Mechanism Design for the Online Allocation of Items without Monetary Payments

Behnam Jalilzadeh*, Léon Planken, and Mathijs de Weerdt

Delft University of Technology, P.O. Box 5031, 2600 GA, Delft, the Netherlands
B.Jalilzadeh@student.tudelft.nl, L.R.Planken@tudelft.nl,
M.M.deWeerdt@tudelft.nl

Abstract. We consider online mechanism design without money, where agents are allowed to trade items with other agents, in an attempt to improve their own allocation. In an off-line context, this problem is known as the House Allocation Problem (HAP). We extend HAP to an online problem and call it the Online House Allocation Problem (OHAP). In OHAP, agents can choose when to arrive and depart over time and are allowed to be indifferent between items. Subsequently, we present our Agent Shifting Algorithm (ASA) for OHAP. A mechanism that uses ASA as its allocation rule is shown to be strategy-proof, individually rational and Pareto optimal. Moreover, we argue that any mechanism that obtains an outcome in OHAP that cannot be obtained by using ASA fails to be strategy-proof or is not Pareto optimal.

1 Introduction

Consider a setting with non-identical items where every agent desires a single item and can choose when to enter and leave the system. Furthermore, presume that each agent is able to set up a preference order for all items. Now, a mechanism is able to compute a matching between agents and items by allocating items to newly arrived agents and by suggesting possible swaps of items between two or more agents. We are only interested in mechanisms that are individually rational (i.e. always encourage agents to participate in a game induced by the mechanism), Pareto optimal (i.e. always obtain matchings where it is not possible to make any agent strictly better off without at least one agent becoming strictly worse off) and strategy-proof (i.e. truthfulness is a dominant strategy for all agents).

The motivation for an analysis of this setting stems from a real-life problem concerning barges that need to use locks. The process of regulating the water level inside a lock takes several minutes and a lock has room for a limited number of barges. This can lead to the formation of queues near locks, where barges in the queues are awaiting their turn for lockage. Furthermore, since barges are granted access to a lock in a first-come first-served fashion, they tend to sail at high speeds in order to arrive at a lock before other barges so as to reduce their

* This paper is an excerpt from the first author's Master's thesis [3].

waiting time to a minimum. This, in turn, leads to inefficient fuel consumption by barges. A possible approach to solving this problem is to design an online mechanism that allocates time slots to barges for lockage.

2 Prior Work and Our Contributions

The situation we portrayed in the introduction resembles a problem which was studied by Shapley and Scarf [9]. They performed pioneering work in this field, and called this problem the House Allocation Problem (HAP). This original work was concerned with a *static* setting without monetary payments, where there is a fixed set of n agents and every agent is the tenant of a unique house and has a preference order over the n houses. The objective of every agent is to swap houses with other agents to acquire the best possible house. Other authors have devoted themselves to this problem in a static context, as well [1, 5–7].

In recent years, however, there has been more focus on HAP in a *dynamic* setting. Bloch and Cantala [2] characterise a dynamic setting with the aid of Markov chains. However, they do not take issues of strategy-proofness into account. Roth et al. [8] use a random Poisson process to model a kidney exchange setting where donors (i.e. the items) and recipients (i.e. the agents) arrive and depart in pairs over time and can trade with other pairs of donors and recipients. Kurino [4] considers HAP in a dynamic setting where a finite number of agents can arrive and depart per time step. Although [8] and [4] bear a resemblance to our work, there are also some differences. Firstly, Roth et al. consider a setting where newly arrived agents have an initial endowment. Secondly, both Roth et al. and Kurino restrict their attention to agents with strict preference orders. Finally, although both deal with strategy-proofness with respect to possible misreports of preferences, they do not consider the temporal aspect of agents being able to postpone reports of their preference orders to the mechanism. Instead, the arrival and departure of agents is treated as an exogenous process, so agents can only influence their allocation by the preference order that they report. In our work, we take into account that agents themselves can choose when they enter and leave the system.

To the best of our knowledge, our study is the first to consider a general framework without monetary payments where agents are allowed to be indifferent between items and to determine when they enter and leave the system (i.e. request or relinquish an allocation). Henceforth, we refer to this problem as the Online House Allocation Problem (OHAP). OHAP covers a wide range of applications, e.g. customers who desire a hotel room and have the option to cancel a reservation, patients who require a time slot for medical care from a general practitioner, or the aforementioned scheduling of barges.

In this paper, we present the Agent Shifting Algorithm (ASA), which is an online variant of the Top Trading Cycle Algorithm used for HAP [9]. A key issue in OHAP is that we want to reallocate items that are owned by agents, rather than by the mechanism. This matter is taken into account in ASA. We demonstrate that a mechanism that uses ASA as its allocation rule always ob-

tains a Pareto optimal solution. More importantly, we prove that if another mechanism M for OHAP obtains an outcome that cannot be obtained using ASA, then either M is not strategy-proof or the outcome obtained by M is not Pareto optimal. Finally, our initial simulations of ASA versus a first-come, first-served (FCFS) mechanism reveal promising results for future research.

3 Preliminaries

In our model of OHAP, we consider time as an infinite series of discrete time steps $T = \{0, 1, 2, \ldots\}$. In each time step, either an agent arrival can occur or an agent departure, but not both.[1] An agent i has *private* type (a_i, d_i, p_i), where $a_i, d_i \in T$ represent the arrival and departure time of agent i, respectively, and $a_i < d_i$. The preference order of agent i over all items is denoted by p_i. Each agent i may report its preference order at time $t \geq a_i$. Thus, we assume no false early arrival reports.[2]

By $h \succ_i h'$ we denote the fact that an agent i *strictly* prefers item h to h'. Similarly, $h \sim_i h'$ means that i is *indifferent* between (i.e. has no preference for) h and h'.[3] Each agent requires precisely one item and is able to constitute a preference order for all items. Receiving an allocation of an item is strictly preferred by all agents to not owning any item at all.

An instance of OHAP at time t consists of:

- A set of agents $A = \{1, 2, \ldots, m\}$ that are present at time t, so $\forall i \in A : a_i \leq t \leq d_i$.
- A set of items $H = \{h_1, h_2, \ldots, h_n\}$, with $n \geq m$. Thus at any time step t, an allocation of an item to every agent is always possible.
- A preference order p_i over H for each agent i.

A solution to an instance of OHAP at time t is then a matching μ of items to agents, such that every agent is matched to a single item and all these items are matched to exactly one agent.

4 Possible Mechanisms for OHAP

In this section, we examine what the allocation rule of a mechanism M for OHAP should look like. We stated earlier that in OHAP, reallocations of items are

[1] We justify this assumption by pointing out that we can decrease the size of a time step to the point where simultaneous agent arrivals and departures do not occur in practice.

[2] The assumption of no false early arrival reports is common in online mechanism design, if it is possible to verify an agent's presence. Agents are able to hide their presence by postponing a report of their preferences, but they cannot establish a "phantom" presence by reporting their preferences before their true arrival time.

[3] $h \succ h'$ if and only if $h \succeq h'$ and not $h' \succeq h$. The binary relation $\succ$ is irreflexive, transitive and asymmetric. Furthermore, $h \sim h'$ if and only if $h \succeq h'$ and $h' \succeq h$. The binary relation $\sim$ is reflexive, transitive and symmetric.

possible. We formally define a reallocation as a *chain of shifts*. In the following, let $C = \{1, 2, \ldots, k\} \subseteq A$, in which we assume w.l.o.g. that for all $i \in C$, agent i owns h_i at time t.

Definition 1. *A **chain of shifts** is a non-empty set of agents C, where it is possible for all $i \in C$ to reallocate item h_{i+1} to agent i, which must not strictly prefer its previous allocation h_i to h_{i+1}, and where h_{k+1} is some **previously unallocated** item. The chain of shifts thus frees up item h_1.*

Note the requirement that in a chain of shifts no agent i may strictly prefer its previous allocation to the new one, which would immediately imply that the new matching is not Pareto optimal. Also, note how a chain of shifts can transform into a "trading cycle". To see this, assign a *dummy agent* to every free item. Furthermore, denote by $\hat{h}$ a virtual item, which we use to represent an "empty allocation". Every dummy agent d is indifferent between all items, i.e. $p_d = \{\hat{h} \sim h_1 \sim \cdots \sim h_n\}$. Suppose that a newly arrived agent i desires some item h_1, which can be allocated to i through a chain of shifts C. We first allocate the virtual item $\hat{h}$ to i. Subsequently, we set up a chain of shifts as described in Definition 1 to free up item h_1 and allocate agent k to some free item h_{k+1}. Let d denote the dummy agent that was assigned to h_{k+1}. We can shift d to $\hat{h}$ and i to h_1. These agents now form a "trading cycle" as in the algorithm used for offline HAP [9].

We define agents that prevent a chain of shifts from existing as *blocking agents* and call these chains *blocked chains*.

Definition 2. *An agent i is called a **blocking agent** if removing i results in a chain of shifts. The chains of shifts that the presence of i prevents from emerging are called **blocked chains**.*

Possible chains of shifts are closely related to the Pareto optimality of mechanisms for OHAP. The following proposition describes a prerequisite for an allocation rule in OHAP if our goal is to always obtain a matching that is Pareto optimal.

Proposition 1. *Upon the arrival of an agent i, any Pareto optimal mechanism must allocate to i the best possible item h that is either directly available or can be made available through a chain of shifts.*

Proof. If h is directly available, then it is trivial that i must receive an allocation of h, in order to obtain a Pareto optimal matching.

In case h is not directly available, but can be made available through a chain of shifts, then the proposition holds due to the very definition of a chain of shifts (see Definition 1): agent i can be made strictly better off, without making any other agent worse off. Thus, an allocation of h to i is necessary to obtain a Pareto optimal matching. □

Now that we have an idea of how to deal with newly arrived agents in OHAP, we shift our attention to agent departures. Whenever an agent i chooses to

depart, i relinquishes its current item and leaves the system. A mechanism for OHAP has to take into consideration that i could have been a blocking agent for one (or possibly several) blocked chain(s). As a consequence, a blocked chain can transform into a chain of shifts C that frees up some item. If there is an agent j present in the system that can become strictly better off through C, then a mechanism should perform this allocation in order to preserve a Pareto optimal matching.

Proposition 2. *Upon the departure of an agent i, any Pareto optimal mechanism should check if i was a blocking agent. If so, and the allocation to an agent j can now strictly be improved by a chain of shifts, this should be done.*

Proof. The proof resembles that of Proposition 1. If there is an agent j that can strictly improve its allocation, then we must perform this allocation as long as no other agent becomes worse off. This is exactly the case, because an allocation to j is made possible through a chain of shifts, so that we are sure that all agents in the chain weakly prefer their new item to their previous one. □

We present one final observation with regard to OHAP, before we discuss our online allocation algorithm for OHAP in the next section. In OHAP, an agent can only improve its allocation if some agent relinquishes its item at some point in time.

Proposition 3. *When an item h is allocated to an agent i at time step t, no Pareto optimal mechanism can allocate an item $h' \succ h$ to i unless some agent j cancels its reservation at time step $t' > t$.*

Proof. If i cannot receive an allocation of h' at time step t, this means that there is a blocking agent obstructing a chain of shifts that makes h' available to i. Since an agent arrival can never transform a blocked chain into a chain of shifts, the only possibility for i is that at some time step t' an agent j relinquishes its time slot. This, in turn, can possibly cause a chain of shifts that frees up h', which can then be allocated to i. □

Looking back at this section, we have touched upon several features that are intrinsic to OHAP. The characterisations we examined in this section aid us to construct a mechanism for OHAP in the next section.

5 The Agent Shifting Algorithm

With the propositions of the previous section borne in mind, we present our online algorithm ASA, which can be used to allocate items to agents in OHAP. This algorithm can be summarised as follows. If an agent i arrives, record its arrival time[4] and allocate to that agent the best possible item in accordance with i's reported preference order, whilst maintaining a Pareto optimal matching. In

[4] This is necessary to ensure time strategy-proofness, which we discuss in Theorem 2.

case an agent relinquishes its item, check which reallocations to the other agents can be performed, due to the emergence of one or more chains of shifts. These reallocations are performed in the order of the recorded arrival time of agents, i.e. if an agent i reported its type earlier than another agent j, possible reallocations favourable for i are checked first before the same is done for j. Otherwise, if there is no agent arrival or departure, no operation is necessary.[5]

We present the online algorithm ASA for OHAP shortly, but first we need to introduce two definitions. Let $P = \{h_1 \succ h_2 \succ \cdots \succ h_n\}$ represent a preference order. Then we say that h_i has *index* i.

Definition 3. *Two preference orders p_1 and p_2 are equivalent, written as $p_1 \equiv p_2$, if they reflect the exact same preference for an agent.*

Definition 4. *Given a preference ordering $P = \{h_1 \succ h_2 \succ \cdots \succ h_n\}$, the **rank** of an item $h_k \in P$ is equal to the highest possible index that h_k can have in a preference ordering P', where $P' \equiv P$.*

Example 1. In $P = \{h_1 \succ h_2 \sim h_3 \succ h_4\}$ the rank of h_3 is 2, because we can position h_3 at index 2, which leads to $P' = \{h_1 \succ h_3 \sim h_2 \succ h_4\} \equiv P$. Similarly, the rank of h_4 is 4, because this is the highest index at which we can position h_4. □

The pseudo-code of ASA is presented below. It consists of a function Agent Shifting Algorithm (ASA) that uses the recursive functions arrival and departure to deal with agent arrivals and departures, respectively. Note that both recursive functions are guaranteed to eventually terminate: in the case of arrival, this follows from our assumption that there are more items than agents, whereas in the case of departure, it is implied by the fact that the number of agents is finite, and no agent j will appear in the set S after receiving a new allocation (within a single time step t).

In the remainder if this paper, we denote by ASM the mechanism that uses ASA as its allocation rule. We begin our analysis of ASM by demonstrating that it is strategy-proof. Since ASM is an online mechanism, it is necessary to verify that it is preference strategy-proof (i.e. it is a dominant strategy for every agent to truthfully report its preferences), as well as time strategy-proof (i.e. it is a dominant strategy for every agent i to report its preferences at time $t = a_i$). We first show that no agent can obtain a better outcome by misrepresenting its preference order.

Theorem 1. *The mechanism ASM is preference strategy-proof.*

[5] If it is desirable that the possible outputs of the algorithm over time encompass all possible sequences of Pareto optimal matchings, an arbitrary number of chains of shifts can be performed as a final step; this is the only step if there is no arrival or departure. This can be useful, if e.g. the mechanism has a preference for some matchings to others.

Function Agent Shifting Algorithm

Input: agents $\{1, \ldots, m\}$, items $\{h_1, \ldots h_n\}$, preference orders $\{p_1, \ldots p_m\}$, time step t
Output: matching μ of agents to items
if *some agent i arrives at t* **then**
  Label i with timestamp t `// the time i reported its type`
  arrival(i, p_i)
else if *some agent i departs at t* **then**
  Remove i
  if *i was a blocking agent* **then**
    $S \leftarrow$ {agents that can become strictly better off through a chain of shifts}
    departure(S)
  end
end

Function arrival

Input: newly arrived agent i, preference order p_i of i
$M \leftarrow$ {items in p_i with rank 1 which are free or can be allocated after a chain of shifts}
if $M = \varnothing$ **then**
  arrival(i, $p_i \setminus$ {all items in p_i with rank 1}) `// recursive call`
else
  Select an arbitrary $h \in M$
  Perform the chain of shifts that frees up h (if necessary)
  Allocate h to i
end

Function departure

Input: set S of agents that can be made strictly better off through a chain of shifts
if $S \neq \varnothing$ **then**
  $j \leftarrow$ agent that has the earliest timestamp among all agents in S
  $h \leftarrow$ arbitrary lowest-ranked item that can be allocated to j through a chain of shifts
  Perform the chain of shifts that allocates h to j
  if *j was a blocking agent* **then**
    $S' \leftarrow$ {agents that can become strictly better off through a chain of shifts}
    departure(S') `// recursive call`
  end
end

Proof. Under ASA, upon arrival an agent i receives the best possible item that is either directly available or can be made available through a chain of shifts based on its reported preference order. Hereafter, i's allocation can never deteriorate. Furthermore, if in the future the allocation of i can be improved due to an agent cancellation, i receives the best possible item that can be made available through a chain of shifts, based on its reported preference order. It follows that for every agent i truthfully reporting its preference order is a dominant strategy. Hence, ASM is preference strategy-proof. □

In order to show that ASM is time strategy-proof, we point out that no agent can obtain a better allocation by postponing the report of its preference order.

Theorem 2. *The mechanism ASM is time strategy-proof.*

Proof. Recall that by assumption false early arrival reports are not allowed in OHAP. Therefore, we merely need to prove that no agent can strictly gain by postponing the report of its type.

Let h and h' represent the allocation of an agent i at some time step, if i reports its type at time $t = a_i$ and $t' > a_i$, respectively. It immediately follows that $h \succeq_i h'$ at time step t'. Furthermore, if agents can ever improve their allocations due to an agent departure, the order in which these reallocations are performed is based on the timestamps of these agents. Therefore, it holds that $h \succeq_i h'$ at every time step t. Hence, it is a dominant strategy for every agent i to report its type at arrival time and so ASM is time strategy-proof. □

It remains to be shown that ASM is individually rational so that we are sure that agents are not reluctant to participate in OHAP.

Theorem 3. *The mechanism ASM is individually rational.*

Proof. The proof is rather trivial. Agents that are yet to report their type to ASM, do not have an initial endowment. After reporting their types, agents are certain to receive an allocation of some item. Because owning an item is strictly preferred by all agents to not owning any item at all, the ASM mechanism is individually rational. □

In this section we presented our mechanism ASM that uses ASA to allocate items to agents arriving over time, and performs swaps of allocations between agents. We showed that ASM is suitable to be employed as an allocation mechanism, because it adheres to the prerequisites of online strategy-proofness and individual rationality. In the next section, we put forward that matchings obtained by ASM are always Pareto optimal. Furthermore, we study how ASM relates to other possible mechanisms that can be used for OHAP.

6 Comparing ASM with Other Mechanisms

In this section, we explain why a mechanism that uses ASA as its allocation rule adheres to the prerequisite of Pareto optimality. Subsequently, we present

the main theorem of this paper. This theorem tells us that ASM is the "best" mechanism we can use for OHAP, because if another mechanism M obtains a matching that we cannot obtain by using ASM, then either M is not strategy-proof, or M is not Pareto optimal.

Theorem 4. *ASM always returns a Pareto optimal matching.*

Proof. Let μ_t denote a matching obtained by ASM at time step t and let μ_1 represent the first matching obtained by ASM, which consists of a single agent. Note that this is w.l.o.g. since we can choose $t = 1$ to be the time step of the first agent arrival into the system. We use a proof by induction to show that every μ_t computed by ASM is Pareto optimal.

(i) **Basis:** It holds that μ_1 is Pareto optimal.
(ii) **Inductive step:** If $\mu_{t'}$ is Pareto optimal, then $\mu_{t'+1}$ is Pareto optimal.

Proof of (i): If there are no agents in the system, upon arrival of the first agent i, ASM always gives i one of its favourite items. Hence, μ_1 is Pareto optimal.

Proof of (ii): Assume $\mu_{t'}$ is Pareto optimal. We consider the three possibilities at time step $t' + 1$, namely either nothing happens, or there is an agent arrival or there is an agent cancellation.

- If nothing happens at time step $t'+1$, any permutation of matching $\mu_{t'}$, which is obtained through a number of chains of shifts, is still Pareto optimal.
- Let $S_{t'}$ represent the set of agents that are present at time t' and assume that an agent i reports its type to ASM at time step $t'+1$. From Proposition 3 we know that no agent $j \in S_{t'}$ can be allocated an item that it strictly prefers to its current item. Therefore, if i is allocated the best possible item given matching $\mu_{t'}$, the new matching must be Pareto optimal if no agent in $S_{t'}$ is assigned an item which it finds strictly less preferable than its current item. Since this is exactly what happens in ASM, we know that $\mu_{t'+1}$ is Pareto optimal.
- Assume an agent i relinquishes its item at time step $t' + 1$. We distinguish two cases.
  If i was not a blocking agent, then no blocked chain is transformed into a chain of shifts. Consequently, no agent can strictly improve its current allocation. Hence, if ASM outputs any permutation of matching $\mu_{t'}$ excluding agent i, which is obtained through a number of chains of shifts, we are sure that $\mu_{t'+1}$ is Pareto optimal.
  Otherwise i was a blocking agent and a new chain of shifts could possibly lead to a new allocation to some agents that strictly prefer this to their current item. It is sufficient to perform such an allocation to only one agent j, since a chain of shifts only frees up a single item (see Definition 1). Then, if j was not a blocking agent, we are sure that the new matching obtained by ASM is Pareto optimal. If j itself was a blocking agent, ASM repeats

this process until we reach a non-blocking agent.[6] The obtained matching by ASM will then be Pareto optimal.

Since the base and inductive step have been proved, we conclude that the theorem holds. □

Before we present the main theorem of this paper, we explain why ASM chooses to improve the allocation of an agent with the earliest possible timestamp in case of an agent departure.

Lemma 1. *If an agent j relinquishes its item and a set S of agents can improve their allocation due to the departure of j, any strategy-proof mechanism M must improve the allocation of the agent in S that was the first to arrive among all agents in S.*

Proof. Denote by $V = \{1, 2, \ldots, k\}$ the agents that are able to become strictly better off at some time step, due to the departure of an agent j. Furthermore, let $i \in S$ be the agent that was the first among all agents in S to arrive into the system. To reach a contradiction, assume a strategy-proof mechanism M does not consider to improve the allocation of i first.

If agent i knows that M does not choose the agent in V that was the first to arrive into the system, then i has an incentive to postpone reporting its preference order to M. Since i knows that this strategy might lead to an allocation that is more preferable than an allocation when it reports its preference order at its arrival time, it is not a dominant strategy for i to report at its true arrival time. This contradicts the assumption that M is strategy-proof and therefore the lemma holds. □

With Lemma 1 borne in mind, we present our main theorem.

Theorem 5. *For every time step t, if some mechanism M obtains a matching μ that cannot be obtained by ASM, then either M is not strategy-proof or M is not Pareto optimal.*

Proof. Let μ_t be a matching that is computed by ASM at time t and let μ'_t denote the matching computed by some other mechanism M at time t. Furthermore, let $\Omega_t = \{\mu_t^1, \mu_t^2, \ldots \mu_t^n\}$ denote the set of all possible matchings that can be obtained by ASM at time t. Choose w.l.o.g. $t = 1$ to be the time of the first agent arrival into the system. We use a proof by induction.

(i) **Basis:** Any Pareto optimal strategy-proof mechanism M must choose $\mu'_1 \in \Omega_1$.
(ii) **Inductive step:** If a Pareto optimal strategy-proof mechanism M must choose $\mu'_{t'} \in \Omega_{t'}$, then M must choose $\mu'_{t'+1} \in \Omega_{t'+1}$.

[6] This procedure will not repeat endlessly in practice, because the number of agents will be finite.

Proof of (i): Upon arrival of the first agent i at $t = 1$, ASM allocates to i one of its favourite items. This constitutes the set of possible matchings Ω_1. From Proposition 1 we know that any matching $\mu'_1 \notin \Omega_1$ cannot be Pareto optimal. Hence, any Pareto optimal strategy-proof mechanism M must choose $\mu'_1 \in \Omega_1$ at $t = 1$.
Proof of (ii): Assume that M must choose $\mu'_{t'} \in \Omega_{t'}$. We now consider all possibilities for time step $t' + 1$.

- If there is no agent arrival or cancellation at time step $t'+1$, ASM suggests the same matching as the previous time step, i.e. $\mu_{t'+1} = \mu_{t'}$, or a permutation of $\mu_{t'+1}$ obtained through a chain of shifts. This constitutes the set of possible matchings $\Omega_{t'+1}$. Note that due to Proposition 3, we know that no other matching exists that can strictly improve the allocation of an agent, without some other agent becoming strictly worse off. Therefore, $\Omega_{t'+1}$ contains all the Pareto optimal matchings that can be obtained from $\Omega_{t'}$ and M must choose $\mu'_{t'+1} \in \Omega_{t'+1}$.
- If an agent reports its type at time step $t' + 1$, denote this agent by j. ASM allocates to j one of its favourite items among the ones that are either directly available or can be made available by a chain of shifts. This constitutes the set of possible matchings $\Omega_{t'+1}$. From Proposition 1 we know that any matching $\mu'_{t'+1} \notin \Omega_{t'+1}$ cannot be Pareto optimal. Hence, any Pareto optimal mechanism M must choose $\mu'_{t'+1} \in \Omega_{t'+1}$ at time step $t' + 1$.
- If an agent cancels its reservation at time step $t' + 1$, denote this agent by j. If j was not a blocking agent, the set of possible Pareto optimal matchings $\Omega_{t'+1}$ is equal to $\Omega_{t'}$ (where these matchings exclude agent j). Hence, M must choose a matching $\mu'_{t'+1} \in \Omega_{t'+1}$.
  If j was a blocking agent, let V represent the set of agents that can obtain an allocation that they strictly prefer to their current one, due to the departure of j. ASM performs this allocation for the agent $k \in V$, that was the first to report its type among all agents in V. Any mechanism M that chooses to improve the allocation of another agent in V than j also obtains a Pareto optimal matching, but now M cannot be time strategy-proof due to Lemma 1. Therefore, a Pareto optimal strategy-proof mechanism M must choose $\mu'_{t'+1} \in \Omega_{t'+1}$.

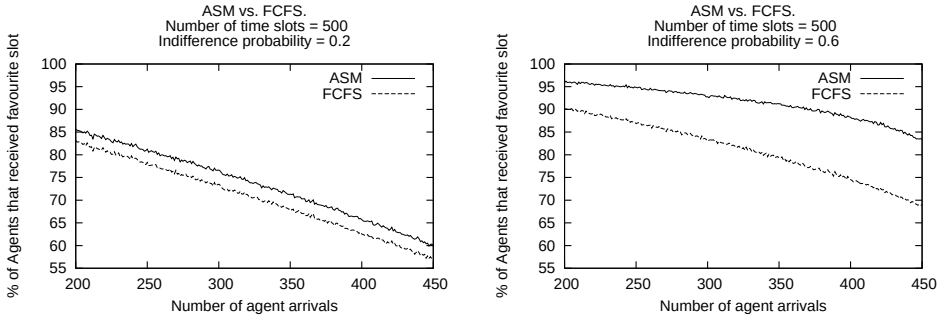
By proving the basis and the inductive step, we have shown that for every t, any Pareto optimal strategy-proof mechanism M must choose $\mu'_t \in \Omega_t$. □

7 Experimental Results

In this section we discuss the experiments we conducted to determine the performance of ASM. Our experiments were geared towards the problem of assigning time slots to barges for lockage, as described earlier. We used the outcomes of an FCFS mechanism as a yardstick against which we measured the outcomes obtained by our mechanism ASM.

In our simulations, we modelled time as discrete time steps of 1 minute. Furthermore, we presumed that lockage takes 10 minutes for every barge, and so every time slot lasts 10 minutes. We examined the efficiency of ASM and FCFS as the number of agent arrivals into the system increases. For the sake of simplicity, we only ran simulations in which there was an agent arrival at every time step and where there were no agent departures from the system. Every agent in the simulation randomly chooses a speed between the 1 (km/h) and 20 (km/h) at which it sails towards the lock. The distance from the starting point of every agent to the lock was set to 10 (km).

We examined the performance of ASM and FCFS where agents have random preference orderings. In reality, preference orderings are very likely not be random. More plausible possibilities, especially when dealing with time slots, are e.g. barges with preference orderings that are monotonically non-increasing in time. We used an indifference parameter to control the probability that an agent is indifferent between two adjacent time slots. For example, if indifference is set to 0.3, we assign rank 0 to the first reachable time slot, given the current time and a barge's speed. Then, with probability 0.3 the second time slot is also assigned rank 0 and with probability 0.7 it is assigned rank 1. Again, with probability 0.3 the third time slot is set to the rank of the previous time slot etc. Hereafter, the ordering can be shuffled to create a (pseudo) random preference ordering. Our obtained results depicted in the following graphs are all averages based on 100 runs.



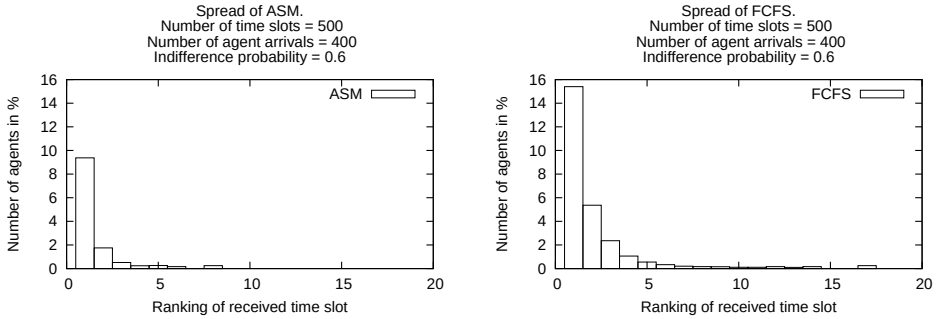
(a) More agents receive their favourite time slot in ASM than in FCFS.

(b) The performance gap between ASM and FCFS becomes larger when we increase the indifference parameter.

Fig. 1: A comparison of the number of agents that received their favourite in ASM and FCFS.

First, we take a look at Figure 1a. The figure shows the result of an experiment where the indifference parameter is set to 0.2 for every agent and where we focus on which percentage of all agents received their favourite time slot. We see that in all cases ASM outperforms FCFS and that both graphs are decreasing

(approximately) linearly in the number of agent arrivals. This is due the fact that the system becomes more "crowded", which leads to fewer time slots being available to newly arrived agents. In Figure 1b, we see that increasing the indifference parameter leads to better performance of ASM in comparison to FCFS. It is not very surprising that the performance gap between ASM and FCFS becomes bigger as the value of the indifference parameter increases, because this makes more and more trades of time slots between agents possible. A remarkable result is that in Figure 1b the performance of both ASM and FCFS seems to degrade approximately exponentially, whereas in Figure 1a the performance of both mechanisms keeps degrading linearly. At the moment, we cannot find an obvious explanation for this observation.



(a) In ASM, the spread of the distribution of time slots over agents is low.

(b) FCFS allocates less favourable time slots to agents that did not receive their favourite time slots, in comparison to ASM.

Fig. 2: A comparison of the agents that did not receive their favourite time slot in ASM and FCFS.

Another important issue is to consider the agents that did not receive their favourite time slot. Did they receive their second or third best choice, or did they receive a worse allocation? This information illustrates the spread of the distribution of time slots over the agents. The results for ASM and FCFS can be seen in Figures 2a and 2b, respectively, for the indifference parameter 0.6. These graphs portray the situation for 400 agent arrivals.

Note that these percentages are not relative to the remaining agents that did not receive their favourite time slot, but to the total number of agents. For example, from Figure 2a it follows that approximately 9.5% of the agents received their second best time slot in ASM. Furthermore, note that the sum of the bars that indicate the spread for ASM is lower than that of FCFS. This is because the percentage of agents that received their favourite time slot in ASM is higher than in FCFS, as can be seen in Figure 1b. By looking at Figure 2a and

2b, we see that ASM allocates "more favourable" time slots to agents, whereas in FCFS the allocations are spread out across time slots that are less favourable.

8 Summary and Future Work

In this paper, we discussed an online extension of the House Allocation Problem (OHAP). We showed that at any time t, there exists a Pareto optimal outcome in OHAP and that such an outcome can be obtained by using our online strategy-proof Agent Shifting Mechanism (ASM). Moreover, we argued that if any other mechanism M obtains an outcome in OHAP, which cannot be obtained by using ASM, then either M is not strategy-proof or M is not Pareto optimal. Our initial experiments with ASM show promising results for the allocation of time slots for lockage to barges.

Future simulations where agents have other preference orders and arrive and depart according to e.g. a Poisson process may shed more light on the performance of ASM. In a setting where items can expire (e.g. time slots) it is realistic to consider preference orders of agents that change over time. We want to know what claims we can make about the strategy-proofness and Pareto optimality of possible mechanisms in such a setting. Finally, it is interesting to consider a model where we make a distinction between different Pareto optimal matchings. For instance, in the problem of scheduling barges for lockage, a lock can have room for several barges and not all barges are of the same size. In this case, the mechanism wants to start the regulation of the water level inside a lock, as soon as the lock is (nearly) full. In other words, now the mechanism has preferences for different outcomes. We like to know if there is an efficient allocation algorithm that can be used for a strategy-proof Pareto optimal mechanism, in such a setting.

References

1. Abdulkadiroğlu, A., Sönmez, T.: House Allocation with Existing Tenants. Journal of Mathematical Economics 88(2) (1999) 233–260.
2. Bloch, F., Cantala, D.: Markovian Assignment Rules. (2008) Working paper.
3. Jalilzadeh, B.: Employing Mechanism Design for the Online Allocation of Items (2009) To appear.
4. Kurino, M.: House Allocation with Overlapping Agents: A Dynamic Mechanism Design Approach. (2008) Working paper.
5. Ma, Jinpeng.: Strategy-Proofness and the Strict Core in a Market with Indivisibilities. International Journal of Game Theory 23(1) (1994), 75-83.
6. Quint, T.: Restricted houseswapping games. Journal of Mathematical Economics 27(4) (1997), 451-470.
7. Roth, A. E., Postlewaite, A.: Weak versus strong domination in a market with indivisible goods. Journal of Mathematical Economics 4(2) (1977) 131–137.
8. Roth, A., Sönmez, T., Ünver, U.: Pairwise kidney exchange. Journal of Economic Theory. 125(2) (2005) 151–188.
9. Shapley, L.. Scarf, H,: On cores and indivisibility. Journal of Mathematical Economics 1(1) (1974) 23–37.