A Generalization of the AL method for Fair Allocation of Indivisible Objects

Haris Aziz

Abstract

We consider the assignment problem in which agents express ordinal preferences over \( m \) objects and the objects are allocated to the agents based on the preferences. In a recent paper, Brams, Kilgour, and Klamler (2014) presented the AL method to compute an envy-free assignment for two agents. The AL method crucially depends on the assumption that agents have strict preferences over objects. We generalize the AL method to the case where agents may express indifferences and prove the axiomatic properties satisfied by the algorithm. As a result of the generalization, we also get a \( O(m) \) speedup on previous algorithms to check whether a complete envy-free assignment exists or not.

Keywords Fair division · envy-freeness · AL method

1 Introduction

Fair allocation of resources is one of the most critical issues for society. A basic, yet widely applicable, problem in computer science and economics is to allocate discrete objects to agents given the ordinal preferences of the agents over the objects. The setting is referred to as the assignment problem or the house allocation problem (see, e.g., Abraham et al., 2005; Aziz et al., 2014a; Bouveret et al., 2010; Brams and Kaplan, 2004; Brams et al., 2003; Brams and Fishburn, 2000; Brams et al., 2012; Demko and Hill, 1988; Gärdenfors, 1973; Manlove, 2013; Wilson, 1977; Young, 1995). In this setting, there is a set of agents \( N = \{1, \ldots, n\} \), a set of objects \( O = \{o_1, \ldots, o_m\} \) with each agent \( i \in N \) expressing ordinal preferences \( \succ_i \) over \( O \). The goal is to allocate the objects among the agents in a fair or
optimal manner without allowing transfer of money. The model is applicable to many resource allocation or fair division settings where the objects may be public houses, school seats, course enrolments, kidneys for transplant, car park spaces, chores, joint assets of a divorcing couple, or time slots in schedules.

For the assignment problem, the case of two agents is especially central. Many disputes are between two parties and may require division of common resources. Divorce proceedings are one of the settings when common assets need to be divided among the two parties. Other examples in history include partition of countries which results in the need to divide common assets.

When objects are allocated among agents, it is desirable that they are allocated in a fair and efficient manner. For fairness, one of the most established concepts is envy-freeness. A formal study of envy-freeness in microeconomics can be traced back to the work of Foley (1967). Envy-freeness requires that each agent should prefer its allocation over other agents’ allocations. Envy-freeness can be trivially satisfied by not giving any objects to any agents. However, if we insist on allocating all the objects, no assignment may be envy-free as is the case when there is only one object! The most established notion of efficiency is Pareto optimality which requires that there should be no other allocation which each agent weakly prefers and at least one agent strictly prefers. Pareto optimality has been termed the “single most important tool of normative economic analysis” (Moulin, 2003).

In view of the importance of the two-agent setting and the fundamental goals of envy-freeness and Pareto optimality, Brams et al. (2014a) presented an elegant algorithm called AL for the case of two agents that computes a maximal assignment that is envy-free as well as locally Pareto optimal (Pareto optimal for the set of allocated objects). Since there may not exist a Pareto optimal and envy-free assignment, Brams et al. (2014a) relax the requirement of Pareto optimality to local Pareto optimality. The algorithm has received attention (see e.g., Brams, 2014a,b). The desirable aspect of AL is that if there exists a complete envy-free assignment, it returns it. If there does not exist a complete envy-free assignment it still returns a partial assignment that is is envy-free. One possible limitation of the AL method is that it assumes that agents have strict preferences over objects. We present a generalization of the AL method when agents may express indifferences among agents.

Indifferences in preferences are not only a natural relaxation but are also a practical reality in many cases. For example, if there are multiple copies of the same object with the same characteristics, then an agent is invariably indifferent among all such copies. Indifferences can lead to various challenges. The complexity of solution concepts for the case of indifferences can be considerably more. A famous example is that of roommate markets for which the problem of finding a stable matching is polynomial-time solvable for strict preferences but NP-complete for weak orders (Ronn, 1990). Similarly, a number of fairness concepts are harder to compute when weak orders are allowed (Aziz et al., 2014a). In view of this, effort has been taken to generalize algorithms and rules for the case of indifferences in voting (see e.g., Aziz et al., 2013a; Cullinan et al., 2014), housing mar-

---

1 The notion of envy-freeness that they use is equivalent to SD (stochastic dominance) envy-freeness (Aziz et al., 2014a) and necessary envy-freeness (Bouveret et al., 2016).
kets (see e.g., Aziz and de Keijzer, 2013; Saban and Sethuraman, 2013), coalition formation (Aziz et al., 2013b), and various matching models (Manlove, 2013). The main contribution of this paper is a generalization of AL which we refer to as GAL for the case when agents may express indifferences. The main result of the paper is as follows.

**Theorem 1** For two agents, GAL returns in time $O(m^2)$ a maximal envy-free and locally Pareto optimal assignment even if agents express weak preferences. Moreover, there exists no other assignment that Pareto dominates it and is still envy-free.

Previously, Bouveret et al. (2010) and Aziz et al. (2014a) presented $O(m^3)$ time algorithms to check whether a complete EF assignment exists or not. The algorithms require solving networks flows or maximum matching. As a corollary of GAL, we obtain a $O(m^2)$ simpler algorithm to check whether there exists a complete assignment that is EF.

The critical reader may ask whether GAL can be generalized to handle an arbitrary number of agents. We show that unless complexity classes P and NP coincide (Fortnow, 2013), there exists no polynomial-time algorithm for an arbitrary number of agents that satisfies the same properties as GAL.

2 Related work

Computation of fair discrete assignments has been intensely studied in the last decade. In many of the papers considered, agents express cardinal utilities for the objects and the goal is to compute fair assignments (see e.g., Lipton et al., 2004; Procaccia and Wang, 2014). We consider the setting in which agents only express ordinal preferences over objects (Aziz et al., 2014a; Bouveret et al., 2010; Brams and Kaplan, 2004; Brams et al., 2003; Brams and Fishburn, 2000; Pruhs and Woeginger, 2012) which are less demanding to elicit.

When agents express preferences over objects and we need to reason about preferences over allocations, there are different ways one can define envy-freeness. In this paper we will use the strongest known reasonable notion of envy-freeness. The notion is equivalent to necessary envy-freeness in (Bouveret et al., 2010), SD-envy-freeness in (Aziz et al., 2014a), EF notion used in (Brams et al., 2014a) and pairwise envy-freeness in (Brams et al., 2014b). We will refer to the notion simply as EF just like Brams et al. (2014a) do. Aziz et al. (2014b) and Bouveret et al. (2010) presented $O(m^3)$ algorithms to check whether there exists an EF assignment. We show that there exists a simple $O(m^2)$ algorithm for the problem even if agents express weak preferences.

There are other papers in fair division in which agents explicitly express ordinal preferences over sets of objects rather than simply expressing preferences over objects. For these more expressive models, the computational complexity of computing fair assignments is either even higher (Chevaleyre et al., 2006; de Keijzer et al., 2009) or representing preferences requires exponential space (Aziz, 2014; Brams et al., 2012). In this paper, we restrict agents to simply express ordinal preferences over objects.
3 Preliminaries

An assignment problem is a triple \((N, O, \succeq)\) such that \(N = \{1, \ldots, n\}\) is a set of agents, \(O = \{o_1, \ldots, o_m\}\) is a set of objects, and the preference profile \(\succeq = (\succeq_1, \ldots, \succeq_n)\) specifies for each agent \(i\) its preference \(\succeq_i\) over \(O\). Agents may be indifferent among objects. We denote \(\succeq_i: E_i^1, \ldots, E_i^{k_i}\) for each agent \(i\) with equivalence classes in decreasing order of preferences. Thus, each set \(E_i^j\) is a maximal equivalence class of objects among which agent \(i\) is indifferent, and \(k_i\) is the number of equivalence classes of agent \(i\). If an equivalence class is a singleton \(\{o\}\), we list the object \(o\) in the list without the curly brackets. An assignment \(p = (p(1), \ldots, p(n))\) specified the allocation of objects \(p(i)\) to each \(i \in N\) such that \(p(i) \subseteq O\) and \(p(i) \cap p(j) = \emptyset\) for all \(i \neq j\).

We first show that EF implies SD envy-freeness. Suppose \(p\) satisfies EF and \(q(i) \succeq_{SD} p(i)\) for each \(o \in O\).

Agent \(i\) strictly SD prefers \(p(i)\) to \(q(i)\) if \(p(i) \succ_{SD} q(i)\) and \(\neg[q(i) \succ_{SD} p(i)]\). Although each agent \(i\) expresses ordinal preferences over objects, he could have a private cardinal utility \(u_i\) consistent with \(\succeq_i\): \(u_i(o) \geq u_i(o')\) if and only if \(o \succeq_i o'\).

An assignment \(p\) is SD-efficient if there exists no other assignment \(q\) such that \(q(i) \succeq_{SD} p(i)\) for all \(i \in N\) and \(q(i) \succ_{SD} p(i)\) for some \(i \in N\). SD-efficiency is equivalent to Pareto optimality for discrete assignments. Hence we will refer to SD-efficiency as Pareto optimality and SD-domination as Pareto domination.

An assignment \(p\) satisfies SD envy-freeness if each agent weakly SD prefers its allocation to that of any other agent:

\[p(i) \succeq_{SD} p(j)\] for all \(i, j \in N\).

From the definition it is easy to see that a necessary condition for SD envy-freeness is that each agent gets the same number of objects.

Brams et al. (2014a) defined EF as follows. An allocation \((p(1), p(2))\) satisfies EF if \(|p(1)| = |p(2)|\) and there exists an injection \(f_1: p(1) \to p(2)\) and an injection \(f_2: p(2) \to p(1)\) such that for each object \(o \in p(1)\), \(1\) (weakly) prefers \(o\) to \(f_1(o)\) and for each object \(o \in p(2)\), \(2\) (weakly) prefers \(o\) to \(f_2(o)\). Then, by using a similar argument as (Lemma 1, Brams et al. (2014a), we can show that EF is equivalent to SD envy-freeness. We detail the argument for the sake of completeness and to formally extend Lemma 1 (Brams et al., 2014a) to the case of indifferences.

**Lemma 1** EF is equivalent to SD envy-freeness.

**Proof** We first show that EF implies SD envy-freeness. Suppose \(p\) satisfies EF and take any object \(o \in O\). Suppose that there is an object \(o' \in p(i)\) such that \(f_i(o') \succeq_i o\). By the definition of \(f_i\), we know that \(o' \succeq_i f_i(o')\). Since \(f_i(o') \succeq_i o\), we get that \(o' \succeq_i f_i(o')\). Hence,

\[\footnote{We use the same definition as in Brams et al. (2014a) except that we use weakly prefers rather than strictly prefers since we are considering weak preferences.} \]
Since, if the number of agents is constant, it can be checked in \( O(n^2) \) time whether a given assignment is EF or not.

**Proof** We show that it can be checked in \( O(m) \) time whether a given assignment for constant number agents is SD envy-free or not. We first show that an SD comparison between any two allocations can be made in \( O(m) \) time. Let us say that we want to check whether \( p(i) \) and \( p(-i) \) are different. Without loss of generality, assume that \( i \)'s preferences are a coarsening of linear order \( o_1, \ldots, o_m \).

- We construct in \( O(m) \) a vector \( x(p(i)) = (x_1, \ldots, x_m) \) where \( x_i = 1 \) if \( o_i \in p(i) \) and \( x_i = 0 \) otherwise. Using \( x(p(i)) \) we construct in \( O(k_i) \) time a vector \( s'(p(i)) = (s'_1, \ldots, s'_m) \) where \( s'_j = |E'_j| \cap p(i) \). Using \( s'(p(-i)) \) we construct in \( O(k_i) \) time a vector \( s(p(i)) = (s_1, \ldots, s_k) \) where \( s_j = \sum_{i=1}^m s'_i \).

- In a similar way, we construct in \( O(m) \) a vector \( y(p(-i)) = (y_1, \ldots, y_m) \) where \( y_j = 1 \) if \( o_j \in p(-i) \) and \( y_j = 0 \) otherwise. Using \( y(p(-i)) \) we construct in \( O(k_i) \) time a vector \( t'(p(-i)) = (t'_1, \ldots, t'_k) \) where \( t_j = |E'_j| \cap p(-i) \). Using \( t'(p(-i)) \) we construct in \( O(k_i) \) time a vector \( t(p(i)) = (t_1, \ldots, t_k) \) where \( t_j = \sum_{i=1}^k t'_j \).

Now \( p(i) \) and \( p(-i) \) are different if and only if there exists some \( j \) such that \( t_j \neq s_j \). This again takes time \( O(k_i) \). Hence an SD comparison between allocation takes time \( O(m) + 4O(k_i) = O(m) \).

In order to test EF, we need to make \( n(n - 1) \) comparisons which is constant if \( n \) is constant. Hence testing EF of an assignment for constant number of agents takes time \( O(m) \). \( \square \)

In the paper, we will assume that \( n = 2 \) i.e., there are two agents. If we refer to some agent as \( i \in \{1, 2\} \), then we will refer to the other agent as \(-i\).

### 4 GAL — Generalized AL

Before we delve into GAL, we first informally describe a simplified version of AL that still satisfies the properties of AL as described in Brans et al. (2014). Agents have strict preferences and in each round they pick one object each. The algorithm repeats the following until all objects have been allocated to agent 1, 2, or contested pile \( C \). If the most preferred unallocated object of the agents is not the same, each
agent picks its most preferred object. Otherwise, if the most preferred unallocated object \( o \) coincides, then we check whether we can give it to agent 1. If \( o \) is given to agent 1 and the next most preferred unallocated object is given to agent 2 and the partial assignment satisfies EF, then we allow such an allocation in the round. If not, we check in the same way whether we can give it to agent 2. If \( o \) cannot be given to either of the two, we put it in \( C \).

The general idea of GAL is as follows. Since the preferences of the two agents are weak orders, we first construct unique linear orders called priority orders based on the preferences. Although, the comparisons to check the feasibility of EF assignments are still done with respect to the original preferences, the constructed linear orders help identify which unique object should each agent try to get first. The priority orders are refinements of the preferences where, if an agent is indifferent between two objects, it has higher priority for the object less preferred by the other agent. If both agents are indifferent among two objects, then agent 1 has higher priority for the object with the lower index and agent 2 has higher priority for the object with the higher index. After suitably constructing the linear orders, \( >_1 \) and \( >_2 \), agents try to take the maximal object according to the priority orders. If agents have a different maximal object, they take their maximal object. Otherwise there is a conflict so we must try to give one of the agents the maximal object and give the other agent the next maximal object according to the priority list if it does not violate EF. If this cannot be done, we send the contested object to \( C \), the so called contested pile. A key idea behind GAL is that if an object \( o^* \) is sent to the contested pile, then it cannot be the case that \( o^* \) along with some subsequent less preferred objects are allocated to agents and EF is not violated. The algorithm is formally defined as Algorithm 1. Note there is an asymmetry in the algorithm in that agent one is considered first to get object \( o^* \) in Step 17. One can consider any of the two agents first or even toss a coin to select one agent. The properties of the algorithm are not affected.

We present a couple of examples to illustrate how GAL works. The contested pile is empty in one example and non-empty in another.

**Example 1**

\[
\geq_1: \{o_1, o_2, o_3\}, \{o_4, o_5, o_6\} \\
\geq_2: \{o_2, o_3, o_4\}, \{o_6\}, \{o_1, o_5\}
\]

\[
>_1: \{o_1, o_2, o_3, o_5, o_6, o_4\} \\
>_2: \{o_4, o_3, o_2, o_6, o_5, o_1\}
\]

(i) Round 1: \( p(1) = \{o_1\}, p(2) = \{o_4\}, C = \emptyset \);  
(ii) Round 2: \( p(1) = \{o_1, o_2\}, p(2) = \{o_4, o_3\}, C = \emptyset \);  
(iii) Round 2: \( p(1) = \{o_1, o_2\}, p(2) = \{o_4, o_3\}, C = \emptyset \);  
(iv) Round 3: \( p(1) = \{o_1, o_2, o_3\}, p(2) = \{o_4, o_3, o_6\}, C = \emptyset \).

---

3 This feasibility check is phrased in a different way in the original description of AL but is equivalent to checking for EF.

4 The view of EF as being defined with respect to the SD relation makes it easy to argue for a maximal EF assignment.
Algorithm 1 GAL — algorithm for envy-free assignment of indivisible objects to two agents

Input: \((\mathcal{A}, \mathcal{I}), O)\)

Output: EF assignment \(p\)

1. Construct a linear order \(\succ\) for agent 1: for all \(i, j \in \{1, \ldots, m\}, a_i \succ_1 a_j\) if \(a_i \succ_1 a_j; a_j \succ_1 a_i\) if \(a_i \geq_1 a_j\)

2. Construct the linear order \(\succ_2\) for agent 2: for all \(i, j \in \{1, \ldots, m\}, a_i \succ_2 a_j\) if \(a_i \succ_2 a_j; a_j \succ_2 a_i\) if \(a_i \geq_1 a_j\) and \(a_i \geq_2 a_j\) if \(a_i \geq_2 a_j\) and \(a_i \geq_1 a_j\) and \(i < j\).

3. \(O' \leftarrow O\)

4. \(p(1) \leftarrow \emptyset; p(2) \leftarrow \emptyset\)

5. \(C \leftarrow \emptyset\)

6. round number \(t \leftarrow 0\)

7. while \(O' \neq \emptyset\) do

8. \(t \leftarrow t + 1\)

9. if \(\max_{\succ_1}(O') \neq \max_{\succ_2}(O')\) then

10. \(p(1) \leftarrow p(1) \cup \max_{\succ_1}(O')\)

11. \(p(2) \leftarrow p(2) \cup \max_{\succ_2}(O')\)

12. \(O' \leftarrow O' \setminus \{\max_{\succ_1}(O'), \max_{\succ_2}(O')\}\)

13. return “no”

14. else

15. \(o' \leftarrow \max_{\succ_1}(O') = \max_{\succ_2}(O')\)

16. \(O' \leftarrow O' \setminus \{o'\}\)

17. if \((p(1) \cup \{o'\}, p(2) \cup \{\max_{\succ_2}(O')\})\) is EF w.r.t \(\succeq_1\) then

18. \(p(1) \leftarrow p(1) \cup \{o'\}\)

19. \(p(2) \leftarrow p(2) \cup \{\max_{\succ_2}(O')\}\)

20. \(O' \leftarrow O' \setminus \{\max_{\succ_2}(O')\}\)

21. else if \((p(1) \cup \{\max_{\succ_1}(O')\}, p(2) \cup \{o'\})\) is EF w.r.t \(\succeq_1\) then

22. \(p(2) \leftarrow p(2) \cup \{o'\}\)

23. \(p(1) \leftarrow p(1) \cup \{\max_{\succ_1}(O')\}\)

24. \(O' \leftarrow O' \setminus \{\max_{\succ_1}(O')\}\)

25. else

26. \(C \leftarrow C \cup \{o'\}\)

27. end if

28. end if

29. end while

30. return \((p(1), p(2))\)

Example 2

\(\succeq_1: \{a_7\}, \{a_1, a_2, a_3\}, \{a_4, a_5, a_6\}\)

\(\succeq_2: \{a_7\}, \{a_1\}, \{a_3\}, \{a_4, a_5\}, \{a_2, a_6\}\)

\(\succeq_1: \{a_7, a_2, a_3, a_1, a_5, a_4, a_6\}\)

\(\succeq_2: \{a_7, a_1, a_3, a_5, a_2, a_6, a_4\}\)

(i) Round 1: \(p(1) = \emptyset, p(2) = \emptyset, C = \{a_7\}\)

(ii) Round 1: \(p(1) = \{a_2\}, p(2) = \{a_1\}, C = \{a_7\}\)

(iii) Round 2: \(p(1) = \{a_2, a_3\}, p(2) = \{a_1, a_5\}, C = \{a_7\}\)

(iv) Round 3: \(p(1) = \{a_2, a_3, a_4\}, p(2) = \{a_1, a_5, a_6\}, C = \{a_7\}\)

Proposition 1 GAL runs in \(O(m^2)\) time and returns a unique assignment.
In each round, either one object each is allocated to the agents or one contested object is sent to \( C \). If each agent has a different highest priority unallocated object, then the allocation takes constant time. Otherwise, the agents have the same highest priority contested object \( o^* \). In this case, we need to make at most two checks for whether there exists an EF partial assignment that allocated \( o^* \) to one of the agents. In either of these checks, we simply need to verify whether the given partial assignment is EF or not which takes time \( O(m) \) according to Lemma 2. Thus, GAL takes time \( O(m^2) \).

**Proposition 2** GAL returns a maximal EF assignment.

**Proof** The GAL outcome is EF. This follows from the way the partial assignments are constructed so that EF is maintained.

We now show that the outcome is a maximal EF assignment. Assume for contradiction that GAL’s outcome \( p \) is not maximal EF. This means that for some object \( o \in C \) there exists an assignment \( q \) that matches the objects matched by \( p \) as well as \( o \) and possible other objects. But there was a stage in Algorithm 1 where \( o \) was sent to \( C \). If \( o \) was given to agent \(-i\), then agent \( i \) was given the next highest priority object \( o' \) according to \( >i \) which still leads to infeasibility of EF. Clearly \( o > o' \) or else the partial assignment \( p \) at the stage wouldn’t fail EF. For every other unallocated object \( o'' \) at that stage, it holds that \( o' \gtrsim_i o'' \). Hence no object \( o'' \) can be given to agent \( i \) while \( o \) is given to \(-i\) so that \( p \) is still EF.

**Proposition 3** For strict preferences, GAL returns an AL outcome.

**Proof** For strict preferences, there exists a unique priority order irrespective of any lexicographical tie-breaking order. If both agents have different most preferred (equivalent to highest priority since the preferences are strict) unallocated objects, then both GAL and AL behave in the same manner. In case there is a contested object \( o^* \), both agents have a feasibility check in which one of the agents is given \( o^* \) and the other agent is given the next most preferred object (equivalent to highest priority since the preferences are strict). Note that the feasibility check in the case of AL is equivalent to checking whether the tentative new assignment is EF.

Next we show that the GAL outcome is LPO. Unlike in [Brams et al., 2014a], we cannot use the characterization of [Brams and King, 2005] that if agents have strict preferences, any assignment as a result of sequential allocation is Pareto optimal. Hence we need a lemma.

Let \((N, O, \succeq)\) be an assignment problem and \( p \) be a discrete assignment. We will create an auxiliary assignment problem and assignment where each agent is allocated exactly one object (see e.g., [Aziz et al., 2014b]). The clones of an agent \( i \in N \) are the agents in \( N'_i = \{ o : o \in O \text{ and } o \in p(i) \} \). The cloned assignment problem of \((N, O, \succeq)\) is \((N', O', \succeq')\) such that \( N' = \bigcup_{i \in N} N'_i \), and for each \( i_0 \in N', \succeq'_{i_0} = \succeq_i \). The cloned assignment of \( p \) is the discrete assignment \( p' \) in which \( o \in p'(i_o) \) if \( o \in p(i) \) and \( o \not\in p'(i_o) \) otherwise. A cloned assignment can easily be transformed back into the original assignment where each agent \( i \in N \) is allocated all the objects assigned by \( p' \) to the clones of \( i \).
Lemma 3 An assignment for two agents is Pareto optimal over the set of allocated objects if there exist no objects \( o, o' \) such that \( o \succ_i o' \) and \( o' \succeq_{-i} o \).

Proof By (Lemma 2, Aziz et al., 2014b), an assignment is Pareto optimal if and only if its cloned assignment is Pareto optimal for the cloned assignment problem. Hence, we can restrict our attention to the cloned assignment and the cloned assignment setting. If the cloned assignment is Pareto optimal, the original assignment is Pareto optimal. If the cloned assignment is not Pareto optimal, then there exists a ‘trading cycle’ in which each object points to its owner, each cloned agent in the cycle points to an object that is at least as preferred as its own object and at least one agent in the cycle points to a strictly more preferred object than the one it owns (Aziz and de Keijzer, 2012).

Firstly, we claim that there exists no trading cycle consisting only of clones of one agent. Assume for contradiction that there exist a trading cycle consisting of only clones of the same agent. Then there exists at least one object that is minimally preferred. The agent who points to this object also owns a minimally preferred object. Hence each agent owns a minimally preferred object and thus the cycle is not Pareto improving.

We now show that, if there exists a trading cycle, then there exists one which alternates between clones of the two agents. Consider any cycle which has the following path consisting of multiple clones of the same agent in succession:

\[
o_{c_1} \rightarrow i_{c_1} \rightarrow o_{c_2} \rightarrow i_{c_2} \rightarrow \cdots \rightarrow o_{c_k} \rightarrow i_{c_k} \rightarrow o_{c_{k+1}} \rightarrow -i_{c_{k+1}}.\]

Since clones of each agent \( i \) have the identical preference, \( i_{c_k} \) also points directly to \( o_{c_{k+1}} \). Hence, we know that there is also a path

\[
o_{c_k} \rightarrow i_{c_k} \rightarrow o_{c_{k+1}} \rightarrow -i_{c_{k+1}}.\]

We now show that if there exists a trading cycle which alternates between clones of both agents, then there exists one with exactly one clone of each agent. Assume that a clone of the agent \( i \) gets a strictly more preferred object in the trading cycle. Consider the clone \( i_o \) of agent \( i \) who has the least preferred object among all clones of \( i \). If there are multiple such clones of \( i \), we arbitrarily pick any such clone of \( i \). This means that \( i_o \) points to all the objects in the cycle that are owned by clones of \( -i \). Among the clones of \( -i \), there is one clone \( -i_o' \) that points to \( o \). Hence we have the following cycle:

\[
o \rightarrow i_o \rightarrow o' \rightarrow -i_{o'} \rightarrow o.\]

Thus, there exists a trading cycle consisting of two cloned agent nodes each corresponding to the two agents and two object nodes. This means that there exist \( o, o' \in O \) such that \( o \succ i \) and \( o' \succeq_{-i} o \).

Proposition 4 The GAL outcome is LPO.

Proof Let us constrain ourselves to the set of objects \( O' \subseteq O \) that are allocated to agents 1 and 2. Now let \((N', O', \succeq')\) be the cloned assignment problem. Then assignment \( p \) for objects in \( O' \) is PO iff the corresponding assignment is PO for \((N', O', \succeq')\). Now assume that the GAL outcome is no LPO. Then the assignment with respect to
$O'$ is not PO. By Lemma 3, there exists $i \in \{1, 2\}$ such that $i$ gets $o$ in some round $t$, $o' >_i o$ where $o'$ was allocated to $-i$ and $o \succeq_{-i} o'$. This means that $o'$ was allocated to $i$ in round $t' \leq t$. Now if $o >_{-i} o'$, then $o$ would be a higher priority object for $-i$ so that it would not have gone for $o'$ before $o$. Then it must be that $o \sim_{-i} o'$. But, if $o \sim_{-i} o'$, then $o$ would again be a higher priority object for $-i$ so that it would not have gone for $o'$ before $o$. Hence a contradiction. □

In Proposition 4 we showed that there exists no other (not necessarily EF) assignment that uses the same objects as the GAL outcome and is Pareto improvement over the GAL outcome. Next we show that there exists no other EF assignment that may use any objects and is a Pareto improvement over the GAL outcome.

**Proposition 5** GAL returns an assignment such that there exists no other assignment that Pareto dominates it and is still envy-free.

**Proof** Assume for contradiction that GAL’s outcome $p$ is SD-dominated by another EF assignment $q$ such that $q(i) \succeq_{SD} p(i)$ for both $i$ and $q(i) >_{SD} p(i)$ for at least one $i$. We now proceed in rounds where in each round we check the highest priority object allocated to each of the two agents that have not been checked. We check the partial assignments $p'$ and $q'$ in each round $t$ to see whether $q'(i) \succeq_{SD} p'(i)$ and $q'(-i) \succeq_{SD} p'(-i)$. If both $q'(i) >_{SD} p'(i)$ and $q'(-i) >_{SD} p'(-i)$, then it means that both get higher priority objects, than $p$ in that round. This is a contradiction as GAL would allocated these higher priority objects to the agents. Now assume that $q'(i) >_{SD} p'(i)$ and $q'(-i) \sim_{SD} p'(-i)$. This means that agent $-i$ gets the same object and the other agent $i$ gets a higher priority object. But this is again a contradiction, because GAL would have allocated the more preferred object to $i$ in that round. □

As a corollary, we also obtain the same statement for the simplified version of AL.

Note that for the case of two agents, Aziz et al. (2014a) presented a polynomial time algorithm to check whether a complete SD-envy-free assignment exists or not. In order to compute a maximum size SD envy-free assignment, one can consider different subsets $O' \subset O$ and check whether a complete SD-envy-free assignment exists or not for $O'$. However this approach would require checking exponential number of subsets.

**5 Discussion**

In this paper, we presented GAL that is a generalization of the AL method of Brams et al. (2014a) for the fair allocation of indivisible objects among two agents. A crucial advantage of extending AL to GAL is for the case when agents have identical preferences. If agents have strict and identical preferences, then AL assigns all the objects to the contested pile. However if the preferences are really coarse, such as when all objects are equally preferred, then GAL assigns $\lfloor m/2 \rfloor$ to each agent.

GAL can also be used as an algorithm to solve previously studied problems within fair division:
A Generalization of the AL method for Fair Allocation of Indivisible Objects

**Theorem 2** There exists a $O(m^2)$ algorithm to check whether there exists a complete assignment that is EF.

*Proof* By Proposition 2 if there exists a complete EF assignment, GAL returns such an assignment. □

Previous algorithms to solve this problem take time $O(m^3)$ and require solving network flows or maximum matching (Bouveret et al., 2010; Aziz et al., 2014a).

GAL is specifically designed for the case of two agents. This raises the question whether there exists a polynomial-time algorithm that can compute a maximal EF assignment when the number of agents is not constant. The answer to this question is negative unless P=NP.

**Theorem 3** Unless $P = NP$, there exists no polynomial-time algorithm that computes a maximal EF assignment even if preferences are dichotomous or strict.

*Proof* Bouveret et al. (2010) proved that checking whether there exists a complete EF assignment is NP-complete for strict preferences. Aziz et al. (2014a) proved that checking whether there exists a complete EF assignment is NP-complete for dichotomous preferences.

If there was a polynomial-time algorithm to compute a maximal EF assignment, one can check whether the assignment is complete or not. It is complete iff there exists a complete EF assignment. Hence a polynomial-time algorithm to compute a maximal EF assignment can solve an NP-complete problem in polynomial time. □

GAL can also be seen as a discrete version of the probabilistic serial (PS) algorithm (Bogomolnaia and Moulin, 2001; Katta and Sethuraman, 2006) for computing a fractional assignment. PS is SD-efficient and SD-envy-free. In other words it returns a maximal fractional assignment that is both SD-efficient and SD-envy-free. In the randomized setting, there is always complete assignment that satisfies both properties. Similarly, a GAL outcome is a maximal discrete assignment that is both SD-efficient and SD-envy-free. If we restrict ourselves to discrete assignments, then there may not exist a complete and envy-free assignment.

It will be interesting to apply the approach of maximal EF to weaker notions of fairness. Finally, extending GAL to the case of constant number of agents is left as future work.

**Acknowledgments**

The author thanks Steven Brams for sharing the paper on the AL method with him. He also appreciates Sajid Aziz, Steven Brams and Christian Klamler for their useful feedback. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

---

5 There is a trivial exponential-time extension of GAL for the case of arbitrary number of agents: enumerate all partial assignments and check which ones are EF.
References

D. J. Abraham, K. Cechlárová, D. Manlove, and K. Mehlhorn. Pareto optimality in house allocation problems. In Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC), volume 3341 of Lecture Notes in Computer Science (LNCS), pages 1163–1175, 2005.

H. Aziz. A note on the undercut procedure. In Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pages 1361–1362, 2014.

H. Aziz and B. de Keijzer. Housing markets with indifferences: a tale of two mechanisms. In Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI), pages 1249–1255, 2012.

H. Aziz, F. Brandt, and M. Brill. On the tradeoff between economic efficiency and strategyproofness in randomized social choice. In Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pages 455–462. IFAAMAS, 2013a.

H. Aziz, F. Brandt, and P. Harrenstein. Pareto optimality in coalition formation. Games and Economic Behavior, 82:562–581, 2013b.

H. Aziz, S. Gaspers, S. Mackenzie, and T. Walsh. Fair assignment of indivisible objects under ordinal preferences. In Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pages 1305–1312, 2014a.

H. Aziz, S. Gaspers, S. Mackenzie, and T. Walsh. Fair assignment of indivisible objects under ordinal preferences. Technical Report 1312.6546, arXiv.org, 2014b.

A. Bogomolnaia and H. Moulin. A new solution to the random assignment problem. Journal of Economic Theory, 100(2):295–328, 2001.

S. Bouveret, U. Endriss, and J. Lang. Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods. In Proceedings of the 19th European Conference on Artificial Intelligence (ECAI), pages 387–392, 2010.

S. J. Brams. Dispute Over Divorce or Inheritance? Try This ‘Envy-Free’ Algorithm. Huffington Post, 2014a.

S. J. Brams. Dividing the indivisible. Plus Magazine, 2014b.

S. J. Brams and P. C. Fishburn. Fair division of indivisible items between two people with identical preferences: Envy-freeness, Pareto-optimality, and equity. Social Choice and Welfare, 17:247–267, 2000.

S. J. Brams and T. R. Kaplan. Dividing the indivisible procedures for allocating cabinet ministries to political parties in a parliamentary system. Journal of Theoretical Politics, 16(2):143–173, 2004.

S. J. Brams and D. L. King. Efficient fair division: Help the worst off or avoid envy? Rationality and Society, 17(4):387–421, 2005.

S. J. Brams, P. Edelman, and P. C. Fishburn. Fair division of indivisible items. Theory and Decision, 55:147–180, 2003.

S. J. Brams, D. M. Kilgour, and C. Klamler. The undercut procedure: an algorithm for the envy-free division of indivisible items. Social Choice and Welfare, 39:615–631, 2012.
S. J. Brams, D. M. Kilgour, and C. Klamler. Two-person fair division of indivisible items: An efficient, envy-free algorithm. Notices of the AMS, 61(2):130–141, 2014a.

S. J. Brams, D. M. Kilgour, and C. Klamler. An algorithm for the proportional division of indivisible items. May 2014b.

Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. Informatica, 30:3–31, 2006.

J. Cullinan, S. H. Hsiao, and D. Polett. A borda count for partially ordered ballots. Social Choice and Welfare, 42(4):913–926, 2014.

B. de Keijzer, S. Bouveret, T. Klos, and Y. Zhang. On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In Proceedings of the 1st International Conference on Algorithmic Decision Theory, pages 98–110, 2009.

S. Demko and T. P. Hill. Equitable distribution of indivisible objects. Mathematical Social Sciences, 16:145–158, 1988.

D. Foley. Resource allocation and the public sector. Yale Econ Essays, 7:45—98, 1967.

L. Fortnow. The Golden Ticket: P, NP, and the Search for the Impossible. Princeton University Press, 2013.

P. Gärdenfors. Assignment problem based on ordinal preferences. Management Science, 20:331–340, 1973.

A-K. Katta and J. Sethuraman. A solution to the random assignment problem on the full preference domain. Journal of Economic Theory, 131(1):231–250, 2006.

R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In Proceedings of the 5th ACM Conference on Electronic Commerce (ACM-EC), pages 125–131, 2004.

D. Manlove. Algorithmics of Matching Under Preferences. World Scientific Publishing Company, 2013.

H. Moulin. Fair Division and Collective Welfare. The MIT Press, 2003.

A. D. Procaccia and J. Wang. Fair enough: Guaranteeing approximate maximin shares. In Proceedings of the 15th ACM Conference on Economics and Computation (ACM-EC), pages 675–692, 2014.

K. Pruhs and G. J. Woeginger. Divorcing made easy. In E. Kranakis, D. Krizanc, and F. Luccio, editors, Proceedings of FUN, number 7288 in Lecture Notes in Computer Science (LNCS), pages 305–314, 2012.

E. Ronn. NP-complete stable matching problems. Journal of Algorithms, 11(2): 285–304, 1990.

D. Saban and J. Sethuraman. House allocation with indifferences: a generalization and a unified view. In Proceedings of the 14th ACM Conference on Electronic Commerce (ACM-EC), pages 803–820, 2013.

L. Wilson. Assignment using choice lists. Operations Research Quarterly, 28(3): 569—578, 1977.

H. P. Young. Dividing the indivisible. American Behavioral Scientist, 38:904–920, 1995.