

Object Allocation via Swaps along a Social Network

Laurent Gourvès, Julien Lesca, Anaelle Wilczynski

▶ **To cite this version:**

Laurent Gourvès, Julien Lesca, Anaelle Wilczynski. Object Allocation via Swaps along a Social Network. 26th International Joint Conference on Artificial Intelligence (IJCAI'17), Aug 2017, Melbourne, Australia. pp.213-219, 10.24963/ijcai.2017/31 . hal-01741519

HAL Id: hal-01741519
https://hal.science/hal-01741519

Submitted on 30 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Object Allocation via Swaps along a Social Network

Laurent Gourvès, Julien Lesca and **Anaëlle Wilczynski**
Univ. Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France
{laurent.gourves, julien.lesca, anaelle.wilczynski}@dauphine.fr

Abstract

This article deals with the allocation of objects where each agent receives a single item. Starting from an initial endowment, the agents can be better off by exchanging their objects. However, not all trades are likely because some participants are unable to communicate. By considering that the agents are embedded in a social network, we propose to study the possible allocations emerging from a sequence of simple swaps between pairs of neighbors in the network. This model raises natural questions regarding (*i*) the reachability of a given full allocation, (*ii*) the ability of an agent to obtain a given object, and (*iii*) the search of Pareto-efficient allocations. We investigate the complexity of these problems by providing, according to the structure of the social network, polynomial and NP-complete cases.

1 Introduction

Allocating indivisible resources to agents is a fundamental problem in AI [Chevaleyre *et al.*, 2006], which lies at the interface of computer science and economics. When the set of solutions is restricted to the allocations where each agent receives a single resource, this problem corresponds to a specific type of matching under preferences [Manlove, 2013; Klaus *et al.*, 2016], called *one-sided matching*. The specificity of the one-sided matching lies on the fact that the resources have no preferences over the possible allocations, on the contrary of two-sided matching. One-sided matchings have been widely studied under different names: assignment problem [Gardenfors, 1973], house allocation [Abdulkadiroğlu and Sönmez, 1998]. When each agent is initially endowed with a resource, the problem is known in the literature as the housing market problem [Shapley and Scarf, 1974]. Starting from the initial endowment and performing a sequence of exchanges among the agents is a standard approach to reallocate the resources. The *top-trading cycle* algorithm and its variants [Shapley and Scarf, 1974; Abdulkadiroğlu and Sönmez, 1999; Aziz and De Keijzer, 2012] exploit this idea in order to compute a better allocation. Another way to grasp the problem is by letting the agents perform the exchanges by themselves. A rich literature has been developed on this approach, formalizing conditions for realistic trades and analyzing the quality of the possible outcomes [Sandholm, 1998; Dunne *et al.*, 2005; Chevaleyre *et al.*, 2005; Endriss *et al.*, 2006; Chevaleyre *et al.*, 2007; Dunne and Chevaleyre, 2008; Chevaleyre *et al.*, 2008; Aziz *et al.*, 2016]. Most part of this literature is devoted to general resource allocations where agents typically receive bundles of resources, but only a few articles specifically address the housing market problem [Damamme *et al.*, 2015].

In the housing market setting, it is implicitly assumed that all the agents have the capacity to perform direct deals. This assumption is unlikely in large scale instances, where some agents are unable to communicate. Restricting the set of direct exchanges to the ones which are actually possible seems more realistic and relevant in large scale instances. The ability of agents to exchange resources can be modeled as a social network [Jackson, 2008; Easley and Kleinberg, 2010]. Recent works on matching have investigated the use of a social network in order to capture other types of social behavior such as altruism [Anshelevich *et al.*, 2013], social contact [Arcaute and Vassilvitskii, 2009], peer effects [Bodine-Baron *et al.*, 2011] or collaboration [Hoefer, 2013]. Most of these works focus on two-sided matchings and, up to our knowledge, only a few articles are devoted to resource allocation. In this line, the notion of *negotiation topology graph* [Chevaleyre *et al.*, 2007], where the exchanges are restricted to agents belonging to the same clique of the social network, is noteworthy.

This article deals with a variant of housing market, where the agents are embedded in a social network which determines their ability to exchange their objects. Each participant is initially endowed with a single object, and she has strict ordinal preferences over objects. The agents may exchange their items under two conditions: they find it mutually profitable, and they are socially tied. Though sophisticated exchanges involving multiple agents have been analyzed [Sandholm, 1998; Dunne and Chevaleyre, 2008; Damamme *et al.*, 2015], we focus on simple trades between pairs of neighbors.

The exchanges are made without payments or monetary compensations. Our main question is, starting from the initial endowment, which allocation of the objects can emerge? Indeed, some solutions are ruled out because of the agents' preferences over the objects (e.g. no one is interested in exchanging her current object with something that she considers as worse). In addition, the network limits the access of certain participants to each other. Therefore, it is particularly challenging to understand how the combination of these two natural ingredients influence the outcome of a dynamics in which connected agents agree on mutually profitable swaps of objects. Moreover, it appears interesting to consider the quality of a resulting allocation by sequence of profitable exchanges. In the context of ordinal preferences, Pareto-efficiency appears as the minimal requirement for an allocation to be socially acceptable. Pareto-efficiency has been widely studied in the context of house allocation [Abraham *et al.*, 2005] and housing market [Aziz *et al.*, 2016]. As far as we know, the computation of Pareto-efficient allocations has not been investigated when the possible allocations are constrained by a social network.

The paper is organized as follows. The model of object allocation along a social network is defined in Section 2, as well as the problems under consideration. Then, in Section 3, we address the problem of reachability for an object and in Section 4 the reachability of a given full allocation. Before concluding, the search of Pareto-efficient allocations is investigated in Section 5.

2 Model

2.1 Notations

Let $N = \{1, \dots n\}$ be a set of agents and $X = \{x_1, \dots, x_n\}$ be a set of objects such that $|N| = |X| = n$. An allocation is a bijection $\sigma : N \to X$, where $\sigma(i)$ denotes the object assigned to agent i in σ. We may write σ as an n-tuple $\sigma = (\sigma(1), \dots, \sigma(n))$. Each agent is initially endowed with an object; σ_0 denotes this initial allocation. We assume w.l.o.g. that $\sigma_0(i) = x_i$ for every agent i. Each agent $i \in N$ has strict preferences over the objects, represented by a strict linear order $\succ_i$ over X. The agents' preference profile is denoted by $\succ$.

Let $G = (N, E)$ be an undirected graph representing a social network over the agents, where the edges capture the possibility of communication and trade between two agents. An instance is a tuple $(N, X, \succ, G, \sigma_0)$.

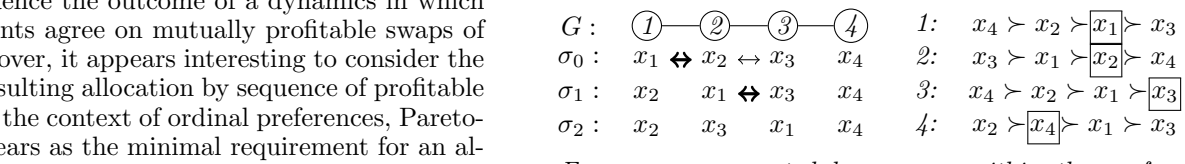
2.2 Swap Dynamics

In this article we focus on *rational trades* between pairs of agents. A trade between two agents is *rational* if both agents benefit by exchanging their objects. An allocation σ' results from a rational trade from σ for agents i and j if $\sigma'(i) = \sigma(j)$, $\sigma'(j) = \sigma(i)$, $\sigma'(i) \succ_i \sigma(i)$ and $\sigma'(j) \succ_j \sigma(j)$, and for any agent $k \notin \{i, j\}$, $\sigma'(k) = \sigma(k)$.

We assume that every trade is necessarily performed between two neighbors in the network. Rational trades defined according to G are called *swaps*. An assignment σ' results from a swap from σ w.r.t. $G = (N, E)$ if there exist two agents i and j such that $(i, j) \in E$ and σ' is a rational trade from σ for i and j.

A sequence of swaps is a sequence of assignments $(\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_t)$ such that for any $k \in \{1, \dots, t\}$, σ_k results from a swap from σ_{k-1}. An allocation σ is *stable* if no swap can be performed from σ.

Example 1 *Consider an instance where $n = 4$. The network, the preferences and a sequence of swaps are described below.*



From σ_0, represented by squares within the preferences, agents 1 and 2 can perform a swap: they are linked and they both prefer the object of the other. The same holds for agents 2 and 3. If agents 1 and 2 exchange their objects (bold arrow in the figure), we obtain allocation σ_1. From σ_1, only agents 2 and 3 can perform a swap, leading to allocation σ_2. In σ_2, agents 1 and 4 prefer the object of the other but they are not connected in G, thus this exchange is not feasible. No swap is possible in σ_2, hence σ_2 is stable.

An assignment σ' is *reachable* if there exists a sequence of swaps $(\sigma_0, \dots, \sigma_t)$ such that $\sigma_t = \sigma'$. Let us denote by RA the set of all reachable assignments. By extension, an object $x \in X$ is *reachable* for an agent $i \in N$ if there is a sequence of swaps $(\sigma_0, \dots, \sigma_t)$ such that $\sigma_t(i) = x$.

Swap dynamics is a distributed process where, starting from σ_0, the agents exchange their objects without any external intervention, until a stable allocation is reached.

2.3 Issues

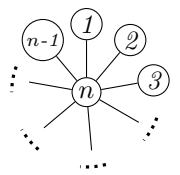
In order to analyze the distributed process of swap dynamics, we are interested in all the allocations obtained from σ_0 by sequences of swaps. The following two decision problems naturally arise from our model.

Reachable Object	
Instance:	$(N, X, \succ, G, \sigma_0)$, agent i, object x
Question:	Is x reachable for i?

Reachable Assignment	
Instance:	$(N, X, \succ, G, \sigma_0)$, assignment σ
Question:	Is σ reachable?

The swaps dynamics always converges to a stable allocation. However, the following example shows that we can reach a rather bad assignment if the agents exchange their objects in an uncoordinated way.

Example 2 *Consider an instance with n agents. The network and the preferences are described below.*



1: $x_n \succ [\ldots] \succ \boxed{x_1}$
2: $x_1 \succ [\ldots] \succ \boxed{x_2}$
$\vdots$ $\vdots$
n-2: $x_{n-3} \succ [\ldots] \succ \boxed{x_{n-2}}$
n-1: $x_{n-2} \succ [\ldots] \succ x_n \succ \boxed{x_{n-1}}$
n: $x_{n-1} \succ x_{n-2} \succ [\ldots] \succ x_1 \succ \boxed{x_n}$

Consider allocation σ_1 resulting from the swap between agents n-1 and n. The center agent n obtains her most preferred object, thus no further swap is possible and σ_1 is stable. However, if we consider the sequence of swaps performed by the pairs of agents (n,1),(n,2),. . . ,(n,n-2), (n,n-1), then we reach an allocation σ' where every agent obtains her most preferred object.

An allocation σ is *Pareto-efficient* if there is no assignment σ' such that for all $i \in N$, $\sigma'(i) \succeq_i \sigma(i)$ and there exists $j \in N$ such that $\sigma'(j) \succ_j \sigma(j)$. We restrict the definition of Pareto-efficiency within the set RA of reachable allocations, otherwise a Pareto-efficient allocation in the standard meaning may not be reachable.

As informally noticed in Example 2, swap dynamics can reach an allocation that is Pareto dominated. This observation also leads us to consider the swap dynamics under a centralized perspective. The swaps, though constrained by the social network and the mutual benefit for the two involved agents, are guided by a coordinator, in order to reach a Pareto-efficient allocation among all reachable allocations. Consequently, we also study the following optimization problem.

Pareto
Instance: $(N, X, \succ, G, \sigma_0)$
Goal: Find a Pareto-efficient allocation σ within RA

3 Reachable Object

In this section we focus on Reachable Object. We prove that the problem is NP-complete even when the network is a tree. However, for some further restrictions on the graph, the problem becomes polynomial.

First observe that in any network, an object cannot pass twice by the same agent. This is derived from the definition of a swap where both agents are better off.

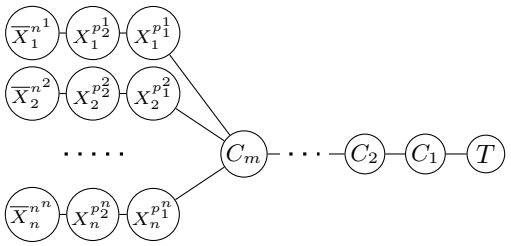
Observation 1 *If for a reachable assignment σ and an agent i, $\sigma(i) \neq \sigma_0(i)$ then in any assignment σ' reachable from σ, $\sigma'(i) \neq \sigma_0(i)$.*

Theorem 1 Reachable Object *is NP-complete even when the network is a tree.*

Sketch of proof: One can easily verify that the problem is in NP. We use a reduction from the NP-complete problem 2P1N-SAT [Yoshinaka, 2005]. In 2P1N-SAT, we are given a set $V = \{v_1, \ldots, v_n\}$ of variables, and a collection $\mathcal{C} = \{C_1, \ldots, C_m\}$ of clauses over V such that each positive (resp., negative) literal occurs exactly 2 (resp., 1) times in $\mathcal{C}$. Is $\mathcal{C}$ satisfiable?

Let v_k^i (resp., $\overline{v}_k^i$) denote the positive (resp., negative) literal v_k if present in C_i. Index p_j^k (resp., n^k) refers to the clause containing the j^{th} occurrence ($j \in \{1, 2\}$) of v_k (resp., the occurrence of $\overline{v}_k$).

We construct an instance $(N, X, \succ, G, \sigma_0)$ of Reachable Object as follows. Each literal v_k^i (resp., $\overline{v}_k^i$) is associated with an agent X_k^i (resp., $\overline{X}_k^i$) which is initially endowed with object x_k^i (resp., $\overline{x}_k^i$). Every clause C_i is associated with an agent C_i initially endowed with object c_i. We add an agent T initially endowed with object t, leading to $|N| = |X| = m + 3n + 1$. The graph $G = (N, E)$ is as follows.



The preference profile $\succ$ is defined below. We only represent the objects that each agent prefers to her initial one, and $\{\ell_i\}$ denotes the set of the objects associated with the literals of clause C_i, which are ranked in arbitrary order.

T: $\{\ell_1\} \succ \boxed{t}$
C_i: $\{\ell_{i+1}\} \succ t \succ \{\ell_i\} \succ c_1 \succ \{\ell_{i-1}\} \succ \ldots \succ c_{i-1} \succ \{\ell_1\} \succ \boxed{c_i}$
C_m: $t \succ \{\ell_m\} \succ c_1 \succ \{\ell_{m-1}\} \succ \ldots \succ c_{m-1} \succ \{\ell_1\} \succ \boxed{c_m}$
$X_k^{p_1^k}$: $c_{m-n^k+1} \succ c_{m-p_2^k+1} \succ x_k^{p_2^k} \succ c_{m-p_1^k+1} \succ \overline{x}_k^{n^k} \succ \boxed{x_k^{p_1^k}}$
$X_k^{p_2^k}$: $c_{m-p_1^k+1} \succ x_k^{p_1^k} \succ \overline{x}_k^{n^k} \succ \boxed{x_k^{p_2^k}}$
$\overline{X}_k^{n^k}$: $x_k^{p_2^k} \succ \boxed{\overline{x}_k^{n^k}}$

Now, we have an instance of Reachable Object and we claim that $\mathcal{C}$ is satisfiable iff t is reachable for C_m. The details of the equivalence are omitted.

Observe, on one hand, that in the dipath τ from T to C_m, every agent prefers t to her initial object, and only accepts to exchange t with an object of her successor in τ, corresponding to a literal that satisfies the clause associated with the successor. Therefore, the only way to move t from T to C_m is to give to each agent C_i an object associated with a literal that satisfies the clause C_i. Thus, for $i = 1$ to m, an object of $\{\ell_i\}$ moves to C_i.

On the other hand, we ensure in each branch corresponding to a variable v_k, that if an object x_k^i (resp., $\overline{x}_k^j$) moves out of the branch, then an object $\overline{x}_k^{i'}$ (resp., $x_k^{j'}$) cannot move out thereafter. Indeed, if $\overline{x}_k^{n^k}$ is the first object to move out of the branch, then agent $X_k^{p_1^k}$ obtains her most preferred object and no other object in this branch can move out. Otherwise, if $x_k^{p_1^k}$ or $x_k^{p_2^k}$ moves first, then agent $X_k^{p_1^k}$ has received an object that she prefers to $\overline{x}_k^{n^k}$, so $\overline{x}_k^{n^k}$ is blocked. ■

The network in the previous proof is a tree where all the subtrees derived from a common root are paths (called *generalized star* thereafter). Nevertheless, Reachable Object is solvable in polynomial time when the network is a star.

Proposition 1 *When G is a star, there exists a polynomial algorithm for* Reachable Object.

Proof: The problem asks whether agent i can get object x. The network consists of a center denoted by n, and $n-1$ leaves denoted by $1, \ldots, n-1$. A swap always involves the center and a leaf. Once a leaf has exchanged her initial object, she is not involved in a subsequent swap (Observation 1). Thus, any sequence of swaps reduces to an ordered list (without repetition) of leaves, indicating with which agents the center exchanges her object.

Let us first focus on the case where agent i is the center. The problem reduces to the search of a path in a digraph $G_D = (N, E')$ where $(a, b) \in E'$ with $a \in N$ and $b \in N \setminus \{n\}$ iff the center and leaf b can rationally trade when the center owns the initial object of a while b owns her initial object. There is a path from n to b in the digraph iff the center can get the initial object of agent b. A linear algorithm solves this path problem and the construction of the digraph is polynomial.

In case i is a leaf, the problem reduces to the previous one: the center gets x and then, i and n swap their objects. Return "yes" if these two steps are feasible. ■

Now we suppose that the network is a path. Each time G is a path, we assume that $E = \{(j, j+1) : 1 \leq j < n\}$. Observation 1 implies that when G is a path, once an object "moves" in a given direction, then it cannot "move" in the opposite direction.

Let us define as *canonical sequence of exchanges* $\kappa(j, k)$ the sequence of exchanges which assigns object x_j to agent k by directly moving it along the dipath from j to k. This is the sequence of exchanges between the following pairs of agents if $j < k$: $(j, j+1), (j+1, j+2), \ldots, (k-1, k)$. It transforms σ_0 into an assignment σ where $\sigma(\ell) = x_\ell$ if $\ell < j$, $\sigma(\ell) = x_{\ell+1}$ if $j \leq \ell < k$ and $\sigma(k) = x_j$. This sequence is said to be a sequence of swaps if all its exchanges are rational.

Proposition 2 *When G is a path, if object x_j is reachable for agent n, then $\kappa(j, n)$ is a sequence of swaps.*

Proof: Assume by contradiction that there exists $k \in \{j+1, \ldots, n\}$ such that the exchange between $k-1$ and k is not rational i.e., $x_k \succ_k x_j$ or $x_j \succ_{k-1} x_k$.

Assume first that $x_k \succ_k x_j$ holds. Note that if there exists a sequence of swaps leading to assign x_j to n, then x_j must be assigned once to agent k since the path from j to n is unique. Because $x_k \succ_k x_j$, agent k will never accept x_j, contradiction.

Assume now that $x_j \succ_{k-1} x_k$ holds. If there exists a sequence of swaps leading to assign x_j to n, then x_j must be exchanged once with x_k, because x_j must reach n and there is no agent after n to receive x_k. This swap cannot be performed between $k-1$ and k since $x_j \succ_{k-1} x_k$, and thus occurs between some agents before k. Consequently, agents $k-1$ and k must have performed an earlier swap in order to make x_k moving to the agents with lower indices than k. After this swap, the object currently owned by $k-1$ must move to the direction of n before x_j, and x_j cannot overtake this object, contradiction. ■

For testing that an object is reachable for a leaf of the path, it suffices to verify that the associated canonical sequence of exchanges is a sequence of swaps.

Corollary 1 *When G is a path and agent i a leaf of G,* Reachable Object *is solvable in polynomial time.*

Other solvable cases of Reachable Object in a path can be listed. The main one is when the distance between the agent and the object is a constant. We omit the details due to the space limitation but they essentially rely on the following observations. If agent i tries to get object x_j, with $i < j$, then we can ignore agent k and object x_k, for $k > j$. If one can guess which set $N' \subset N$ of agents eventually get the objects of $\{x_k : i \leq k < j\}$, then we can deduce the final allocation: for two agents $a, b \in N'$ such that $a < b$, the object received by agent a must have a smaller index than the object received by agent b. It suffices to test whether the final allocation is reachable, a problem that is solved in polynomial time in the next section (Proposition 3).

Despite its apparent simplicity, Reachable Object in a path is a challenging open problem when no restriction on the agent's location is made. We believe that this case is at the frontier of tractability.

4 Reachable Assignment

In this section, we address the decision problem Reachable Assignment: does σ belong to RA? We prove that this problem is NP-complete in general but polynomial when the network is a tree.

Theorem 2 Reachable Assignment *is NP-complete.*

Proof: One can easily verify that the problem is in NP. We propose a reduction from Reachable Object. Take an instance $\mathcal{I} = (N, X, G, \succ, \sigma_0)$. The problem asks whether an agent (called 1 w.l.o.g.) can reach a given object (called x_ℓ thereafter). We construct an instance of Reachable Assignment $\mathcal{I}' = (N \cup N', X \cup X', G', \succ', \sigma'_0)$ where each element of N (resp., X) has a copy in N' (resp., X'). The copies of j and x_j are denoted by j' and x'_j, respectively. The edge set of G' is a superset of E and each $j' \in N'$ has an edge with every $k \in N' \cup \{j\}$. The initial assignment σ'_0 is such that $\sigma'_0(j) = \sigma_0(j)$ when $j \in N$, and $\sigma'_0(j) = x'_j$ when $j \in N'$. For every $j \in N$, $\succ'_j$ consists of x'_j on top, followed by $\succ_j$, and the remaining objects are put on the last positions. For the agents of N', the preferences $\succ'$ are defined as follows.

$1'$: $x_\ell \succ' \boxed{x'_1}$

ℓ': $x_1 \succ' \ldots \succ' x_n \succ' \boxed{x'_\ell} \succ' x_\ell$

$j' \notin \{1', \ell'\}$: $x_j \succ' x_{j+1} \succ' \ldots \succ' x_n \succ' x_{j-1} \succ' \ldots \succ' x_1 \succ' \boxed{x'_j} \succ' x_\ell$

We claim that x_ℓ is reachable for agent 1 in $\mathcal{I}$ iff every agent gets her most preferred object in $\mathcal{I}'$.

Suppose that x_ℓ is reachable for agent 1 in $\mathcal{I}$. By construction, it is also the case in $\mathcal{I}'$. Once x_ℓ has reached agent 1, every agent $j \in N$ exchanges her object with her

copy $j' \in N'$. Thus, each $j \in N$ possesses her most preferred object x'_j. The same holds for agent $1'$. Then, by starting with ℓ' and afterwards considering every j' by increasing order of indices, j' exchanges her object with the current owner of her most preferred object. Those exchanges are swaps by construction.

Suppose that every agent can get her most preferred object in $\mathcal{I}'$. Agent $1'$ must receive x_ℓ after an exchange with agent 1 because no agent in $N' \setminus \{1'\}$ prefers x_ℓ to her initial object. For the same reason, x_ℓ cannot reach $1'$ through the agents of $N' \setminus \{1'\}$. The preferences and the network topology impose that no object of X' can be involved in the move of x_ℓ to agent 1. ■

In the previous reduction, we have constructed a graph which contains cycles. When G is a tree, a polynomial algorithm (Algorithm 1) solves Reachable Assignment.

The global idea of the algorithm is that every object must move along a unique dipath in order to reach its owner in σ from its owner in σ_0. Therefore, it suffices to verify that every dipath intersects another one of the opposite direction within a swap for the involved agents. In the pseudocode, list L stores the first arc of each dipath and $pop(P)$ outputs the first arc of dipath P and deletes this arc from P.

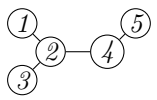
Algorithm 1:

Input: $(N, X, \succ, G, \sigma_0)$, assignment σ
Output: whether σ is reachable from σ_0

$L \leftarrow \emptyset$; $\sigma' \leftarrow \sigma_0$;
for each $x \in X$ **do**
  $P_x \leftarrow$ the unique dipath from the owner of x in σ_0 to the owner of x in σ;
  $L \leftarrow L \cup \{pop(P_x)\}$;
while $L \neq \emptyset$ **do**
  if $\exists_{i,j \in N, i \neq j}: (i,j),(j,i) \in L$ **then**
    if $(\sigma'(i) \succ_i \sigma'(j)) \vee (\sigma'(j) \succ_j \sigma'(i))$ **then**
      return false;
    Update σ' with exchange between i and j;
    $L \leftarrow L \setminus \{(i,j),(j,i)\}$;
    $L \leftarrow L \cup \{pop(P_{\sigma'(i)})\} \cup \{pop(P_{\sigma'(j)})\}$;
  else return false;
return true;

Let us illustrate Algorithm 1 with an example.

Example 3 *Consider an instance with* 5 *agents trying to reach allocation* $(x_4, x_5, x_1, x_3, x_2)$. *The social network and the preferences are as follows.*



1: $x_4 \succ \boxed{x_1}$
2: $x_5 \succ x_3 \succ x_1 \succ x_4 \succ \boxed{x_2}$
3: $x_1 \succ \boxed{x_3}$
4: $x_3 \succ x_5 \succ x_2 \succ \boxed{x_4}$
5: $x_2 \succ \boxed{x_5}$

The dipaths P_x computed at line 3 of Algorithm 1 are:

$P_{x_1} = \{(1,2),(2,3)\}$ $P_{x_3} = \{(3,2),(2,4)\}$ $P_{x_5} = \{(5,4),(4,2)\}$
$P_{x_2} = \{(2,4),(4,5)\}$ $P_{x_4} = \{(4,2),(2,1)\}$

The following table illustrates the differents steps of the while loop (lines 5-12):

L	*Swap*	σ'
$\{(1,2),(2,4),(3,2),(4,2),(5,4)\}$	$2 \leftrightarrow 4$	(x_1,x_4,x_3,x_2,x_5)
$\{(1,2),(4,5),(3,2),(2,1),(5,4)\}$	$1 \leftrightarrow 2$	(x_4,x_1,x_3,x_2,x_5)
$\{(2,3),(4,5),(3,2),\emptyset,(5,4)\}$	$2 \leftrightarrow 3$	(x_4,x_3,x_1,x_2,x_5)
$\{\emptyset,(4,5),(2,4),\emptyset,(5,4)\}$	$4 \leftrightarrow 5$	(x_4,x_3,x_1,x_5,x_2)
$\{\emptyset,\emptyset,(2,4),\emptyset,(4,2)\}$	$2 \leftrightarrow 4$	(x_4,x_5,x_1,x_3,x_2)
$\emptyset$	-	-

At each step L stores the first arc of each P_{x_i}. At step 1, only one exchange is possible: between agents 2 and 4. This exchange being rational, σ' is updated by performing the swap. Arcs $(2,4)$ and $(4,2)$ are removed from L. Arcs $(4,5)$ and $(2,1)$, which are respectively the new first arcs of P_{x_2} and P_{x_4}, are inserted in L. The algorithm stops when L is empty, implying that σ is reached.

Proposition 3 *Algorithm 1 solves* Reachable Assignment *in polynomial time when G is a tree.*

Proof: Consider an instance with a minimum number of agents where σ is reachable but Algorithm 1 returns "false". Let s be a feasible sequence of swaps from σ_0 to σ. The allocation reached by Algorithm 1 is denoted by σ'. Since some objects have not reached their destination in σ', L is non empty at the end of the execution.
Case 1. L contains no pair of opposite arcs. Start from a node x such that $\sigma(x) \neq \sigma'(x)$ and follow the arcs of L. Since G is acyclic, the walk is finite and ends with an arc, say (a,b). Object $\sigma'(a)$ must pass through agent b to reach it owner in σ, but object $\sigma'(b)$ has reached its destination (no arc goes out of b because $\sigma'(b) = \sigma(b)$).

Because a unique path links two nodes in a tree, we know that in s, agent a exchanges $\sigma'(a)$ with an object held by agent b. Suppose this object is assigned in σ' to agent c. It holds that $\sigma'(c) \neq \sigma'(b)$ and $\sigma'(c) \succ_a \sigma'(a)$.

Consider the path τ from a to c in G. Suppose $b \in \tau$. By considering b as the root of G, we observe that a and c belong to two distinct subtrees. Moreover, $\sigma'(c) \neq \sigma(c)$ and $\sigma'(a) \neq \sigma(a)$. Therefore by removing $\sigma'(a)$ or $\sigma'(c)$, we get a smaller counterexample to Algorithm 1, contradiction. Now suppose $b \notin \tau$. Since $\sigma'(c)$ passes through b then a in s, a has been assigned $\sigma'(c)$ before getting $\sigma'(a)$, which implies that $\sigma'(a) \succ_a \sigma'(c)$, contradiction.
Case 2. Two opposite arcs $(a,b),(b,a)$ belong to L but $\sigma'(a) \succ_a \sigma'(b)$. In s, agent a exchanges $\sigma'(a)$ with an object held by agent b. Suppose this object is assigned in σ' to agent c. It holds that $\sigma'(c) \neq \sigma'(b)$ and $\sigma'(c) \succ_a \sigma'(a)$. Since $\sigma'(c)$ passes through (b,a) in s, if a has been assigned $\sigma'(c)$ before getting $\sigma'(a)$, then $\sigma'(a) \succ_a \sigma'(c)$, contradiction. If a has not been assigned $\sigma'(c)$ before getting $\sigma'(a)$, then $\sigma'(c)$, together with $\sigma'(a)$ and $\sigma'(b)$, has not reached its destination. Thus, a smaller counterexample without one of these objects exists. ■

5 Pareto-Efficient Allocations

The question of this section is how to coordinate the swaps in order to reach a Pareto-efficient allocation within RA. Note that a Pareto-efficient allocation is

stable, otherwise at least two agents will benefit from a possible swap. However, the reverse does not hold, as we can see in Example 2 with allocation σ_1 that is stable but Pareto dominated by another reachable allocation.

Proposition 4 PARETO *is NP-hard.*

Proof: The reduction is the same as in the proof of Theorem 2. $\mathcal{I}$ is a yes-instance iff every agent gets her most preferred object in $\mathcal{I}'$ (which must be the unique Pareto-efficient assignment). Therefore, an algorithm computing a Pareto-efficient assignment can be used to recognize a yes-instance of REACHABLE OBJECT. ■

This negative result does not prevent the existence of a polynomial algorithm for constructing a Pareto-efficient allocation in a specific class of instances. The remainder of this section is devoted to the resolution of PARETO in a path and in a star.

A classical algorithm for achieving Pareto-efficiency is *Serial Dictatorship* [Abdulkadiroğlu and Sönmez, 1998]. It ranks the agents in an arbitrary manner, and assigns them in turns their most favorite object within the set of unassigned objects, until each object is assigned. When the social network is a path, one can use this idea to compute a Pareto-efficient allocation within RA.

Algorithm 2:

Input: $(N, X, \succ, G, \sigma_0)$, agent k
Output: an assignment σ
if $k = 1$ **then**
  return σ_0;
$x_i \leftarrow$ best reachable object for k;
$\sigma \leftarrow$ apply $\kappa(i, k)$ on σ_0;
return *Algorithm 2* $((N, X, \succ, G, \sigma), k-1)$*;*

The parameter k of Algorithm 2 designates the dictator who chooses her best reachable object, and $\kappa(i, k)$ is the canonical sequence (see Proposition 2).

Proposition 5 *When G is a path, Algorithm 2 with $k = n$ solves* PARETO *in polynomial time.*

Sketch of proof: The algorithm starts with a leaf of the path, modifies the current allocation in such a way that the leaf agent obtains her best object (Corollary 1 is used), and continues on the subpath from $k-1$ to 1. The proof is by induction on k: when the decision for agent k is made, the partial allocation for agents $\{k+1, \ldots, n\}$ is Pareto-efficient. ■

Now we study the case of a star. The network consists of a center denoted by n, and $n-1$ leaves denoted by $1, \ldots, n-1$ (see Example 2 for an illustration). We can suppose w.l.o.g. that $\forall i < n$, $x_i \succ_n x_{i+1}$. Indeed, if $x_n \succ_n x_j$ for some j then the center will never exchange her object with j, so j keeps her object in any sequence of swaps. The algorithm is simple: for $i = n-1$ down to 1, exchange the objects of n and i if it is rational.

Proposition 6 *When G is a star, there exists a linear time algorithm for* PARETO*.*

Proof: As already mentioned, a leaf who has exchanged her initial object is not involved in a subsequent swap. The algorithm considers the objects by increasing order of preference of the center agent. Suppose by contradiction that the allocation σ returned by the algorithm is Pareto dominated by another reachable allocation σ'. Let us denote by s and s' the sequences of exchanges leading respectively to σ and σ'. The key observation is that any feasible sequence of swaps is performed by decreasing index of the leaves. In the first step for which s and s' differ, the center swaps her object with ℓ and ℓ', respectively. Since $\ell < \ell'$, ℓ and n cannot swap their object in s'. Therefore $\sigma(\ell) \succ_\ell \sigma'(\ell)$, contradiction. ■

It appears interesting to see if PARETO is polynomial time solvable in a generalized star by a combination of the techniques used to solve the cases of paths and stars.

6 Conclusion

We have investigated some natural problems arising when a group of agents exchange their object along a social network. Our results show that, beyond the agents' preferences, the network can widely influence and constrain the possible allocations. In particular, we proved that deciding whether an agent can obtain a given object (REACHABLE OBJECT) is computationally difficult, even if the network is a tree. Nevertheless, an efficient algorithm can determine if a complete allocation (REACHABLE ASSIGNMENT) is reachable in a tree. Concerning simple graph structures like paths, we were able to decide if a leaf agent can acquire a given object. This result can be extended to the case where the distance between a non-leaf agent and the original location of the object is bounded by a constant. We left open the question whether REACHABLE OBJECT can be efficiently solved in a path, without restriction on the agent's location.

In this article, the social quality of reachable allocations has also been studied through the search of Pareto-efficient allocations. This problem is shown difficult in general networks. On the positive side, polynomial algorithms have been presented for paths and stars. It seems challenging to settle the complexity of computing a Pareto-efficient allocation when the network is a tree.

As future works, many additional aspects of the model deserve attention. For example, we have not investigated the impact if some agents adopt a strategic behavior. Reasoning strategically can drive an agent to refuse a profitable deal (e.g. in a lookahead search). It would also be important to study the social welfare of the feasible allocations. Beyond Pareto-efficiency, is it hard to optimize the egalitarian or the utilitarian social welfare? Like in the price of anarchy/stability, how bad a stable outcome can be, compared to an allocation that is not constrained by the social network? Another future direction is to allow more than two agents to exchange their objects along the network and see which allocations emerge. Finally, our work does not assume any restriction on the preference domain (e.g. single peakedness or single crossingness) which may fit well with our model.

References

[Abdulkadiroğlu and Sönmez, 1998] Atila Abdulkadiroğlu and Tayfun Sönmez. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701, 1998.

[Abdulkadiroğlu and Sönmez, 1999] Atila Abdulkadiroğlu and Tayfun Sönmez. House allocation with existing tenants. *Journal of Economic Theory*, 88(2):233–260, 1999.

[Abraham *et al.*, 2005] David J Abraham, Katarína Cechlárová, David F Manlove, and Kurt Mehlhorn. Pareto optimality in house allocation problems. In *Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC)*, pages 1163–1175, 2005.

[Anshelevich *et al.*, 2013] Elliot Anshelevich, Onkar Bhardwaj, and Martin Hoefer. Friendship and stable matching. In *Proceedings of the 21st European Symposium on Algorithms (ESA)*, pages 49–60, 2013.

[Arcaute and Vassilvitskii, 2009] Esteban Arcaute and Sergei Vassilvitskii. Social networks and stable matchings in the job market. In *Proceedings of the 5th International Workshop on Internet and Network Economics (WINE)*, pages 220–231, 2009.

[Aziz and De Keijzer, 2012] Haris Aziz and Bart De Keijzer. Housing markets with indifferences: A tale of two mechanisms. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*, pages 1249–1255, 2012.

[Aziz *et al.*, 2016] Haris Aziz, Páter Biró, Jérôme Lang, Julien Lesca, and Jérôme Monnot. Optimal reallocation under additive and ordinal preferences. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 402–410, 2016.

[Bodine-Baron *et al.*, 2011] Elizabeth Bodine-Baron, Christina Lee, Anthony Chong, Babak Hassibi, and Adam Wierman. Peer effects and stability in matching markets. In *Proceedings of the 4th International Symposium on Algorithmic Game Theory (SAGT)*, pages 117–129, 2011.

[Chevaleyre *et al.*, 2005] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. Negotiating over small bundles of resources. In *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 296–302, 2005.

[Chevaleyre *et al.*, 2006] Yann Chevaleyre, Paul Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaître, Nicolas Maudet, Julian Padget, Steve Phelps, Juan A Rodrıguez-Aguilar, and Paulo Sousa. Issues in multiagent resource allocation. *Informatica*, 30(1):3–31, 2006.

[Chevaleyre *et al.*, 2007] Yann Chevaleyre, Ulle Endriss, and Nicolas Maudet. Allocating goods on a graph to eliminate envy. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI)*, pages 700–705, 2007.

[Chevaleyre *et al.*, 2008] Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, and Nicolas Maudet. Multiagent resource allocation in k-additive domains: Preference representation and complexity. *Annals of Operations Research*, 163(1):49–62, 2008.

[Damamme *et al.*, 2015] Anastasia Damamme, Aurélie Beynier, Yann Chevaleyre, and Nicolas Maudet. The power of swap deals in distributed resource allocation. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 625–633, 2015.

[Dunne and Chevaleyre, 2008] Paul E Dunne and Yann Chevaleyre. The complexity of deciding reachability properties of distributed negotiation schemes. *Theoretical Computer Science*, 396(1-3):113–144, 2008.

[Dunne *et al.*, 2005] Paul E Dunne, Michael Wooldridge, and Michael Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 164(1-2):23–46, 2005.

[Easley and Kleinberg, 2010] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.

[Endriss *et al.*, 2006] Ulle Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25:315–348, 2006.

[Gardenfors, 1973] Peter Gardenfors. Assignment problem based on ordinal preferences. *Management Science*, 20(3):331–340, 1973.

[Hoefer, 2013] Martin Hoefer. Local matching dynamics in social networks. *Information and Computation*, 222:20–35, 2013.

[Jackson, 2008] Matthew O Jackson. *Social and Economic Networks*. Princeton University Press, 2008.

[Klaus *et al.*, 2016] Bettina Klaus, David F Manlove, and Francesca Rossi. Matching under preferences. In *Handbook of Computational Social Choice*, pages 333–355. Cambridge University Press, 2016.

[Manlove, 2013] David F Manlove. *Algorithmics of Matching under Preferences*, volume 2. World Scientific, 2013.

[Sandholm, 1998] Tuomas W Sandholm. Contract types for satisficing task allocation. In *Proceedings of the AAAI Spring Symposium*, pages 23–25, 1998.

[Shapley and Scarf, 1974] Lloyd Shapley and Herbert Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37, 1974.

[Yoshinaka, 2005] Ryo Yoshinaka. Higher-order matching in the linear lambda calculus in the absence of constants is NP-complete. In *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA)*, pages 235–249, 2005.