi_a^{\overbrace{\phi^{I'}_a(R,e)}\cup\{s\}}$. Let $a(j)$ be the most important flight that might be assigned s in all $\Pi_a^{\overbrace{\phi^{I'}_a(R,e)}\cup\{s\}}$'s.

Case 1: If $a(j)$ obtains some scarce resource s' in MTC. The fact that s, which is not demanded by another airline, might be given to $a(j)$ implies that s is earlier than the sequence of slots that contains s' (suppose s is later than the sequence. This contradicts the way we pick $a(j)$ as $a(j)$ always gets an earlier slot in the sequence). The expected delay for $a(j)$ is therefore constant as it always gets s. Still, we have $D^{i,I}_1 \leq D^{i,I'}_1$ for each $a(i)$—in addition to all of its flights pick weakly later, each remaining flight that is less important than $a(j)$ picks strictly later because $a(j+1)$ can use $a(j)$'s position in z if s is in the instance, and so on. So a is not better off in this case (the best thing a can do is to let $a(j)$ pick a slot for

$a(j+1)$, $a(j+1)$ pick a slot for $a(j+2)$, etc. But this is the same as having s put in the instance).

Case 2: If $a(j)$ obtains some non-scarce resource s' in MTC. The fact that s, which is not demanded by another airline, might be given to $a(j)$ implies that s is earlier than s'. The expected delay for $a(j)$ is therefore constant as it always gets s.

(♣) If s' cannot be used by other flights of a or s' can be used by some flights of a but each of these flights obtains a slot better than s' in MTC, we are done because of $D_1^{i,I} \leq D_1^{i,I'}$ for each $a(i)$. Otherwise, let $f \in F_a$ be the most important flight that might be assigned s'. If f obtains some scarce resource s'' in MTC, the expected delay for f is therefore constant as it always gets s'. Still, we have $D_1^{i,I} \leq D_1^{i,I'}$ for each $a(i)$—in addition to all of its flights pick weakly later, each remaining flight that is less important than f picks strictly later (as in Case 1). If f obtains some non-scarce resource s'' in MTC, repeat (♣) with s'' in place of s'. We will eventually have $D_1^{i,I} \leq D_1^{i,I'}$ for each $a(i)$ since a has finitely many flights. So a is not better off in this case.

Now suppose s might be used to trade. If s is used by a itself under all z's, we are done as $D_1^{i,I} \leq D_1^{i,I'}$ for all $a(i)$. Suppose s is traded for some slot under some z. Let $a(k)$ be the most important flight that gets an earlier slot under all such z's. If there is no flight that gets an earlier slot under all z's, we are done as $D_1^{i,I} \leq D_1^{i,I'}$ for all $a(i)$. Otherwise, observe that $D_1^{k,I}$ is lowered as some d_t^i is reduced and thus $D_1^{k,I} < D_1^{k,I'}$ (in this case, it is possible that $D_1^{m,I} > D_1^{m,I'}$ for some $m > k$). So a is not better off in this case.

(4-2) Let $f_{a,1}$ be the most important flight of a. Suppose the earliest feasible available slot for $f_{a,1}$, s, is in S_a and s is a scarce resource. $a(1)$ in I represents $f_{a,1}$. $a(2)$ in I represents some flight of a. $D_1^{2,I}$ is the expected delay for this flight in instance I in MTC. Let $a(1)$ in I' represent this flight in I'. $D_1^{1,I'}$ is the expected delay for this flight in instance I' in MTC. We want to show $D_1^{2,I} \geq D_1^{1,I'}$.

Now d_t^i in I might be different from d_t^i in I'. Let $d_t^{i,I}$ denote the one in I. For $i \geq 2$, let $\tilde{d_t^{i,I}}$ be the expected delay for flight $a(i)$ in I in landing schedules that are induced by initial ordering z's with $a(i)$ in position t conditioning on $a(1)$ gets the first position in z with probability 1 (that is, $q_{I,1}^1 = 1$). Observe that

$$D_t^{2,I} = q_{I,t}^1 \cdot p_{I,t}^2 \cdot d_t^{2,I} + (1 - q_{I,t}^1 \cdot p_{I,t}^2) D_{t+1}^{2,I} \geq p_{I,t}^2 \cdot \tilde{d_t^{2,I}} + (1 - p_{I,t}^2) \tilde{D_{t+1}^{2,I}} = p_{I',t}^1 \cdot d_{t-1}^{1,I'} + (1 - p_{I',t}^1) D_t^{1,I'} = D_t^{1,I'}.$$

For $t \geq i \geq 2$, each $q_{I,t}^i$ is monotone in $q_{I,t-i+1}^1$ and 1 is the optimum for all $q_{I,t-i+1}^1$ (as $q_{I,1}^1 = 1$), but in the second term, each $q_{I,t-i+1}^1$ is weakly less than 1, which would result in weakly larger expected delays for all other flights of a (this gives the weak inequality). $d_{t-1}^{1,I'}$ is effectively the same as $\tilde{d_t^{2,I}}$ for all $t \geq 2$ (this gives the second equality). If $D_1^{2,I} = D_1^{1,I'}$, then we need to show $D_1^{3,I} \geq D_1^{2,I'}$. But the argument is the same for all $a(i)$ in I with $i \geq 2$. ■

Proof of Proposition 4: (All proofs are the same except the one for Theorem 3. We provide the modification to the Proof of Theorem 3 for the modified pre-competition stage (all other parts are the same).)

...In the pre-competition stage, the only possible way for airline a to get a slot that is feasible for $f_{a,j}$ but not being assigned in this stage is to make it a non-scarce resource (type 4 slot) such that it will be assigned to $f_{a,j}$. We show that any slot that can be converted to a type 4 slot by a cannot be s.

Case 1: Suppose s is a type 1 slot (either in the type 1 slot group or a slot group with both type 1 and type 2 slots). a can give s to $f_{a,j}$ only when the type 1 slots at the same time are demanded by a with $f_{a,x}, ..., f_{a,x'}, ...$, where $f_{a,x,...} R_a f_{a,j} R_a f_{a,x'} ....$ and at most $n(s) - |f_{a,x,...} f_{a,j}|$ flights that do not belong to a (call these flight $f_{b,x}, ...$), where $n(s)$ is the number of these type 1 slots and $|f_{a,x,...} f_{a,j}|$ is the number of a's flights that are weakly more important than $f_{a,j}$. Then a can misreport $\widehat{e_{f_{a,x'}}, ...}$ to give s to $f_{a,j}$ ($f_{a,x}, ...$ and $f_{b,x}, ...$ also get these slots). (Note that this procedure (of misreporting $\widehat{e_{f_{a,x'}}, ...}$) makes the type 1 slots satisfy (c-ii). It also makes the next slot group a type 1 slot group if it was a type 2 slot group or a slot group with both type 1 and type 2 slots; moreover, it makes the next slot group a type 4 slot group if it was a type 3 slot group or a slot group with both type 3 and type 4 slots. In other words, this procedure makes the slots in next slot group satisfy (c-i).)

Case 2: Suppose s is a type 2 (either in a type 2 slot group or a slot group with both type 1 and type 2 slots). If not impossible, a can give s to $f_{a,j}$ only by repeating the procedure in Case 1. This requires at each previous iteration, type 1 slots at the same time are demanded by a with $f_{a,y}, ..., f_{a,y'}, ...$, where $f_{a,y,...} R_a f_{a,j} R_a f_{a,y'} ....$ and at most $n(s) - |f_{a,y,...}|$ flights that do not belong to a (call these flight $f_{c,y}, ...$). Then a can misreport $\widehat{e_{f_{a,y'}}, ...}$ so that $f_{a,y}, ...$ and $f_{c,y}, ...$ get these slots. At the end, it requires s to become a type 1 slot that can be given to $f_{a,j}$ as described in Case 1. (If not impossible, a can convert s to a type 3 slot by using the procedure in Case 1 to make s satisfy (c-ii). But there is no way for a to convert it to a type 4 slot after that.)

Case 3: Suppose s is a type 3 slot (either in a type 3 slot group or a slot group with both type 3 and type 4 slots). a can give s to $f_{a,j}$ by change $e_{f_{a,j}}$ to s ($> e_{f_{a,j}}$) and have this slot assigned to $f_{a,j}$ (if there is more than 1 such type 3 slots. Let flights that are assigned type 3 slots in $\hat{\Pi}$ be $f_{a,x}, f_{a,x'}, ...$ a also needs to misreport $e_{f_{a,x}}, \widehat{e_{f_{a,x'}}}, ...$ But this even requires $f_{a,j}$ to be the most important flight among $f_{a,j}, f_{a,x}, f_{a,x'}, ...$ since a is only willing to sacrifice flights that are less important than $f_{a,j}$ to help it).

In Case 1 and 2, there are sufficient slots to accommodate the only competitors $f_{b,x}, ..., f_{c,y}, ...$

and $f_{a,x}, ..., f_{a,y}, ...$ before s is being picked as $f_{b,x}, ..., f_{c,y}, ...$ and $f_{a,x}, ..., f_{a,y}, ...$ will always try to get a slot (weakly) earlier than s and flights $f_{a,x'}, ..., f_{a,y'}, ...$ that are less important than $f_{a,j}$ will not compete with them before $f_{a,j}$ gets a slot in the main stage. In Case 3, $f_{a,j}$ can get a weakly earlier slot in the main stage. In all cases, $f_{a,j}$ can always get a weakly earlier slot in the main stage, so $\varphi^z_{f_{a,j}}(\boldsymbol{R}, \boldsymbol{e}) \leq \varphi^z_{f_{a,j}}(\widehat{\boldsymbol{R}_a, \boldsymbol{e}_a}, (\boldsymbol{R}, \boldsymbol{e})_{-a}) = s$, a contradiction... ■

B Summary of properties

	Compression	TC	DASO	MTC
Preference domain*	-	$e, \Pi^{current}$	e, w**	e, R
Individual rationality	No***	Yes	Yes	Yes
Pareto efficiency	No***	Yes	No	Yes
Core	No	Yes	No	Yes
Strategy-proofness	No***	Yes	No	Yes
Non-manipulable****	No	No	Yes	Yes
Ex post property rights	No	Yes	Yes	Yes
Ex ante property rights	No (because of RBS)			Yes

* Properties defined by preference are different in different preference domains.
** w are weights of flights.
*** Yes in the preference domain of Schummer and Vohra (2013).
****Non-manipulable by postponing flight cancellation.

C Examples for Compression

We use the following example to show Compression is not strategy-proof. This is the same example Schummer and Abizada (2017) used to show Compression is not strategy-proof. We convert weights (in that example) to rankings.

Example 15:

GDP slot	s_1	s_2	s_3	s_4
Flight	-	$f_{a,2}$	$f_{b,1}$	$f_{a,1}$
$\Phi(\cdot, S)$	c	a	b	a
e_f	-	2	1	1
$\widehat{e_f}$	-	1	1	2
R	-	2	1	1
Compression (e)	$f_{b,1}$	$f_{a,2}$	$f_{a,1}$	c
Compression ($\widehat{e}$)	$f_{a,2}$	$f_{a,1}$	$f_{b,1}$	c

If a slot is unusable to its owner, then the next flight in the schedule that can feasibly use it would be assigned this slot in Compression. If a reports e_a, then b obtains s_1 for $f_{b,1}$ and c obtains s_3. Then a obtains s_3 for $f_{a,1}$ and c obtains s_4. But if a reports $\widehat{e_a}$, then a obtains s_1 for $f_{a,2}$ and c obtains s_2. Then a obtains s_2 for $f_{a,1}$ and c obtains s_4. We can see a strictly gain in this case (a can swap slots for $f_{a,1}$ and $f_{a,2}$).

We use the following example to show Compression is manipulable by postponing a flight cancellation. This is the same example Schummer and Vohra (2013) used to show Compression is manipulable via slot destruction. We put rankings on flights.

Example 16:

GDP slot	s_1	s_2	s_3	s_4	s_5	s_6	s_7
Flight	$f_{a,(1)}$	$f_{b,(1)}$	$f_{a,(2)}$	$f_{c,1}$	$f_{b,1}$	$f_{a,1}$	$f_{b,2}$
$\Phi(\cdot, S)$	a	b	a	c	b	a	b
e_f	-	-	-	2	4	4	1
R	-	-	-	1	2	1	1
Compression	$f_{b,2}$	$f_{c,1}$	a	$f_{b,1}$	$f_{a,1}$	b	a
Compression′	$f_{a,(1)}$	$f_{b,2}$	$f_{c,1}$	$f_{a,1}$	$f_{b,1}$	a	b

$f_{a,(1)}$, $f_{b,(1)}$ and $f_{a,(2)}$ are canceled flights. In the first step of Compression, b obtains s_1 for $f_{b,2}$ while a obtains s_7. Then b obtains s_4 for $f_{b,1}$ while c obtains s_2 for $f_{c,1}$. Finally, a obtains s_5 for $f_{a,1}$ while b obtains s_6. But if a freezes $f_{a,(1)}$ in s_1. In the first step of Compression′, $f_{b,2}$ is moved to s_2. Then c obtains s_3 for $f_{c,1}$ while a obtains f_4 for $f_{a,1}$. Finally, b obtains s_5 for $f_{b,1}$. It is easy to see a gains by having s_4.