Pareto optimality in house allocation problems

David J. Abraham[1]*, Katarína Cechlárová[2], David F. Manlove[3]** and Kurt Mehlhorn[4]

[1] Computer Science Department, Carnegie-Mellon University, 5000 Forbes Ave, Pittsburgh PA 15213-3890, USA. *Email:* dabraham@cs.cmu.edu.
[2] Institute of Mathematics, P.J. Šafárik University in Košice, Faculty of Science, Jesenná 5, 040 01 Košice, Slovakia. *Email:* cechlarova@science.upjs.sk.
[3] Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK. *Email:* davidm@dcs.gla.ac.uk.
[4] Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany. *Email:* mehlhorn@mpi-sb.mpg.de.

Abstract. We study Pareto optimal matchings in the context of house allocation problems. We present an $O(\sqrt{n}m)$ algorithm, based on Gale's Top Trading Cycles Method, for finding a maximum cardinality Pareto optimal matching, where n is the number of agents and m is the total length of the preference lists. By contrast, we show that the problem of finding a minimum cardinality Pareto optimal matching is NP-hard, though approximable within a factor of 2. We then show that there exist Pareto optimal matchings of all sizes between a minimum and maximum cardinality Pareto optimal matching. Finally, we introduce the concept of a signature, which allows us to give a characterization, checkable in linear time, of instances that admit a unique Pareto optimal matching.

1 Introduction

We study the problem of allocating a set H of heterogeneous indivisible goods among a set A of agents [14, 8, 3, 4]. We assume that each agent $a \in A$ ranks in order of preference a subset of H (the *acceptable* goods for a) and that monetary compensations are not possible. In the literature the situation in which each agent initially owns one good is known as a *housing market* [14, 12, 11]. When there are no initial property rights, we obtain the *house allocation problem* [8, 16, 1]. A mixed model, in which a subset of agents initially owns a good has also been studied [2]. Yuan [15] describes a large-scale application of these problems in the allocation of families to government-subsidized housing in China.

Following convention we refer to the elements of H as *houses*, though the class of problems under consideration could equally be formulated in terms of allocating graduates to trainee positions, professors to offices, clients to servers, etc. For ease of exposition we begin by assuming that there are no initial property rights, though we later show how to take account of such a situation.

* Work done whilst at Department of Computing Science, University of Glasgow, and Max-Planck-Institut für Informatik.

** Supported by grant GR/R84597/01 from the Engineering and Physical Sciences Research Council and RSE/Scottish Executive Personal Research Fellowship.

Given such a problem instance, the task is to construct a *matching*, i.e. a subset M of $A \times H$ such that $(a, h) \in M$ implies that a finds h acceptable, each agent is assigned to at most one house and vice versa. Furthermore one seeks a matching that is *optimal* in a precise sense, taking into account the agents' preferences. Various notions of optimality have been considered in the literature, but a criterion that has received much attention, particularly from economists, is *Pareto optimality.* A matching M is Pareto optimal if there is no other matching M' such that no agent is worse off in M' than in M, whilst some agent is better off in M' than in M. For example, a matching M is not Pareto optimal if two agents could improve by swapping the houses that they are assigned to in M.

There is a straightforward greedy algorithm, which we denote by Greedy-POM, for finding a Pareto optimal matching [1]: consider each agent a in turn, giving a his/her most-preferred vacant house (assuming such a house exists). This algorithm is also known as a *serial dictatorship* mechanism [1]. Roth and Sotomayor [13, Example 4.3] remark that a similar mechanism is used by the United States Naval Academy in order to match graduating students to their first posts as Naval Officers (in this context however, the algorithm considers each student in non-decreasing order of graduation results). However one may construct an example instance (see Section 2 for further details) in which Pareto optimal matchings may have different cardinalities and Greedy-POM could fail to produce a Pareto optimal matching of maximum size. Yet in many applications, one wishes to match as many agents as possible.

Stronger notions of optimality have been considered in the literature. For example a matching M is *rank-maximal* [10] if, in M, the maximum number of agents are matched to their first-choice house, and subject to this condition, the maximum number of agents are matched to their second-choice house, and so on. Irving et al. [10] describe two algorithms for finding a rank-maximal matching, with complexities $O(\min\{n + C, C\sqrt{n}\}m)$ and $O(Cnm)$, where $n = |A| + |H|$, m is the total length of the preference lists and C is the maximum k such that some agent is assigned to his/her kth-choice house in the constructed matching. Clearly a rank-maximal matching is Pareto optimal, however a rank-maximal matching need not be a maximum cardinality Pareto optimal matching (henceforth a maximum Pareto optimal matching). Alternatively, one may consider a *maximum cardinality maximum utility* matching M, in which we maximise $\sum_{(a,h)\in M} u_{a,h}$ over all maximum cardinality matchings, where $u_{a,h}$ indicates the utility of house h being allocated to agent a. If one defines $u_{a,h} = l - rank_{a,h}$, where $rank_{a,h}$ is the rank of house h in agent a's preference list and l is the maximum length of an agent's list, then a maximum cardinality maximum utility matching is in turn a maximum Pareto optimal matching. Since all utilities are integral, a maximum cardinality maximum utility matching may be found in $O(\sqrt{n}m \log n)$ time [5]. However if one only requires to find a maximum cardinality matching that satisfies the weaker condition of being Pareto optimal, it is of interest to consider whether faster algorithms exist.

The next two sections of this paper work towards answering this question. In Section 2 we give a formal definition of the problem model, and present necessary and sufficient conditions for a matching to be Pareto optimal. In Section

3 we use these conditions as the basis for an $O(\sqrt{n}m)$ algorithm for finding a maximum Pareto optimal matching. This algorithm extends the Top Trading Cycles Method due to Gale [14], which has been the focus of much attention, particularly in the game theory and economics literature [14, 12, 11, 15, 2]. We then show that any improvement to the complexity of our algorithm would imply an improved algorithm for finding a maximum matching in a bipartite graph. We also demonstrate how to modify our algorithm in order to take account of initial property rights, guaranteeing that those who own a good initially will end up with a good that is either the same or better.

In the remainder of the paper, we prove several related results. In Section 4 we consider the problem of finding a minimum Pareto optimal matching, showing that this problem is NP-hard, though approximable within a factor of 2. In Section 5 we prove an interpolation result, showing that there exist Pareto optimal matchings of all sizes between a minimum and maximum Pareto optimal matching. Finally, in Section 6 we give a characterization, checkable in linear time, of instances that admit a unique Pareto optimal matching.

2 Preliminaries

We begin with a formal definition of the problem model under consideration. An instance I of the PARETO OPTIMAL MATCHING problem (POM) comprises a bipartite graph $G = (A, H, E)$, where $A = \{a_1, a_2, \dots, a_r\}$ is the set of *agents* and $H = \{h_1, h_2, \dots, h_s\}$ is the set of *houses*. For each $a_i \in A$, we denote by $A_i \subseteq H$ the vertices adjacent to a_i – these are referred to as the *acceptable* houses for a_i. Moreover a_i has a linear order over A_i. We let $n = r + s$ and $m = |E|$. Henceforth we assume that G contains no isolated vertices.

An *assignment* M is a subset of $A \times H$ such that $(a_i, h_j) \in M$ only if a_i finds h_j acceptable (i.e. $h_j \in A_i$). If $(a_i, h_j) \in M$, we say that a_i and h_j are *assigned* to one another. For each $q \in A \cup H$, let $M(q)$ denote the assignees of q in M. A *matching* is an assignment M such that $|M(q)| \leq 1$ for each $q \in A \cup H$. If $M(q) = \emptyset$, we say that q is *unmatched* in M, otherwise q is *matched* in M.

Let M be a matching in I. M is *maximal* if there is no (agent,house) pair (a_i, h_j) such that a_i and h_j are both unmatched in M and $h_j \in A_i$. Also M is *trade-in-free* if there is no (agent,house) pair (a_i, h_j) such that a_i is matched in M, h_j is unmatched in M, and a_i prefers h_j to $M(a_i)$. Finally M is *coalition-free* if M admits no *coalition*, which is a sequence of matched agents $C = \langle a_0, a_1, \dots, a_{k-1} \rangle$, for some $k \geq 2$, such that a_i prefers $M(a_{i+1})$ to $M(a_i)$ $(0 \leq i \leq k-1)$ (here, and in the remainder of this paper, all subscripts are taken modulo k when reasoning about coalitions). The matching

$$M' = (M \backslash \{(a_i, M(a_i)) : 0 \leq i \leq k-1\}) \cup \{(a_i, M(a_{i+1})) : 0 \leq i \leq k-1\}$$

is defined to be the matching obtained from M by *satisfying* C.

The preferences of an agent extend to matchings as follows. Given two matchings M and M', we say that an agent a_i *prefers* M' *to* M if either (i) a_i is matched in M' and unmatched in M, or (ii) a_i is matched in both M and M' and prefers

$M'(a_i)$ to $M(a_i)$. Given this definition, we may define a relation $\prec$ on the set of all matchings as follows: $M' \prec M$ if and only if no agent prefers M to M', and some agent prefers M' to M. It is straightforward to then prove the following.

Proposition 1. *Given an instance I of POM, the relation $\prec$ forms a strict partial order over the set of matchings in I.*

A matching is defined to be *Pareto optimal* if and only if it is $\prec$-minimal. Intuitively a matching is Pareto optimal if no agent a_i can be better off without requiring another agent a_j to be worse off. The following proposition gives necessary and sufficient conditions for a matching to be Pareto optimal.

Proposition 2. *Let M be a matching in a given instance of POM. Then M is Pareto optimal if and only if M is maximal, trade-in-free and coalition-free.*

Proof. Let M be a Pareto optimal matching. If M is not maximal, then there exists an agent a_i and a house h_j, both unmatched in M, such that $h_j \in A_i$. Let $M' = M \cup \{(a_i, h_j)\}$. If M is not trade-in-free, then there exist an agent a_i and a house h_j, such that a_i is matched in M, h_j is unmatched in M, and a_i prefers h_j to $M(a_i)$. Let $M' = (M \backslash \{(a_i, M(a_i))\}) \cup \{(a_i, h_j)\}$. Finally if M admits some coalition C, let M' be the matching obtained by satisfying C. In all three cases, $M' \prec M$, a contradiction.

Conversely let M be a matching that is maximal, trade-in-free and coalition-free, and suppose for a contradiction that M is not Pareto optimal. Then there exists some matching M' such that $M' \prec M$. Let a_0 be any agent matched in M who prefers M' to M. Note that such an agent must exist, since M is maximal and at least one agent prefers M' to M.

It follows that $M'(a_0)$ is matched in M, say to a_1, for otherwise M is not trade-in-free. Therefore, $M'(a_1) \neq M(a_1)$, and so a_1 must also prefer M' to M. Using this same argument, $M'(a_1)$ is matched in M, say to a_2. We can continue in this manner finding a sequence of agents $\langle a_0, a_1, a_2, \ldots \rangle$, where a_i prefers $M(a_{i+1})$ to $M(a_i)$. Since the number of agents is finite, this sequence must cycle, thereby contradicting the assumption that M is coalition-free. □

Henceforth we will establish the Pareto optimality of a given matching by showing that the conditions of the above proposition are satisfied. For a given matching M, we can trivially check whether M satisfies the maximality and trade-in-free properties in $O(m)$ time. To check for the absence of coalitions, we construct the *envy graph* of M. This is a directed graph, denoted by $G(M)$, consisting of one vertex for each agent, with an edge from a_i to a_j whenever a_j is matched in M and either (i) a_i is unmatched in M and finds $M(a_j)$ acceptable, or (ii) a_i is matched in M and prefers $M(a_j)$ to $M(a_i)$. It is clear that M is coalition-free if and only if $G(M)$ is acyclic. So we can perform this last check in $O(m)$ time by using a cycle-detection algorithm on $G(M)$. Putting these observations together, we have the following result.

Proposition 3. *Let M be a matching in a given instance of POM. Then we may check whether M is Pareto optimal in $O(m)$ time.*

It is easy to construct an instance of POM in which the Pareto optimal matchings are of different sizes. For example let $A = \{a_1, a_2\}$ and let $H = \{h_1, h_2\}$. Suppose that a_1 prefers h_1 to h_2, whilst a_2 finds only h_1 acceptable. Then both $M_1 = \{(a_1, h_1)\}$ and $M_2 = \{(a_1, h_2), (a_2, h_1)\}$ are Pareto optimal. Given this observation it is natural to consider the complexity of each of the problems of finding a maximum and minimum Pareto optimal matching. (Note that Greedy-POM produces M_1 given the agent ordering $\langle a_1, a_2 \rangle$, and produces M_2 given the agent ordering $\langle a_2, a_1 \rangle$.)

3 Maximum Pareto optimal matchings

In this section, we describe a three-phase algorithm for finding a maximum Pareto optimal matching, mirroring the three necessary and sufficient conditions in Proposition 2. We let I be an instance of POM, and we assume the notation and terminology introduced in Section 2. Phase 1 involves using the Hopcroft-Karp algorithm [7] to compute a maximum matching M in G. This phase, which guarantees that M is maximal, takes $O(\sqrt{n}m)$ time and dominates the runtime. The final two phases transform M into a trade-in-free and coalition-free matching respectively. We describe these phases in more detail below.

3.1 Phase 2 of the algorithm

In this phase, we transform M into a trade-in-free matching by repeatedly identifying and promoting agents that prefer an unmatched house to their existing assignment. Each promotion breaks the existing assignment, thereby freeing a house, which itself may be a preferred assignment for a different agent. With the aid of suitable data structures, we can ensure that the next agent and house can be identified efficiently.

For each house h, we maintain a linked list L_h of pairs (a, r), where a is a matched agent who finds h acceptable, and r is the rank of h in a's preference list. Initially the pairs in L_h involve only those matched agents a who prefer h to $M(a)$, though subsequently the pairs in L_h may contain agents a who prefer $M(a)$ to h. The initialization of these lists can be carried out using one traversal of the agent preference lists, which we assume are represented as doubly linked lists or arrays, in $O(m)$ time.

For each matched agent a, we also use this traversal to initialize a variable, denoted by $curr_a$, which stores the rank of $M(a)$ in a's preference list. This variable is maintained during the execution of the algorithm. We also assume that, for each matched agent a we store $M(a)$. One final initialization remains: construct a stack S of all unmatched houses h where L_h is non-empty. We now enter the loop described in Figure 1.

During each loop iteration we pop an unmatched house h from S and remove the first pair (a, r) from the list L_h (which must be non-empty). If a prefers h to $M(a)$ (i.e. $r < curr_a$) then a is promoted from $h' = M(a)$ to h, also M and $curr_a$ are updated, and finally h', which is now unmatched, is pushed onto S if $L_{h'}$ is non-empty. Otherwise h is pushed back onto S if L_h is non-empty.

```
while $S \neq \emptyset$
    $h :=$ $S$.pop();
    $(a, r) :=$ $L_h$.removeHead();
    if $r < curr_a$
        // $h$ is unmatched in $M$, $a$ is matched in $M$ and prefers $h$ to $M(a)$
        $h' := M(a)$;
        $M := (M \backslash \{(a, h')\}) \cup \{(a, h)\}$;
        $curr_a := r$;
        $h := h'$;
    if $L_h \neq \emptyset$
        $S$.push($h$);
```

Fig. 1. Phase 2 loop

Each iteration of the loop removes a pair from a list L_h. Since agent preference lists are finite and no new pair is added to a list L_h during a loop iteration, the while loop must eventually terminate with S empty. At this point no matched agent a would trade $M(a)$ for an unmatched house, and so M is trade-in-free. Additionally, M remains a maximum matching, since any agent matched at the end of Phase 1 is also matched at the end of Phase 2. Finally, it is clear that this phase runs in $O(m)$ time given the data structures described above.

3.2 Phase 3 of the algorithm

In this phase, we transform M into a coalition-free matching. Recall that coalitions in M correspond to cycles in the envy graph $G(M)$. So a natural algorithm involves repeatedly finding and satisfying coalitions in $G(M)$ until no more coalitions remain. This algorithm has a runtime of $O(m^2)$, since there are $O(m)$ coalitions, and cycle-detection takes $O(m)$ time.

A better starting point for an efficient algorithm is Gale's Top Trading Cycles Method [14]. This method is also based on repeatedly finding and satisfying coalitions, however the number of iterations is reduced by the following observation: no agent assigned to his/her first choice can be in a coalition. We remove such agents from consideration, and since the houses assigned to them are no longer exchangeable, they can be deleted from the preference lists of the remaining agents. This observation can now be recursively applied to the reduced preference lists. At some point, either no agents remain, in which case the matching is coalition-free, or no agent is assigned to his/her reduced first choice (i.e. the first choice on his/her reduced preference list).

In this last case, it turns out that there must be a coalition C in M, which can be found in $O(r)$ time by searching the envy graph restricted to reduced first-choice edges. After satisfying C, each agent in C is assigned to his/her reduced first choice. Therefore, no agent is in more than one coalition, giving $O(r)$ coalitions overall. The runtime of this preliminary implementation then is $\Omega(m + r^2)$. We now present a linear-time extension of Yuan's description of the Top Trading Cycles Method [15].

In our implementation, deletions of houses from agents' preference lists are not explicitly carried out. Instead, a house that is no longer exchangeable is

```
for each matched agent $a$ such that $p(a) \neq M(a)$
    $P := \{a\}$;    // $P$ is a stack of agents
    $c(a) := 1$;    // counters record the number of times an agent is in $P$
    while $P \neq \emptyset$
        $a' := P$.pop();
        $p(a')$ := most-preferred unlabelled house on preference list of $a'$;
        if $c(a') = 2$
            $C$ := coalition in $P$ containing $a'$;
            satisfy $C$;
            for each $a'' \in C$
                label $M(a'')$;
                $c(a'') := 0$;
                $P$.pop();
        else if $p(a') = M(a')$
            label $M(a')$;
            $c(a') := 0$;
        else
            $P$.push($a'$);
            $a'' := M(p(a'))$;
            $c(a'') := c(a'') + 1$;
            $P$.push($a''$);
```

Fig. 2. Phase 3 loop

labelled (all houses are initially unlabelled). For each agent a we maintain a pointer $p(a)$ to the first unlabelled house on a's preference list – this is equivalent to the first house on a's reduced preference list. Initially $p(a)$ points to the first house on a's preference list, and subsequently $p(a)$ traverses left to right. Also, in order to identify coalitions, we initialize a counter $c(a)$ to 0 for each agent a. Then, we enter the main body of the algorithm, as given in Figure 2.

This algorithm repeatedly searches for coalitions, building a path P of agents (represented by a stack) in the (simulated) envy graph restricted to reduced first-choice edges. At each iteration of the while loop, we pop an agent a' from the stack and move up $p(a')$ if necessary. If P cycles (i.e. we find that $c(a') = 2$), there is a coalition C – the agents involved in C are removed from consideration and the houses assigned to these agents are labelled (in practice the agents in C can be identified and C can be satisfied during the stack popping operations). Alternatively, if P reaches a dead-end (a' is already assigned to his/her first choice), this agent is removed from consideration and his/her assigned house is labelled. Otherwise, we keep extending the path by following the reduced first-choice edges.

At the termination of this phase we note that M is coalition-free by the correctness of the Top Trading Cycles Method [14]. Also M remains a maximum trade-in-free matching, since each agent and house matched at the end of Phase 2 is also matched at the end of Phase 3. Finally, it is clear this phase runs in $O(m)$ time given the data structures described above. We summarize the preceding discussion in the following theorem.

Theorem 1. *A maximum Pareto optimal matching can be found in $O(\sqrt{n}m)$ time. Such a matching is also a maximum matching of agents to houses.*

We now show that any improvement to the complexity of the above algorithm would imply an improved algorithm for finding a maximum matching in a bipartite graph. Without loss of generality, let $G = (A, H, E)$ be an arbitrary bipartite graph with no isolated vertices. Construct an instance I of POM with bipartite graph G, where each agent a's preference list in I is an arbitrary permutation over a's neighbours in G. By Theorem 1, any maximum Pareto optimal matching in I is also a maximum matching in G. Since I may be constructed from G in $O(m)$ time, the complexity of finding a maximum matching in a bipartite graph is bounded above by the complexity of finding a maximum Pareto optimal matching.

3.3 Initial property rights

Suppose that a subset A' of the agents already own a house. We now describe an *individually rational* modification of our algorithm, which ensures that every agent in A' ends up with the same house or better.

We begin with a matching M that pre-assigns every agent $a \in A'$ to his/her existing house h. We then truncate the preference list of each such a by removing all houses less preferable than $M(a)$. Now, we enter Phase 1, where we use the Hopcroft-Karp algorithm to exhaustively augment M into some matching M'. Members of A' must still be matched in M', and since their preference lists were truncated, their new assignments must be at least as preferable as those in M. Note that M' may not be a maximum matching of A to H, however M' does have maximum cardinality among all matchings that respect the initial property rights. The remaining two phases do not move any agent from being matched to unmatched, and so the result follows immediately.

In the special case that all agents own a house initially (i.e. I is an instance of a housing market), it is clear that Phases 1 and 2 of the algorithm are not required. Moreover it is known that Phase 3 produces the unique matching that belongs to the core of the market [12], a stronger notion than Pareto optimality.

4 Minimum Pareto optimal matchings

In this section, we consider the problem of finding a minimum Pareto optimal matching. Let MIN-POM denote the problem deciding, given an instance I of POM and an integer K, whether I admits a Pareto optimal matching of size at most K. We firstly prove that MIN-POM is NP-complete via a reduction from MMM, which is the problem of deciding, given a graph G and an integer K, whether G admits a maximal matching of size at most K.

Theorem 2. *MIN-POM is NP-complete.*

Proof. By Proposition 3, MIN-POM belongs to NP. To show NP-hardness, we give a reduction from the NP-complete restriction of MMM to subdivision graphs

[6] (given a graph G, the *subdivision graph* of G is obtained by subdividing each edge $e = \{u, w\}$ into two edges $\{u, v_e\}, \{v_e, w\}$, where v_e is a new vertex corresponding to e).

Let $G = (V, E)$ (a subdivision graph) and K (a positive integer) be given as an instance of MMM. Then V is a disjoint union of two sets U and W, where each edge $e \in E$ joins a vertex in U to a vertex in W. Assume that $U = \{u_1, u_2, \ldots, u_r\}$ and $W = \{w_1, w_2, \ldots, w_s\}$. Without loss of generality assume that each vertex $u_i \in U$ has degree 2, and moreover assume that p_i and q_i are two sequences such that $p_i < q_i$, $\{u_i, w_{p_i}\} \in E$ and $\{u_i, w_{q_i}\} \in E$.

We create an instance I of MIN-POM as follows. Let A be the set of agents and let H be the set of houses, where $A = A^1 \cup A^2$, $A^t = \{a_1^t, a_2^t, \ldots, a_r^t\}$ $(t = 1, 2)$, $H = W \cup X$ and $X = \{x_1, x_2, \ldots, x_r\}$. For each i $(1 \le i \le r)$, we create preference lists for agents a_i^1 and a_i^2 as follows:

$$a_i^1 : x_i \quad w_{p_i} \quad w_{q_i} \qquad\qquad a_i^2 : x_i \quad w_{q_i} \quad w_{p_i}$$

We claim that G has a maximal matching of size at most K if and only if I has a Pareto optimal matching of size at most $K + r$.

For, suppose that M is a maximal matching in G of size at most K. We construct a set M' as follows. For any $u_i \in U$ that is unmatched in M, add the pair (a_i^1, x_i) to M'. Now suppose that $(u_i, w_j) \in M$. If $j = p_i$, add the pairs (a_i^1, w_j) and (a_i^2, x_i) to M'. If $j = q_i$, add the pairs (a_i^1, x_i) and (a_i^2, w_j) to M'. Clearly M' is a matching in I, and $|M'| = |M| + r \le K + r$. It is straightforward to verify that, by the maximality of M in G, M' is Pareto optimal in I.

Conversely suppose that M' is a Pareto optimal matching in I of size at most $K + r$. For each i $(1 \le i \le r)$, either $(a_i^1, x_i) \in M'$ or $(a_i^2, x_i) \in M'$, for otherwise M' is not trade-in-free. Hence we may construct a matching M in G as follows. For each i $(1 \le i \le r)$, if $(a_i^t, w_j) \in M'$ for some t $(1 \le t \le 2)$, add (u_i, w_j) to M. Then $|M| = |M'| - r \le K$. The maximality of M' clearly implies that M is maximal in G. □

For a given instance I of POM with bipartite graph G, we denote by $p^-(I)$ and $p^+(I)$ the sizes of a minimum and maximum Pareto optimal matching in I respectively. Similarly, we denote by $\beta_1^-(G)$ and $\beta_1(G)$ the sizes of a minimum maximal and a maximum matching in G respectively. It is known that $\beta_1^-(G) \ge \beta_1(G)/2$ [9]. By Proposition 2, Pareto optimal matchings in I are maximal matchings in G. Hence, by Theorem 1, we have that $\beta_1^-(G) \le p^-(I) \le p^+(I) = \beta_1(G)$. It is therefore immediate that, for a given instance I of POM, the problem of finding a minimum Pareto optimal matching is approximable within a factor of 2.

5 Interpolation of Pareto optimal matchings

In this section, we prove that, for a given instance I of POM, there are Pareto optimal matchings of all sizes between $p^-(I)$ and $p^+(I)$.

Given a matching M, an augmenting path P for M is an alternating sequence of distinct agents and houses $\langle a_1, h_1, a_2, \ldots, a_k, h_k \rangle$, where a_1 and h_k are unmatched in M, $h_i \in A_i$, and $M(a_{i+1}) = h_i$ $(1 \leq i \leq k-1)$. We associate with each such augmenting path a vector $rank_P$, whose ith component contains the rank of a_i for h_i. Given two augmenting paths P and Q for M, we say that $P \lhd Q$ if (i) both P and Q begin from the same agent, and (ii) $rank_P$ is lexicographically less than $rank_Q$. Also for paths P and Q, we define three operations: $Prefix_P(v)$ is the substring of P from a_1 to $v \in P$, $Suffix_P(v)$ is the substring of P from $v \in P$ to h_k, and $P \cdot Q$ denotes the concatenation of P and Q.

Theorem 3. *For a given instance I of POM, there exist Pareto optimal matchings of size k, for each k such that $p^-(I) \leq k \leq p^+(I)$.*

Proof. Let M be any Pareto optimal matching such that $|M| < p^+(I)$, and let M' be the matching that results from augmenting M along some $\lhd$-minimal augmenting path P. We will show in turn that M' is maximal, trade-in-free and coalition-free; the result then follows by induction.

If M' is not maximal, then clearly we contradict the maximality of M. Now suppose that M' is not trade-in-free. Then there exists an agent a and house h, where a is matched in M', h is unmatched in M', and a prefers h to $M'(a)$. Since h is also unmatched in M, a must be in P, for otherwise $M(a) = M'(a)$, and M is not trade-in-free. But then $P' = Prefix_P(a) \cdot \langle h \rangle$ is an augmenting path for M, contradicting the $\lhd$-minimality of P.

Finally suppose for a contradiction that M' is not coalition-free. Then there exists a coalition $C = \langle a_0, a_1, \ldots, a_{k-1} \rangle$ with respect to M'. At least one agent in P must also be in C, for otherwise M is not coalition-free. Let a_i be the first such agent in P. We establish some properties of $M'(a_{i+1})$. Firstly, $M'(a_{i+1})$ must be matched in M, for otherwise M admits the augmenting path $Prefix_P(a_i) \cdot \langle M'(a_{i+1}) \rangle$, contradicting the $\lhd$-minimality of P. Also, $M'(a_{i+1})$ cannot appear before a_i in P, for otherwise a_i is not the first agent in P to be in C. Lastly, $M'(a_{i+1})$ cannot appear after a_i in P, for otherwise M admits the augmenting path $Prefix_P(a_i) \cdot Suffix_P(M'(a_{i+1}))$, contradicting the $\lhd$-minimality of P. So, it must be the case that $M'(a_{i+1})$ is matched in M and does not appear in P. Let a_{i+j} be the first agent in C after a_i, such that a_{i+j} is in P. Note that $a_{i+j} \neq a_{i+1}$ by the above properties of $M'(a_{i+1})$, but since C is a cycle, $a_{i+j} = a_i$ is possible. It follows that the subsequence $S = \langle M'(a_{i+1}), a_{i+1}, \ldots, M'(a_{i+j-1}), a_{i+j-1} \rangle$ of C is disjoint from P, and so $P' = Prefix_P(a_i) \cdot S \cdot Suffix_P(M'(a_{i+j}))$ is a valid augmenting path of M. But then P' contradicts the $\lhd$-minimality of P, since a_i prefers $M'(a_{i+1})$ to $M'(a_i)$. □

Corollary 1. *Given an instance I of POM and a Pareto optimal matching M in I of size k, we can construct a Pareto optimal matching M' of size $k+1$, or determine that no such matching exists, in $O(m)$ time.*

Proof. Let G be the bipartite graph in I, with edges in M directed from H to A, and edges not in M directed from A to H. Also associate with each non-matching edge (a_i, h_j) the rank of a_i for h_j. We search for a $\lhd$-minimal augmenting path by performing an ordered depth first search of G starting from the set of unmatched agents, where for each agent a in the search, we explore outgoing edges from a in increasing order of rank. In general, ordered depth-first search is asymptotically slower than depth-first search. However, the $O(m)$ result holds, since each preference list is already given in increasing order of rank. □

We remark that the results of this section extend to the case where a subset of the agents have initial property rights.

6 Uniqueness of Pareto optimal matchings

In this section, we give a characterization of instances with no initial property rights that admit a unique Pareto optimal matching. This is based on the concept of a *signature* of a Pareto optimal matching.

If a matching M is Pareto optimal, the envy graph $G(M)$ contains no cycles, and therefore admits a topological ordering. We say that a reversed topological ordering of $G(M)$, denoted by $\sigma(M)$, is a *signature* of M. The next lemma will help us establish that the signature of a matching is unique for that matching. This lemma is similar to [1, Lemma 1], though the proof here, which uses the concept of a signature, is much simpler.

Lemma 1. *Given an instance I of POM, the algorithm Greedy-POM can generate any Pareto optimal matching in I.*

Proof. Let M be an arbitrary Pareto optimal matching in I. We claim that by processing the agents in order of $\sigma(M)$, the greedy algorithm returns M.

Suppose for a contradiction that Greedy-POM returns a matching $M' \neq M$. It follows that since M' is Pareto optimal, some agent must prefer M' to M. Let a be the first such agent in $\sigma(M)$.

Now, $M'(a)$ must be matched in M, say to a', for otherwise M is not maximal (if a is unmatched in M), or M is not trade-in-free (if a is matched in M). $G(M)$ must therefore contain an edge from a to a', meaning that a' precedes a in $\sigma(M)$. At the time a' is processed by Greedy-POM, $M'(a)$ is unmatched (since it is assigned later to a). So, a' must prefer $M'(a')$ to $M(a') = M'(a)$, contradicting the assumption that a was the first such agent in $\sigma(M)$. □

Corollary 2. *Given an instance I of POM, every agent permutation is a signature of exactly one Pareto optimal matching in I.*

We can now present a necessary and sufficient condition, checkable in linear time, for a POM instance to admit a unique Pareto optimal matching.

Theorem 4. *An instance I of POM admits a unique Pareto optimal matching M if and only if every agent is matched in M with his/her first choice.*

Proof. Let M be the unique Pareto optimal matching in I. Since every agent permutation is a signature of M, $G(M)$ contains no edges. Then every agent must be matched to his/her first choice.

Conversely, let M be a matching in I in which every agent is matched with his/her first choice. Then if M' is any matching in I such that $M' \neq M$, it follows that $M \prec M'$. Hence M is the unique Pareto optimal matching in I. □

7 Concluding remarks

We conclude with an open problem. The basic POM definition given in Section 2 can be generalized by permitting agents to contain *ties* in their preference lists (i.e. to rank equally two or more houses). In this context the definition of the relation $\prec$ is the same as that given in Section 2, and hence the definition of Pareto optimality remains unchanged. A maximum Pareto optimal matching can be found in $O(\sqrt{n}m \log n)$ time using a similar reduction to the Assignment problem as described in Section 1 (in this case $rank_{a,h}$ is the number of houses that a prefers to h). However is the problem of finding a maximum Pareto optimal matching solvable in $O(\sqrt{n}m)$ time?

References

1. A. Abdulkadiroğlu and T. Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701, 1998.
2. A. Abdulkadiroğlu and T. Sönmez. House allocation with existing tenants. *Journal of Economic Theory*, 88:233–260, 1999.
3. X. Deng, C. Papadimitriou, and S. Safra. On the complexity of equilibria. *Journal of Computer and System Sciences*, 67(2):311–324, 2003.
4. S.P. Fekete, M. Skutella, and G.J. Woeginger. The complexity of economic equilibria for house allocation markets. *Inf. Proc. Lett.*, 88:219–223, 2003.
5. H.N. Gabow and R.E. Tarjan. Faster scaling algorithms for network problems. *SIAM Journal on Computing*, 18(5):1013–1036, 1989.
6. J.D. Horton and K. Kilakos. Minimum edge dominating sets. *SIAM Journal on Discrete Mathematics*, 6:375–387, 1993.
7. J.E. Hopcroft and R.M. Karp. A $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 2:225–231, 1973.
8. A. Hylland and R. Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):293–314, 1979.
9. B. Korte and D. Hausmann. An analysis of the greedy heuristic for independence systems. *Annals of Discrete Mathematics*, 2:65–74, 1978.
10. R.W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. Rank-maximal matchings. *Proceedings of SODA '04*, pages 68–75. ACM-SIAM, 2004.
11. A.E. Roth. Incentive compatibility in a market with indivisible goods. *Economics Letters*, 9:127–132, 1982.
12. A.E. Roth and A. Postlewaite. Weak versus strong domination in a market with indivisible goods. *Journal of Mathematical Economics*, 4:131–137, 1977.
13. A.E. Roth and M.A.O. Sotomayor. *Two-sided matching: a study in game-theoretic modeling and analysis.* Cambridge University Press, 1990.
14. L. Shapley and H. Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1:23–37, 1974.
15. Y. Yuan. Residence exchange wanted: a stable residence exchange problem. *European Journal of Operational Research*, 90:536–546, 1996.
16. L. Zhou. On a conjecture by Gale about one-sided matching problems. *Journal of Economic Theory*, 52(1):123–135, 1990.