Constrained Swap Dynamics over a Social Network in Distributed Resource Reallocation

Abdallah Saffidine* and Anaëlle Wilczynski†

* Australian National University, College of Engineering & Computer Science, Canberra, Australia, abdallah.saffidine@gmail.com
† Université Paris-Dauphine, PSL, CNRS, LAMSADE, Paris, France, anaelle.wilczynski@dauphine.fr

Abstract. We examine a resource allocation problem where each agent is to be assigned exactly one object. Agents are initially endowed with a resource that they can swap with one another. However, not all exchanges are plausible: we represent required connections between agents with a social network. Agents may only perform pairwise exchanges with their neighbors and only if it brings them preferred objects. We analyze this distributed process through two dual questions. *Could an agent obtain a certain object if the swaps occurred favourably? Can an agent be guaranteed a certain level of satisfaction regardless of the actual exchanges?* These questions are investigated through parameterized complexity, focusing on budget constraints such as the number of exchanges an agent may be involved in or the total duration of the process.

Keywords: Resource allocation; Distributed process; Social network; Parameterized complexity

1 Introduction

Reallocating resources among agents is a central question widely studied both in computer science and economics [14, 10, 3, 1]. This problem refers to a particular setting of resource allocation, where the agents are initially endowed with items [6]. Resource reallocation models many real-life situations, like reallocating tasks between employees or reassigning time slots in schedules. In such examples, agents are often assigned a single task. With exactly one item per agent, the problem is known as *housing-market* [1, 19]. In this context, a central authority may decide how to redistribute the objects [19, 4]. Alternatively, the agents may direct the reallocation by trading and negotiating among them in a distributed process [8, 9, 7]. Although largely studied in general resource reallocation [18, 12, 14, 11], this approach has only recently been introduced in housing market [10].

In a distributed process, natural obstacles may inhibit the agents in the trades. Lack of trust may lead agents to adopt a greedy behavior so as to be immediately better off in their new acquisition. Logistics difficulties, e.g., communication and geographical distance, may also prevent some trades to occur.

This can be modeled by restricting trades to the links of a social network [17, 13], with exchanges limited to the cliques [9] or edges [16] of the graph.

We consider this latter format: a housing market with exchanges between neighbors in a social network [16]. One of the main questions is the Reachable Object (RO) problem: Given a target agent A and a target object x, is there a sequence of exchanges ensuring A is eventually allocated x? This problem is very appealing but is NP-complete even when the network is a tree [16]. We attempt to mitigate this negative result by looking at more realistic constrained settings.

We draw inspiration from the fact that an agent may not be willing to perform a large number of swaps or to wait a long time before getting the desired object. We introduce natural budget constraints, the number of exchanges agents may make and the total duration of the process, and we perform a refined complexity analysis. Moreover, we introduce Guaranteed Level of Satisfaction (GLS), a problem related to RO but more realistic. GLS asks whether an agent can be guaranteed to be eventually allocated an item at least as good as the input target item, regardless of the exchanges other agents perform, provided they are rational swaps. While RO takes an optimistic perspective, GLS adopts a more pessimistic point of view beyond "lucky" exchange sequences. These problems naturally arise when we analyze the distributed process of exchanges in reallocation but they can also model concrete issues. As an example, consider an online exchange platform where users input in the system which item they hold as well as their preference. A user may request a target object to the centralized system which would then suggest a series of intermediate swaps to bring it to her. Even in such a context, restricted rational exchanges are relevant: geographical constraints can still prevent two agents to trade and the guarantee of getting a better object is essential as otherwise an agent could be left worse off than she started, should an intermediate agent exit the system during the process.

When parameterizing the problems by the maximal number of swaps per agent, we show intractability even for highly structured graphs. However, when constraining the duration of the process, we obtain more promising results: RO and GLS are tractable in a very relevant class of network, namely bounded degree graphs, and in general the problems are tractable when the duration does not depend on the input size. These results contrast strongly with both previous work [16] and our first parameterization, and they focus on realistic scenarios: actual social networks have indeed bounded degrees and the time that an agent is willing to wait for a target object is independent from the input size.

We start with the swap dynamics model, RO and GLS, and some complexity background. Section 3 relates RO and GLS. Our results for the max-swaps and for the total-duration parameters are in Section 4 and 5 respectively.

2 Formal Framework

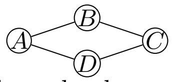
2.1 Swap Dynamics Model

Let N be a set of n agents, and M a set of n resources (or objects). Each agent $i \in N$ has ordinal strict preferences $\succ_i$ over the objects. An allocation σ is a bijection $\sigma : N \to M$, assigning to each agent exactly one object. The object assigned to agent i in σ is denoted by σ_i. Each agent is initially endowed with an object, and we denote by σ^0 the initial allocation. The agents are embedded in a social network, represented by an undirected graph $G = (N, E)$. An instance of swap dynamics model is a tuple $\mathcal{I} = (N, M, \succ, \sigma^0, G)$.

Agents can exchange their objects so as to obtain better objects, but not all exchanges are plausible. The possibilities depend on the social network and on the preferences of the agents. We only admit *swaps*, rational trades between neighbors. Formally, a swap in an allocation σ is a trade between two adjacent agents $(i, j) \in E$ such that the exchange is rational, i.e., $\sigma_i \succ_j \sigma_j$ and $\sigma_j \succ_i \sigma_i$.

A sequence of swaps is a sequence of allocations $(\sigma^0, \ldots, \sigma^t)$ such that a swap is performed between two consecutive allocations σ^i and σ^{i+1}. An allocation σ is *reachable* if there is a sequence of swaps leading to it, i.e., there exists a sequence $(\sigma^0, \ldots, \sigma^t)$ such that $\sigma^t = \sigma$. An allocation σ is *stable* if no swap is possible from σ. An object x is reachable for agent i if there is a sequence of swaps $(\sigma^0, \ldots, \sigma^t)$ where $\sigma_i^t = x$. *Swap dynamics* refers to a distributed process where agents may rationally exchange their objects when they are neighbors in the network, until a stable allocation is reached.

Example 1. Consider an instance where $n = 4$ with the following social network and preferences. The framed objects represent the initial object of each agent.



$A : b \succ c \succ \boxed{a} \succ d$ $\qquad$ $C : d \succ a \succ b \succ \boxed{c}$

$B : c \succ a \succ \boxed{b} \succ d$ $\qquad$ $D : a \succ b \succ \boxed{d} \succ c$

Initially, only the swaps between agents A and B, and B and C are possible. The rational swap (A, C) is not possible because the agents are not adjacent. The swap (A, D) is not possible because it is not rational for A. The sequence of exchanges (A, B), (B, C), and (C, D) gives rise to a reachable allocation where every agent gets her best object. This is stable: no further swap can be performed.

2.2 Questions

We investigate swap dynamics by analyzing two natural decision problems.

Reachable Object (RO):
Instance: $\mathcal{I} = (N, M, \succ, \sigma^0, G)$, $A \in N$, $x \in M$.
Question: Is there a sequence of swaps $(\sigma^0, \ldots, \sigma^t)$ such that $\sigma_A^t = x$?

Guaranteed Level of Satisfaction (GLS):
Instance: $\mathcal{I} = (N, M, \succ, \sigma^0, G)$, $A \in N$, $y \in M$.
Question: For all sequence of swaps $(\sigma^0, \ldots, \sigma^t)$ where σ^t is stable, does it hold that either $\sigma_A^t = y$ or $\sigma_A^t \succ_A y$?

When asking whether an agent can obtain some object, swap dynamics with a large number of swaps may not be realistic. We thus study three variants of RO and GLS where the quantity of swaps in a solution sequence is limited. In each variant, this quantity is measured differently, leading to different complexity-theoretic characterizations of the problem.

– *max*: Every agent is involved in no more than k swaps.
– *sum*: The total length of the sequence is no more than k.
– *makespan*: The makespan of the sequence is no more than k.

The *makespan* of a sequence of swaps is the minimum time that elapses from the beginning to the end, when we allow parallel swaps. This notion can be formalized as follows. Let $s = (\sigma^0, \dots, \sigma^t)$ be a sequence of swaps. A parallel decomposition of s is a tuple of integers $\ell = (\ell_0, \ell_1, \dots, \ell_m)$ of length $|\ell| = m$, such that $0 = \ell_0 < \ell_1 < \cdots < \ell_m = t$, and for all $0 \leq i < m$ the swaps between allocation σ^{ℓ_i} and allocation $\sigma^{\ell_{i+1}}$ do not involve the same agents. In other words, the swaps between σ^{ℓ_i} and $\sigma^{\ell_{i+1}}$ can be performed simultaneously. The makespan is the length m of the shortest parallel decomposition. Observe that the makespan of a sequence can be computed in linear time, and that the *sum* parameter is a worst case bound in case no parallel swaps take place.

2.3 Parameterized complexity

Parameterized complexity aims at solving hard problems in time $f(k)n^{O(1)}$, called FPT time (fixed-parameter tractable), where n is the size of the instance, f is a computable function, and k is a *parameter* of the problem. Assuming the problem we are trying to solve is NP-hard, function f has to be superpolynomial, unless P = NP. However, if our parameter k is *small* compared to the size of the instance n, we achieve that the blow-up is limited to the small value k.

Some problems are highly suspected not to admit any algorithm in time $f(k)n^{\mathcal{O}(1)}$ for any computable function f, and thus to be FPT. There are hierarchies of complexity classes beyond FPT: W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ W[SAT] $\subseteq \dots$, and A[1] $\subseteq$ A[2] $\subseteq \cdots \subseteq$ AW[SAT] $\subseteq \dots \subseteq$ XP, where W[1] = A[1], W[t] $\subseteq$ A[t] for any t >1, and W[SAT] $\subseteq$ AW[SAT]. For instance, W[1] is the class of parameterized decision problems that can be solved by a nondeterministic single-tape Turing machine within k steps, and XP is the class of decision problems solvable in time $\mathcal{O}(n^{f(k)})$ for some computable function f.

As a rudimentary informal intuition, FPT and W[1] can be thought of as corresponding to P and NP in the parameterized world. For instance, CLIQUE, the problem of finding a clique of size k in a graph, is W[1]-complete for FPT reductions, where an FPT reduction may blow-up the instance size n only polynomially but the new parameter can be any computable function of the old parameter.

2.4 First-order logic

A vocabulary τ is a finite set of relation symbols. A finite structure $\mathcal{A}$ over τ consists of a finite set A, called the universe, and for each R in τ a relation over A. We assume a countably infinite set of variables. Atoms over vocabulary τ are of the form $x_1 = x_2$ or $R(x_1, \dots, x_k)$ where $R \in \tau$ and $x_1, \dots, x_k$ are variables. First-order (FO) formulas over τ are built from atoms over τ using standard Boolean connectives $\neg, \wedge, \vee$ and from quantifiers $\exists, \forall$ followed by a variable. Let φ be an FO formula. The variables of φ that are not in scope of a quantifier are its *free variables.* Let $\varphi(\mathcal{A})$ be the set of all assignments of elements of A to φ's free variables, such that φ is satisfied. $\mathcal{A}$ is a *model* of φ if $\varphi(\mathcal{A})$ is not empty. The class Σ_1 (resp. Σ_2) contains all FO formulas of the form $\exists x_1, \dots, \exists x_k \varphi$ (resp. $\exists x_1, \dots, \exists x_k \forall y_1, \dots, \forall y_k \varphi$) where φ is a quantifier free FO formula.

Let Φ be a class of formulas. The model checking problem inputs a finite structure $\mathcal{A}$ and a formula $\varphi \in \Phi$ and asks whether the formula satisfies the model, $\varphi(\mathcal{A}) \neq \emptyset$. A natural parameter is the size of (a reasonable encoding of) φ. We will use one result bridging model checking and parameterized complexity.

Theorem 1. [15] *Model checking the existential fragment of first-order logic,* $\mathrm{MC}(\Sigma_1)$*, is* W[1]*-complete. Model checking the second level of the hierarchy,* $\mathrm{MC}(\Sigma_2)$*, is* A[2]*-complete.*

3 Relation between RO and GLS

Reachable Object (RO) asks whether an agent A can obtain an object x by a sequence of swaps. RO is known to be NP-complete even for trees [16]. Guaranteed Level of Satisfaction (GLS) asks whether agent A is guaranteed to obtain object y or an object preferred to y in any stable reachable allocation. GLS is even more natural than RO since it offers guarantees for the agent and does not only focus on lucky configurations. It is close to the complementary of RO, and thus the study of RO also contributes to the understanding of GLS.

Proposition 2. *co-*RO *is linearly reducible to* GLS*.*

Proof sketch. We perform a reduction from the co-RO problem asking whether object x is unreachable for agent A. Let $\mathcal{I} = \{(N, M, \succ, \sigma^0, G), A, x\}$ be an instance of co-RO. An instance $\mathcal{I}' = \{(N', M', G', \succ', \sigma^{0\prime}), A, y\}$ of GLS is constructed by adding an agent Y and an object y. Initial allocation $\sigma^{0\prime}$ is the same as σ^0 for all agents in N and assigns y to Y. The social network G' has the same structure as G, with one more edge (Y, A). Denote by a the object of agent A in σ^0, and by P_a the set of objects ranked in $\succ_A$ between x and a (a included, x excluded), if x is preferred to a. If A does not prefer x to a, then P_a contains a and all the objects preferred to a. Denote by P_x the set of objects ranked before x in the preferences of A. The preferences $\succ'$ are constructed from $\succ$, by moving the objects in P_x to the end of $\succ'_A$, and by putting y at the top of $\succ'_A$. The agents in $N \setminus \{A\}$ rank y last, and Y only prefers the objects of P_a to object y.

We claim that x is not reachable for A in $\mathcal{I}$ iff A obtains y or an object preferred to y in any reachable stable allocation in $\mathcal{I}'$. This part is omitted. $\square$

This result establishes the computational difficulty of GLS, since RO is NP-complete even on trees [16] and the previous reduction adds only one agent and possibly one swap to an instance of RO. Moreover, observe that GLS belongs to co-NP: after guessing a reachable stable allocation, one can directly check whether agent A owns y or an object that she prefers to y.

Corollary 3. GLS *is* co-NP*-complete even for trees.*

We refine the complexity of the problems using natural parameters: the number of swaps per agent and the length of the sequence. Although RO and GLS are close to be dual problems, that they are indeed not complementary. GLS focuses on stable allocations. A k-bound on the sequence of swaps introduces a dependency on k on the notion of stability: a stable allocation is either stable in the standard meaning, or is reached after k swaps. Stability is not necessary in RO because for an assignment solution σ where agent A gets object x, all the stable allocations reachable from σ assign to A an object preferred to x or x.

4 Maximum Number of Swaps per Agent

Consider that the agents are not willing to perform an important number of swaps in the whole swap process. Surprisingly in this context, our two problems, RO and GLS, remain difficult even for a very small maximum number of swaps.

Theorem 4. *For fixed $k \geq 2$, RO-max is NP-complete, even on degree 4 graphs.*

Proof. Membership in NP is straightforward, as it is a special case of the unconstrained RO problem, known to be in NP.

For hardness, we start with $k = 2$ and reduce from (3, B2)-SAT—the restriction of SAT to instances where each clause contains three literals and each variable occurs exactly twice as a positive literal and twice as a negative literal. This variant of the propositional satisfiability problem is NP-complete [5].

We are given an instance of (3, B2)-SAT with n variables $\{x_1, \ldots, x_n\}$ and m clauses $\{C_1, \ldots, C_m\}$. We create a literal-agent Y_j^ℓ (resp., $\overline{Y_j^\ell}$) for each ℓ^{th} ($\ell \in \{1, 2\}$) occurrence of literal x_j (resp., $\overline{x_j}$), and a variable-agent Y_j for each variable x_j. Two clause-agents K_i and K'_i are created for each $0 < i < m$. Three other agents Y_0, K'_0 and K_m are added. Each agent initially owns an object denoted by the lower-case version of her name, e.g., agent K_i gets object k_i.

In the network, we have the paths $[Y_{j-1}, Y_1^1, Y_1^2, Y_j]$ and $[Y_{j-1}, \overline{Y_1^1}, \overline{Y_1^2}, Y_j]$ for each $1 \leq j \leq n$, and the edge (K_i, K'_i) for each $1 \leq i < m$. If the ℓ^{th} literal x_j (resp., $\overline{x_j}$) belongs to clause C_i, then we have the path $[K'_{i-1}, Y_j^\ell, K_i]$ (resp., $[K'_{i-1}, \overline{Y_j^\ell}, K_i]$). We connect K_m and Y_n. See for an example Figure 1.

The preferences of the agents are given below. Notation $\{\ell_i\}$ stands for the literal-objects of clause C_i ranked in arbitrary order and $k(x_i^j)$ (resp. $k(\overline{x}_i^j)$) for the object related to the clause in which the j^{th} occurrence of x_i (resp. $\overline{x_i}$) appears. The objects that are not mentioned in the preferences are ranked in arbitrary order after the initial endowment.

$K'_0 : \ \{\ell_1\} \succ \boxed{b} \succ [\ldots]$

$K_i : \ k'_i \succ b \succ \boxed{k_i} \succ [\ldots]$ $\qquad Y_j^1 : k(x_j^1) \succ b \succ y_j^2 \succ a \succ \boxed{y_j^1} \succ [\ldots]$

$K'_i : \ \{\ell_{i+1}\} \succ b \succ \boxed{k'_i} \succ [\ldots]$ $\qquad Y_j^2 : k(x_j^2) \succ b \succ y_j \succ a \succ \boxed{y_j^2} \succ [\ldots]$

$K_m : a \succ b \succ \boxed{k_m} \succ [\ldots]$ $\qquad \overline{Y}_j^1 : k(\overline{x}_j^1) \succ b \succ \overline{y}_j^2 \succ a \succ \boxed{\overline{y}_j^1} \succ [\ldots]$

$Y_0 : \ \overline{y}_1^1 \succ y_1^1 \succ \boxed{a} \succ [\ldots]$ $\qquad \overline{Y}_j^2 : k(\overline{x}_j^2) \succ b \succ y_j \succ a \succ \boxed{\overline{y}_j^2} \succ [\ldots]$

$Y_n : \ b \succ a \succ \boxed{y_n} \succ [\ldots]$ $\qquad Y_j : \ \overline{y}_{j+1}^1 \succ y_{j+1}^1 \succ a \succ \boxed{y_j} \succ [\ldots]$

We claim that all clauses are satisfiable iff object b reaches agent Y_n. The only way for Y_n to get hold of b is by swapping a with K_m. Object b can only reach K_m via clause-agents and literal-agents, while a can only reach Y_n via variable-agents and literal-agents. Agents perform at most two swaps, so no literal-agents can be involved in the move of both a and b.

Suppose that truth assignment ϕ satisfies all clauses. Let T_i be a literal-agent of clause C_i related to a true literal in ϕ. Since all clauses are satisfiable, object b can reach K_m via the path $[K'_0, T_1, K_1, K'_1, T_2, \ldots, T_{m-1}, K_{m-1}, K'_{m-1}, T_m, K_m]$. For variable x_j, let Z^1_j and Z^2_j be the literal-agents related to the literal of x_j that is false in ϕ. Clearly, these agents are not an agent T_i. It suffices for a to reach Y_n via the path $[Y_0, Z^1_1, Z^2_1, Y_1, \ldots, Y_{n-1}, Z^1_n, Z^2_n, Y_n]$.

Suppose now that object b is reachable for agent Y_n. By construction, the path of b to K_m goes through exactly one literal-agent per clause, while the path of a to Y_n goes through exactly two literal-agents associated with the same literal for each variable. Thus, the truth assignment of variables that sets to true the literals related to literal-agents in the path of object b, satisfies all the clauses.

If $k > 2$, we adapt the reduction via a delay gadget added to each agent. $\square$

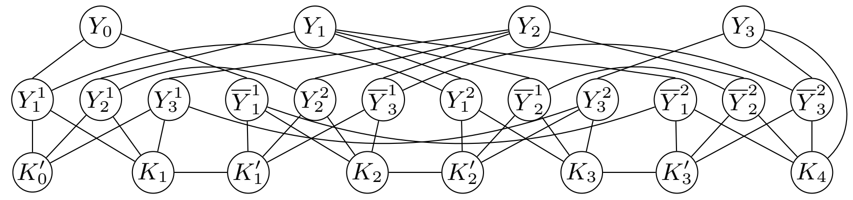


Fig. 1. Graph construction for an instance of (3, B2)-SAT with four clauses where $C_1 = (x_1 \vee x_2 \vee x_3)$, $C_2 = (\overline{x_1} \vee x_2 \vee \overline{x_3})$, $C_3 = (x_1 \vee \overline{x_2} \vee x_3)$, and $C_4 = (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$.

From Prop 2 and its proof, the same hardness exists for GLS, with an additional swap and an additional neighbor for agent A who must obtain the object.

Corollary 5. *For $k \geq 3$, GLS-max is* co-NP*-complete, even on degree 5 graphs.*

One could think that the problem is easier when the structure of the network is restricted to trees. Yet, it is possible to prove that RO-max on trees is W[SAT]-hard. We leave out the lengthy formal proof but we state the main idea. We reduce from Monotone Weighted Satisfiability [2]. *Can an input propositional formula φ with no negations be satisfied with a truth assignment of weight k?* We build an instance of RO-max with a graph based on the syntax tree of φ where k chosen variable-objects must move to the occurrences of their corresponding variable as a prerequisite to given object x reaching given agent A. Since the variable-objects make up almost all the swaps, $O(k)$ swaps per agent suffice.

Globally, the problems remain difficult in very simple graphs even when the number of swaps per agent is limited. Fortunately, the parameters on the length of the sequence lets us circumvent the general difficulty of the two problems.

5 Length of the Sequence of Swaps

Two parameters are used to bound the length of the sequence of swaps: the total number of exchanges and the makespan. Contrary to the previous parameter, they lead to circumscribe the problems into parameterized complexity classes that are not so high in the hierarchy, allowing tractability results when the parameters are bounded by a constant. Moreover, for bounded degree graphs, relevant in the context of social network, we obtain fixed parameter tractability.

Theorem 6. RO-*sum and* RO-*makespan are* W[1]-*hard even for trees.*

Proof sketch. We perform a reduction from CLIQUE, the problem of deciding whether there exists a clique of size k in a graph $G = (V, E)$ such that $V = \{1, ..., n\}$ and $|E| = m$. Assume that each edge in E is written (v, w) such that $v < w$, and consider the lexicographical order over E. Let us denote by e_i^1 and e_i^2 the first and second vertex of the i^{th} edge. Let d_v be the degree of vertex v and $\delta_v(d)$, for $1 \leq d \leq d_v$, the d^{th} edge incident to v. We construct an instance $\mathcal{I}'$ of RO (see Fig. 2 for an example) by creating:

- two connected agents X and Y, and two vertex-agents U_v^{vw} and U_w^{vw} for each edge $(v, w) \in E$, connected via a path $[Y, U_v^{vw}, U_w^{vw}]$.
- agents T and T^ℓ, for $1 \leq \ell \leq k$, representing the k vertices of the clique that we must choose. They are connected via a path to Y: $[Y, T^1, \ldots, T^k, T]$.
- agents A_v and A_v^ℓ, for $v \in V$ and $1 \leq \ell < k$, representing the choice of the $k - 1$ edges of the clique that are incident to v if v belongs to the clique. They are connected via a path to Y for each v: $[Y, A_v^1, \ldots, A_v^{k-1}, A_v]$.
- agents $T^{\ell *}$ adjacent to T^ℓ, for $1 \leq \ell \leq k$, and agents $A_v^{\ell *}$ adjacent to A_v^ℓ, for $1 \leq \ell < k$ and $v \in V$. They are used to "validate" their associated agent by giving to her their initial object once they own an expected object.
- auxiliary agents used to facilitate the passage of some objects: if an agent B has a connected auxiliary agent $B^{[z]}$, then the swap with $B^{[z]}$ must precede a swap for getting an object associated with z. The auxiliary agents we use are agents $Y^{[vw]}$ corresponding to edge (v, w) and connected to Y, agents $Y^{[v]}$ corresponding to vertex v and connected to Y, agent $Y^{[t]}$ corresponding to object t and connected to Y, agents $A_v^{\ell[\delta_v(d)]}$ corresponding to edge $\delta_v(d)$, for $1 \leq d \leq d_i$, and connected to agent A_v^ℓ, for $v \in V$ and $1 \leq \ell < k$, and agents $T^{\ell[v]}$ corresponding to vertex v and connected to agent T^ℓ for $1 \leq \ell \leq k$.

The initial object of an agent is denoted by the lower-case version of her name, e.g., agent $Y^{[v]}$ gets object $y^{[v]}$. The preferences of the agents are as follows (objects in brackets may not exist for all indices).

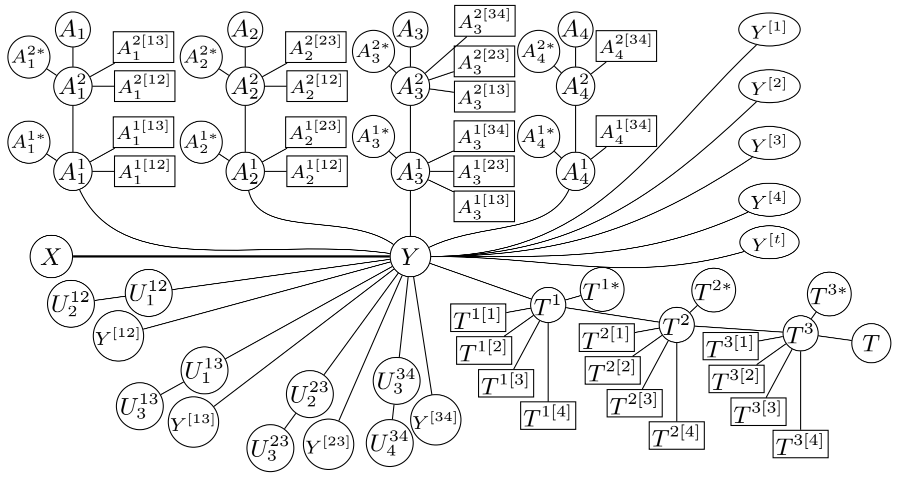


Fig. 2. Graph construction for an instance of CLIQUE with vertices $\{V_1, V_2, V_3, V_4\}$ and $k = 3$. The edges are: $\{V_1, V_2\}$, $\{V_1, V_3\}$, $\{V_2, V_3\}$ and $\{V_3, V_4\}$.

$$\begin{array}{ll|ll}
X: & t \succ \boxed{x} \succ [\dots] \qquad A_v : a_v^{k-1*} \succ \boxed{a_v} \succ [\dots] & T: & t^{k*} \succ \boxed{t} \succ [\dots] \\
U_v^{vw}: & a_v^{1[vw]} \succ u_w^{vw} \succ y^{[vw]} \succ \boxed{u_v^{vw}} \succ [\dots] & U_w^{vw}: & y^{[vw]} \succ \boxed{u_w^{vw}} \succ [\dots] \\
A_v^{\ell[\delta_v(d)]}: & (a_v^{\ell+1[\delta_v(d_v)]}) \succ \dots \succ (a_v^{\ell+1[\delta_v(1)]}) \succ a_v^\ell \succ \boxed{a_v^{\ell[\delta_v(d)]}} \succ [\dots] & A_v^{\ell*}: & u_v^{\delta_v(d_v)} \succ \dots \succ u_v^{\delta_v(1)} \succ \boxed{a_v^{\ell*}} \succ [\dots] \\
T^{\ell[v]}: & (t^{\ell+1[v-1]}) \succ \dots \succ (t^{\ell+1[1]}) \succ t^\ell \succ \boxed{t^{\ell[v]}} \succ [\dots] & T^{\ell*}: & a_1 \succ a_2 \succ \dots \succ a_n \succ \boxed{t^{\ell*}} \succ [\dots]
\end{array}$$

$$\begin{array}{ll}
Y^{[v]}: & (t^{1[v-1]}) \succ \dots \succ t^{1[1]} \succ a_{e_m^2}^{1[e_m]} \succ \dots \succ a_{e_1^2}^{1[e_1]} \succ \boxed{y^{[v]}} \succ [\dots] \\
Y^{[vw]}: & a_{e_m^2}^{1[e_m]} \succ \dots \succ a_{e_1^2}^{1[e_1]} \succ y \succ \boxed{y^{[vw]}} \succ [\dots] \\
A_v^\ell: & y^{[v]} \succ (a_v^{\ell-1*}) \succ a_v \succ a_v^{\ell*} \succ u_v^{\delta_v(d_v)} \succ a_v^{\ell[\delta_v(d_v)]} \succ \dots \succ \\
 & (a_v^{\ell+1[\delta_v(2)]}) \succ u_v^{\delta_v(2)} \succ a_v^{\ell[\delta_v(2)]} \succ (a_v^{\ell+1[\delta_v(1)]}) \succ u_v^{\delta_v(1)} \succ a_v^{\ell[\delta_v(1)]} \succ \boxed{a_v^\ell} \succ [\dots] \\
T^\ell: & y^{[t]} \succ (t^{\ell-1*}) \succ t \succ t^{\ell*} \succ a_n \succ t^{\ell[n]} \succ \dots \succ \\
 & (t^{\ell+1[2]}) \succ a_2 \succ t^{\ell[2]} \succ (t^{\ell+1[1]}) \succ a_1 \succ t^{\ell[1]} \succ \boxed{t^\ell} \succ [\dots] \\
Y: & x \succ t \succ y^{[t]} \succ t^{1[n]} \succ a_n \succ y^{[n]} \succ \dots \succ t^{1[1]} \succ a_1 \succ y^{[1]} \succ \\
 & a_{e_m^2}^{1[e_m]} \succ u_{e_m^2}^{e_m} \succ a_{e_m^1}^{1[e_m]} \succ u_{e_m^1}^{e_m} \succ y^{[e_m]} \succ \dots \succ a_{e_1^2}^{1[e_1]} \succ u_{e_1^2}^{e_1} \succ a_{e_1^1}^{1[e_1]} \succ u_{e_1^1}^{e_1} \succ y^{[e_1]} \succ \boxed{y} \succ [\dots]
\end{array}$$

We claim that there exists a clique of size k in graph G iff object x can reach agent Y within a total of $k^3 + 4k^2 + k + 2$ swaps or a makespan of $5k(k-1)/2 + 3k + 4$. An agent A_v^ℓ (or T^ℓ) is said to be "validated" if she obtains at a moment object $a_v^{\ell*}$ (or $t^{\ell*}$). We omit the details of the proof but the idea is that, let object x reach agent Y, all the k agents T^ℓ and all the $k-1$ agents A_v^ℓ of k branches A_v need to be validated. The associated clique in graph G is given by the vertices v for which all the $k-1$ agents A_v^ℓ have been validated. All the agents A_v^ℓ, for $1 \leq \ell < k$, are validated if we can bring in the branch $k-1$ objects u_v^{vw} (or u_v^{wv}, following the order) representing an edge incident to v. Observe that the given budget allows bringing in the branches only $k(k-1)$ objects u_v^{vw} and the construction forces to choose u_w^{vw} if u_v^{vw} has been chosen. □

Combining the proofs of Thm. 6 and Prop. 2 leads to hardness for GLS.

Corollary 7. GLS-*sum and* GLS-*makespan are* co-W[1] *hard even for trees.*

This W[1]-hardness for RO and GLS parameterized by the length of the sequence rules out the existence of FPT algorithms even in trees under standard complexity assumptions. However, the following results on the membership to respectively W[1] and co-A[2] show that the problems are not so hard. They are notably in XP for any graph, thus tractable when the parameter is a constant.

Theorem 8. *RO-sum is in* W[1].

Proof. An instance $\mathcal{I}$ of RO with a swap dynamics model $(N, M, \succ, \sigma^0, G)$, agent A and object x, and k as a total number of swaps, is transformed into an instance $\mathcal{I}' = (\mathcal{A}, \varphi)$ of $\mathrm{MC}(\Sigma_1)$, known to be W[1]-complete (Thm. 1). Structure $\mathcal{A}$ is an $(E, \succ, \sigma^0, A, X)$-structure with variables in $N \cup M$, where relations E, $\succ$, σ^0, A and X are defined as follows. The binary relation E over N^2 represents the edge set E. The ternary relation $\succ$ over $N \times M^2$ represents the preferences of the agents, i.e. $\succ(i, a, b)$ means that agent i prefers object a to b. For the sake of clarity, we write $a \succ_i b$ instead of $\succ(i, a, b)$. The binary relation σ^0 over $N \times M$ represents the initial allocation σ^0, i.e. $\sigma^0(i, z)$ means that i is initially endowed with z. Finally, the unary relations A and X respectively represent agent A and object x, i.e. $A(y)$ means that y is agent A and $X(y)$ means that y is object x.

The Σ_1-formula φ is defined as $\varphi = \exists x_0 \exists b_0 \exists x_1 \exists y_1 \exists a_1 \exists b_1 \ldots \exists x_k \exists y_k \exists a_k \exists b_k \Big(\sigma^0(x_0, b_0) \wedge \bigvee_{0 \le k' \le k} \psi^{k'}\Big)$ with

$$\psi^{k'} \equiv A(x_{k'}) \wedge X(b_{k'}) \wedge \bigwedge_{i=1}^{k'} \Big(E(x_i, y_i) \wedge b_i \underset{x_i}{\succ} a_i \wedge\ a_i \underset{y_i}{\succ} b_i \wedge\ \mathrm{o}_i(x_i, a_i) \wedge \mathrm{o}_i(y_i, b_i)\Big)$$

where for all i, $\mathrm{o}_i(q, r)$ stands for $\Big(\sigma^0(q, r) \wedge \bigwedge_{j=1}^{i-1} x_j \neq q \wedge y_j \neq q\Big) \vee \bigvee_{j=1}^{i-1} \Big(\bigwedge_{p=j+1}^{i-1} x_p \neq q \wedge q_p \neq q\Big) \wedge \Big((x_j = q \wedge \mathrm{o}_j(y_j, r)) \vee \Big(y_j = q \wedge \mathrm{o}_j(x_j, r)\Big)\Big)$.

One can prove by induction over i that formula $\mathrm{o}_i(q, r)$ is true iff object r is owned by agent q before i^{th} swap. Globally, formula $\psi^{k'}$ is true iff the sequence of exchanges between the agents (x_i, y_i) exchanging the objects (a_i, b_i), for $i \in \{1, \ldots, k'\}$, is a sequence of swaps leading to give object x to agent A. □

The same idea and a slightly different FO formula work for RO-makespan.

Proposition 9. *RO-makespan is in* W[1].

Proof sketch. We reduce to $\mathrm{MC}(\Sigma_1)$ but face a new difficulty. We cannot quantify over all potential exchanges within makespan k: it would lead to a formula of size $\Omega(n)$. The crux of this proof is to observe that not all exchanges are relevant to decide the problem. Assume we process independent swaps in parallel for up to k time steps. Looking at it from the end, the only relevant swap in the last step k involves agent A, so we quantify over a single swap and ignore all concurrent ones. In the one-before-last, only swaps involving A or A's partner at step k may be relevant. So considering two swaps happening at step $k - 1$ and ignoring all other concurrent ones suffices. All in all, we need to quantify over no more than 2^{k+1} exchanges. The rest is similar to that of Thm. 8. □

A similar reasoning is applied to GLS. We reduce GLS to model-checking FO formula using more sophisticated Σ_2 formulas.

Proposition 10. GLS-*sum/-makespan is in* co-A[2].

Proof sketch. We reduce co-GLS to MC(Σ_2), known to be A[2]-complete (Thm 1), following an approach similar to that of Thm 8 and Prop 9. $\square$

The previous results show that RO and GLS are not "so hard" considering the length of the sequence as a parameter. Furthermore, for some natural classes of graphs, the problems are even tractable with respect to these parameters.

Proposition 11. RO/GLS-*sum/makespan are* FPT *on bounded degree graphs.*

Proof. The proof follows the idea developed for Proposition 9. Let Δ be the degree of G and consider the RO problem. At the k^{th} step, the only relevant exchange involves agent A and a neighbor, so there are $\mathcal{O}(\Delta)$ possible swaps. The one-before-last step can only involve A or one her neighbor, therefore there are at most 2Δ possible swaps for RO-sum and at most $\Delta + \Delta^2$ for RO-makespan. This argument applies at any of the k steps, hence there are $\mathcal{O}(\Delta^k.k!)$ sequences of swaps for RO-sum and $\mathcal{O}(\Delta^{k^2})$ for RO-makespan, and it suffices to verify if one sequence assigns x to A. Concerning GLS, it suffices to test the reachability to A of any object x such that $y \succ_A x$, and so it just adds a factor of n. $\square$

6 Conclusion and Perspectives

This article studies the distributed process of swap dynamics along a network for reallocating objects among agents. Two related problems are investigated: Reachable Object (RO), "can a given agent obtain a given object?", and Guaranteed Level of Satisfaction (GLS), "is a given agent guaranteed to get a given object or better?". Both problems are hard but the parameterized approach allows us to escape this difficulty for a relevant class of graphs.

We consider natural parameters constraining the number of swaps per agent or the duration of the sequence. Assuming that they remain small is reasonable in practice as the patience of the agents typically does not increase with the instance size. In the case of few swaps per agent, RO and GLS remain hard even on bounded degree graphs. So, this parameterization, although natural, does not help us to grasp the problems. However, considering the length of the sequence, although both problems are intractable even for trees, this hardness is circumscribed to not "so hard" parameterized complexity classes, leading to the possibility of handling the problems when the parameters do not depend on the instance size, very natural assumption. Furthermore, unlike the first parameter, the length of the sequence permits to obtain fixed parameter tractability on bounded degree graphs, which typically model real social networks.

The parameterized approach allows progress in the understanding of the problems and leads to significant and realistic positive results. So far, we have considered restrictions on the network as well as on the solution size. A natural extension is to investigate the influence of a third dimension: constraints on the preference profile, e.g., single-peaked domains. Furthermore, assuming the full knowledge of the preferences and the network is not relevant in all the contexts. Relaxing this assumption could be a challenging future work.

References

1. Abdulkadiroğlu, A., Sönmez, T.: House allocation with existing tenants. Journal of Economic Theory **88**(2), 233–260 (1999)
2. Abrahamson, K.A., Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogues. Annals of Pure and Applied Logic **73**(3), 235–276 (1995)
3. Aziz, H., Biró, P., Lang, J., Lesca, J., Monnot, J.: Optimal reallocation under additive and ordinal preferences. In: AAMAS. pp. 402–410 (2016)
4. Aziz, H., De Keijzer, B.: Housing markets with indifferences: A tale of two mechanisms. In: AAAI. pp. 1249–1255 (2012)
5. Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness of short symmetric instances of max-3sat. Tech. rep. (2004)
6. Chevaleyre, Y., Dunne, P., Endriss, U., Lang, J., Lemaître, M., Maudet, N., Padget, J., Phelps, S., Rodrıguez-Aguilar, J.A., Sousa, P.: Issues in multiagent resource allocation. Informatica **30**(1), 3–31 (2006)
7. Chevaleyre, Y., Endriss, U., Estivie, S., Maudet, N.: Multiagent resource allocation in k-additive domains: Preference representation and complexity. Annals of Operations Research **163**(1), 49–62 (2008)
8. Chevaleyre, Y., Endriss, U., Lang, J., Maudet, N.: Negotiating over small bundles of resources. In: AAMAS. pp. 296–302 (2005)
9. Chevaleyre, Y., Endriss, U., Maudet, N.: Allocating goods on a graph to eliminate envy. In: AAAI. pp. 700–705 (2007)
10. Damamme, A., Beynier, A., Chevaleyre, Y., Maudet, N.: The power of swap deals in distributed resource allocation. In: AAMAS. pp. 625–633 (2015)
11. Dunne, P.E., Chevaleyre, Y.: The complexity of deciding reachability properties of distributed negotiation schemes. Theoretical Computer Science **396**(1-3), 113–144 (2008)
12. Dunne, P.E., Wooldridge, M., Laurence, M.: The complexity of contract negotiation. Artificial Intelligence **164**(1-2), 23–46 (2005)
13. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning about a Highly Connected World. Cambridge University Press (2010)
14. Endriss, U., Maudet, N., Sadri, F., Toni, F.: Negotiating socially optimal allocations of resources. Journal of Artificial Intelligence Research **25**, 315–348 (2006)
15. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer (2006)
16. Gourvès, L., Lesca, J., Wilczynski, A.: Object allocation via swaps along a social network. In: IJCAI. pp. 213–219 (2017)
17. Jackson, M.O.: Social and Economic Networks. Princeton University Press (2008)
18. Sandholm, T.W.: Contract types for satisficing task allocation. In: AAAI Spring Symposium. pp. 23–25 (1998)
19. Shapley, L., Scarf, H.: On cores and indivisibility. Journal of Mathematical Economics **1**(1), 23–37 (1974)

A Proofs for Section 3 (Relation between RO and GLS)

Proposition 2. *co*-RO *is linearly reducible to* GLS.

Proof. Let us show that x is not reachable for A in $\mathcal{I}$ iff A obtains y or an object preferred to y in any reachable stable allocation in $\mathcal{I}'$.

Suppose that x is not reachable for A in $\mathcal{I}$. At a stable allocation in $\mathcal{I}$, either (i) agent A gets an object from P_x without having owned x during the sequence, but it is not possible in $\mathcal{I}'$ because $a \succ'_A P_x$, and then agent A stays in $\mathcal{I}'$ with object a, or (ii) agent A gets an object from P_a, and it will be also the case in $\mathcal{I}'$. In both cases, agent A can exchange with Y in order to obtain y, and thus gets an object preferred or equal to y.

Suppose now that x is reachable for A in $\mathcal{I}$. It follows that x is also reachable for A in $\mathcal{I}'$ since the social network G' keeps the same structure as G concerning the agents of N, and A does not use the objects of P_x to obtain x in $\mathcal{I}$. However, by obtaining x, agent A cannot exchange any more with Y because of the construction of the preferences $\succ'$ and the social network G'. Thus, agent A cannot obtain an object preferred or equal to y in this sequence of swaps.

Observe that in this reduction, GLS asks whether agent A is guaranteed to obtain her best object. The generalization to an object ranked at k^{th} position is straightforward by inserting $k+1$ agents and objects, not accessible for A. □

B Proofs for Section 5 (Length of the Sequence of Swaps)

Theorem 6. RO-*sum and* RO-*makespan are* W[1]-*hard even for trees.*

Proof. Let us prove that there exists a clique of size k in graph G iff object x is reachable for agent Y within at most $k^3 + 4k^2 + k + 2$ swaps in total.

Suppose there exists a clique of size k in graph G. Let V_C, with $|V_C| = k$, be the set of all vertices belonging to the clique, and C be the set of all edges of the clique. We consider the edges (v, w) in C with respect to the order over the edges previously assumed in the construction of the instance. Let us perform the rational swaps between the following couples of agents for $(v, w) \in C$: $\{Y, Y^{[vw]}\}$, $\{Y, U_v^{vw}\}$, and $\{U_w^{vw}, U_v^{vw}\}$, that lead to give object u_v^{vw} to agent Y. Then, we decide to let object u_v^{vw} pass to branch A_v. At this moment, a further swap can be performed in the branch U^{vw} between Y and U_v^{vw}. Within the U^{vw}'s branches, we perform in total $\mathbf{2k(k-1)}$ **swaps**, considering all the $k(k-1)/2$ edges of the clique in C. Now let us focus on the passage of an object u_v^{vw} (resp. u_w^{vw}) to branch A_v (resp. A_w). To make the swaps rational within branch A_v, we need some auxiliary agents. We previously perform the swaps between agent A_v^ℓ and agent $A_v^{\ell[vw]}$, for each $1 \leq \ell \leq k - j$, if object u_v^{vw} in question is the j^{th} object to come into this branch. Then we perform the swaps along the path $[Y, A_v^1, \ldots, A_v^{k-j}]$ ($\mathbf{k^2(k-1)}$ **swaps** in total by considering all the objects u_v^{vw} and u_w^{vw} associated with an edge of the clique. After having performed all these swaps, all the agents A_v^ℓ, for $1 \leq \ell < k$ and $v \in V_C$, possess an object u_v^{vw} (or u_v^{wv}) associated with an edge (v, w) of the clique, and then can exchange with agent $A_v^{\ell *}$ in order to obtain object $a_v^{\ell *}$ and be "validated" (in total $\mathbf{k(k-1)}$ **swaps**). By rationality of the swaps, object a_v can now go to agent A_v^1 via path $[A_v, A_v^{k-1}, \ldots, A_v^1]$, for each $v \in V_C$ ($\mathbf{k(k-1)}$ swaps in total). Then, by increasing order of vertices v over V_C, let Y swap with $Y^{[v]}$, and then with A_v^1, in order to obtain object a_v. (**2k swaps** in total by considering each $v \in V_C$). Let us focus now on the passage of object a_v to the branch of T^ℓ. Like in the A_v's branch, if a_v is the j^{th} object to come into this branch, then all the agents T^ℓ for $1 \leq \ell \leq k+1-j$ previously perform a swap with agent $T^{\ell[v]}$ in order to let object a_v pass to reach agent T^{k+1-j} ($\mathbf{k(k+1)}$ **swaps** in total). Then, agent T^{k+1-j} can exchange with agent T^{k+1-j*} to obtain object t^{k+1-j*}, and thus T^{k+1-j} is "validated" (**k swaps**). Since there are k validated A_v's branches, all the agents T^ℓ can be validated and thus, object t can go to agent Y via path $[T, T^k, \ldots, T^1, Y]$, for which the swaps are now rational ($\mathbf{k+1}$ **swaps**). Finally, agent Y can swap with agent X since object t is the only object that X prefers to object x (**one swap**), leading to the reachability of x by Y. Observe that we have exactly performed $k^3 + 4k^2 + k + 2$ swaps.

Suppose now that object x is reachable for agent Y within at most $k^3 + 4k^2 + k + 2$ swaps. The only way for agent X to give x in a rational swap is to obtain object t in return. Therefore, agent Y must previously get object t, initially owned by agent T, who only accept to give t against t^{k*}. Moreover, t must pass by all the agents T^ℓ and each of them accepts to give t to their neighbor only against object $t^{\ell-1*}$ or $y^{[t]}$. Since for agent T_1, this object is necessarily $y^{[t]}$, it must be $t^{\ell-1*}$ for all the others. Therefore, all the agents T^ℓ for $1 \leq \ell \leq k$ must obtain object $t^{\ell *}$ from their neighbor $T^{\ell *}$, who only accepts objects in $P := \{a_v : v \in V\}$. Thus, there must be k objects in total within P that move from their branch to the T branch in order to reach an agent T^ℓ. So far, the necessary swaps are those between X and Y (one swap), the swaps between each T^ℓ and $T^{\ell *}$ (k swaps), and the swaps along the path $[T, T^k, \ldots, T_1, Y]$ ($k+1$ swaps), so in total $\mathbf{2k+2}$ **swaps**.

Consider an object $a_v \in P$ which must move to an agent T^ℓ. This object must follow the path $[A_v, A_v^{k-1}, \ldots, A_v^1, Y, T^1, \ldots, T^\ell]$. Consider first the sub-path $[A_v^1, Y, T^1, \ldots, T^\ell]$, from the moment where object a_v reaches agent A_v^1. This agent only accepts to swap it against object $y^{[v]}$, therefore agent Y must previously perform a swap with agent $Y^{[v]}$ (**k swaps** in total by considering the k chosen objects in P). Observe that the j^{th} object in P entering in the T^ℓ's branch must reach agent T_{k+1-j}, otherwise an object should pass twice by the same agent, which contradicts the rationality assumption of the swaps. Therefore, by construction of the preferences, the objects in P must go into the T^ℓ's branch by increasing order of indices. By rationality of the swaps, agent Y accepts in exchange of object a_v in the swap with agent T^1 only objects coming from the T^ℓ's branch, and thus only objects in $\{t^{1[w]} : w \geq v\}$ that do not block the future swaps (typically $t^{1[v]}$ is appropriate). Therefore, T^1 must perform an exchange with one of the $T^{1[w]}$ before a_v. Observe that this also holds for the other agents in this branch and thus concerns all the agents T^ℓ for $1 \leq \ell \leq k+1-j$ if a_v is the j^{th} object of P to come into the branch. By counting the swaps between each such T^ℓ and one agent $T^{\ell[w]}$, and the swaps along the path $[Y, T_1, \ldots, T_{k+1-j}]$ for each object in P, we obtain $\mathbf{k(k+1)}$ **swaps**.

Now consider the first part where object a_v moves to agent Y from A_v along the path $[A_v, A_v^{k-1}, \ldots, A_v^1]$. The conditions are similar to those on the T^ℓ's branch. Each agent A_v^ℓ on the path only accepts object $y^{[v]}$ or $a^{\ell-1*}$ in return of giving object a_i, and agent A_v only prefers a^{k*} to a_v. Since for A_v^1 the preferred object is necessarily $y^{[v]}$, the other agents must obtain $a_v^{\ell-1*}$. However, all the agents $A_v^{\ell*}$, need an object within $D_v := \{u_v^{\delta_v(d)} : 1 \leq d \leq d_v\}$ to give object $A_v^{\ell*}$ in a rational swap. It follows that each agent A_v^ℓ on the path must get object $a_v^{\ell-1*}$ and previously an object within D_v for letting pass object a_v to agent A_v^1. Thus, $k-1$ objects within D_v must be chosen to come into the A_v's branch. Once it is done, the remaining swaps are all the swaps between A_v^ℓ and $A_v^{\ell*}$ which lead to $\mathbf{k(k-1)}$ **swaps** in total, and the swaps for making object a_v reach agent A_v^1 along the path $[A_v, A_v^{k-1}, \ldots, A_v^1, Y]$, leading to $\mathbf{k^2}$ **more swaps**.

Now, consider an object u_v^{vw} (or u_v^{wv} depending on the order) which is chosen to come into the A_v's branch. Similarly as in the T^ℓ's branch, by construction of the preferences, the objects in D_v must arrive by increasing lexicographical order, and if u_v^{vw} is the j^{th} object in D_v which enters in the A_v's branch, then it must come to agent A_v^{k-j}. Moreover, each agent A_v^ℓ on the path $[A_v^1, \ldots, A_v^{k-j}]$ must previously make a swap with an auxiliary agent $A_v^{\ell[\delta_v(d)]}$ for $1 \leq d \leq d_v$ that does not block the future swaps, typically with $A_v^{\ell[vw]}$ (or $A_v^{\ell[wv]}$ depending on the order). Therefore, by combining the swaps along $[Y, A_v^1, \ldots, A_v^{k-j}]$ and the swaps between each A_v^ℓ and one agent in $A_v^{\ell[\delta_v(d)]}$, we obtain in total $\mathbf{k^2(k-1)}$ **swaps**. To sum up, so far, we can count $k^3 + 2k^2 + 3k + 2$ necessary swaps. Therefore, it remains in the budget exactly $2k(k-1)$ swaps. By construction of the preferences, a previous swap between Y and the auxiliary agent $Y^{[vw]}$ is necessary to make a first swap between Y and U_v^{vw} occur. Observe that once an object u_v^{vw} is left from the U^{vw} branch, no other agent $U_{v'}^{v'w'}$ can swap with Y because the swap with the auxiliary agent is not possible. The only possibility is the swap between U_v^{vw} and U_w^{vw}, and then between U_v^{vw} and Y. This leads to the obligation of choosing both objects u_v^{vw} and u_w^{vw} to make them pass to the branch of A_v, and we need four swaps to do it. Hence, with our remaining budget, we can only select $k(k-1)/2$ branches U^{vw} which correspond to an edge. Since the chosen objects allow validating $k-1$ incident edges of k vertices, the associated edges in G form a clique of size k.

The same reasoning holds for the makespan. Concerning the length of the sequence, it suffices to observe that the minimal sequence, by performing parallel swaps, is almost totally conditioned by the exchanges of agent Y. Obviously the number of swaps involving Y, precisely $5k(k-1)/2 + 3k + 3$ swaps, is a lower bound for the makespan. But actually one can verify that all the other swaps can be performed in parallel of a swap involving Y. The only exception concerns the last swap between agent T^2 and agent T^1 for exchanging object t. Indeed, once Y gives to agent T^1 object a_v corresponding to the last vertex v of the clique (with respect to the order on vertices), agent T^1 can swap with agent T^{1*} to obtain object t^{1*}, while agent Y prepares in parallel the swap to obtain object t by swapping with agent $Y^{[t]}$. But agent T^1 still needs to swap with agent T^2 to obtain object t and Y has no swap to perform in parallel. Therefore, the makespan is $5k(k-1)/2 + 3k + 4$.

The only possibility to answer true to a no-instance would be to choose more than $k(k-1)/2$ branches associated with an edge among the U^{vw}'s branches in order to validate more agents in the A_v's branches. However, it would imply at least three more swaps for agent Y and thus would increase the makespan, contradiction. $\square$

Theorem 8. *RO-sum is in* W[1].

Proof. Let us prove by induction over i that formula $\mathrm{o}_i(y,z)$ is true iff object z is owned by agent y before i^{th} swap. Formula $\mathrm{o}_1(y,z)$ is true iff $\sigma^0(y,z)$ is true, i.e. y initially owns z. Consider step i, agent y and object z, and suppose that $\mathrm{o}_j(.,.)$ is correct for all $j = 1,\ldots,i-1$. If y has not performed a swap before step i, then the first member of the disjunction in the formula is false, but the second one is true iff z is the object initially owned by y. Otherwise, there exists some x_j or y_j, say x_j, for $j \leq i$, such that $x_j = y$. Therefore, the second member of the disjunction in the formula is false, but the first one is true iff j is the last swap of agent x_i before step i, and $\mathrm{o}_j(y_j,z)$ is true. So, this correctly expresses the current endowment of agent y just before step i if $\mathrm{o}_j(y_j,z)$ is correct, which is the case by induction assumption. Hence, formula $\mathrm{o}_i(y,z)$ correctly expresses whether agent y owns z just before i^{th} swap.

Now, observe that formula $\psi^{k'}$ is true iff the sequence of exchanges between the agents (x_i,y_i) exchanging the objects (a_i,b_i), for $i \in \{1,\ldots,k'\}$, is a sequence of swaps leading to give object x to agent A. Indeed, $\psi^{k'}$ repertories the two conditions for a swap. The two agents x_i and y_i involved in the i^{th} swap must be connected in the social network and the exchange must be rational: x_i must prefer object b_i, which must be the object of agent y_i just before swap i, to object a_i, and vice versa. Moreover, $\psi^{k'}$ imposes that one of the agent involved in the last swap must be agent A and obtain in this last swap object x. Thus, $\psi^{k'}$ expresses the reachability of object x for agent A after exactly k' swaps.

Hence, φ is true iff object x is reachable for agent A after k' swaps, for $k' \leq k$, or x is initially owned by A. $\square$

Proposition 9. *RO-makespan is in* W[1].

Proof. An instance $\mathcal{I}$ of RO with a swap dynamics model $(N, M, \succ, \sigma^0, G)$, agent A and object x, and k as a total number of swaps, is transformed into an instance $\mathcal{I}' = (\mathcal{A}, \varphi)$ of $\mathrm{MC}(\Sigma_1)$. Structure $\mathcal{A}$ is an $(E, \succ, \sigma^0, A, X)$-structure with variables in $N \cup M$, where relations E, $\succ$, σ^0, A and X are defined as follows. The binary relation E over N^2 represents the edge set E. The ternary relation $\succ$ over $N \times M^2$ represents the preferences of the agents, i.e. $\succ(i,a,b)$ means that agent i prefers object a to object b. For the sake of clarity, we write $a \succ_i b$ instead of $\succ(i,a,b)$. The binary relation σ^0 over $N \times M$ represents the initial allocation σ^0, i.e. $\sigma^0(i,z)$ means that agent i is initially endowed with object z.

Finally, the unary relations A and X respectively represent agent A and object x, i.e. $A(y)$ means that y is agent A and $X(y)$ means that y is object x.

The Σ_1-formula φ is defined as $\varphi = \exists x_0 \exists a_0 \exists x_1^1 \exists y_1^1 \exists a_1^1 \exists b_1^1 \ldots \exists x_1^{L_1} \exists y_1^{L_1} \exists a_1^{L_1} \exists b_1^{L_1} \ldots \exists x_k^1 \exists y_k^1 \exists a_k^1 \exists b_k^1 \ldots \exists x_k^{L_k} \exists y_k^{L_k} \exists a_k^{L_k} \exists b_k^{L_k} \exists y \exists b$ $\bigvee_{k' \leq k} \bigvee_{\ell_1 \leq L_1} \cdots \bigvee_{\ell_{k'} \leq L_{k'}} (\psi^{k'}_{\ell_1,\ldots,\ell_{k'}} \wedge \chi^{k'}_{\ell_1,\ldots,\ell_{k'}}) \vee \Big(\sigma^0(x_0, a_0) \wedge A(x_0) \wedge X(a_0)\Big)$ with $L_i = 2^{k-i}$ for all $i \in \{1, \ldots, k\}$, and

$$\psi^{k'}_{\ell_1,\ldots,\ell_{k'}} \equiv \bigwedge_{i=1}^{k'} \bigwedge_{j=1}^{\ell_i} \Big(E(x_i^j, y_i^j) \wedge \Big(b_i^j \succ_{x_i^j} a_i^j\Big) \wedge \Big(a_i^j \succ_{y_i^j} b_i^j\Big) \wedge \mathrm{o}_{ij}(x_i^j, a_i^j) \wedge \mathrm{o}_{ij}(y_i^j, b_i^j) \wedge \bigvee_{j=1}^{\ell_{k'}} \Big(A(x_{k'}^j) \wedge X(b_{k'}^j)\Big)$$

$$\chi^{k'}_{\ell_1,\ldots,\ell_{k'}} \equiv \bigwedge_{i=1}^{k'} \bigwedge_{j=1}^{\ell_i} \bigwedge_{p=j+1}^{\ell_i} \Big((x_i^j \neq x_i^p) \wedge (y_i^j \neq y_i^p) \wedge (x_i^j \neq y_i^p) \wedge (y_i^j \neq x_i^p) \wedge (a_i^j \neq a_i^p) \wedge (b_i^j \neq b_i^p) \wedge (a_i^j \neq b_i^p) \wedge (b_i^j \neq a_i^p)\Big)$$

where $\mathrm{o}_{ij}(.,.)$ for all i, j stands for

$$\mathrm{o}_{ij}(y, a) \equiv \Big(\sigma^0(y, a) \wedge \bigwedge_{p=1}^{i-1} \bigwedge_{r=1}^{\ell_p} (x_p^r \neq y \wedge y_p^r \neq y)\Big) \vee \bigvee_{p=1}^{i-1} \bigvee_{r=1}^{\ell_p} \Big(\Big(\big(\mathrm{o}_{pr}(y_p^r, a) \wedge y = x_p^r\big) \vee \big(y_p^r = y \wedge \mathrm{o}_{pr}(x_p^r, a)\big)\Big) \wedge \bigwedge_{s=p+1}^{i-1} \bigwedge_{t=1}^{\ell_s} (x_s^t \neq y \wedge y_s^t \neq y)\Big)$$

Formula $\psi^{k'}_{\ell_1,\ldots,\ell_{k'}}$ says, similarly as its definition in the proof of Theorem 8, that any exchange must be a swap and that the last swap must give object x to agent A. Formula $\mathrm{o}_{ij}(y, a)$, like in the previous proposition, is true iff agent y gets object a just before j^{th} swap of the i^{th} sequence of parallel swaps. We omit the proof because it is similar to the proof in Theorem 8. Formula $\chi^{k'}_{\ell_1,\ldots,\ell_{k'}}$ says that any parallel swaps must involve different couples of agents. Consequently, φ is true iff x is reachable for A by a sequence of swaps which can be decomposable into a sequence of at most k sets of parallel swaps. We now verify that the size of the formula is in function of k only, by proving that the i^{th} set of parallel swaps has at most $L_i = 2^{k-i}$ parallel swaps. Observe first that at the k^{th} set, the last one, at most one swap is required: the swap between an agent Y and A, that makes A getting object x. All the other parallel swaps are useless for the reachability of x for A. Therefore, there is at most $2^{k-k} = 1$ useful swaps at the k^{th} set. Consider the i^{th} set and assume that at the $(i+1)^{\text{th}}$ set at most 2^{k-i-1} parallel swaps are useful. All the useful parallel swaps at i^{th} set can only involve agents that are involved in all the j^{th} sets for $j \in \{i+1, \ldots, k\}$. There is at most $2\times$ {the maximum number of useful parallel swaps at the $(i+1)^{\text{th}}$ set} such agents, and thus at most 2^{k-i} parallel swaps at the i^{th} set. □

Proposition 10. GLS-*sum/-makespan is in* co-A[2].

Proof. An instance $\mathcal{I}$ of co-GLS with a swap dynamics model $(N, M, \succ, \sigma^0, G)$, agent A, object y, and at most k swaps in total, is transformed into an instance $\mathcal{I}' = (\mathcal{A}, \varphi)$ of $\mathrm{MC}(\Sigma_2)$, known to be A[2]-complete (Thm 1). Co-GLS problem asks whether there exists a reachable allocation, stable under the condition of at most k exchanges, where agent A obtains an object less preferred than y. Structure $\mathcal{A}$ is an $(E, \succ, \sigma^0, A, Y)$-structure with variables in $N \cup M$, where relations E, $\succ$, σ^0, A are defined in the same way as in the proof of Thm. 8, and Y is an unary relation in which $Y(z)$ means that z is object y.

The Σ_2-formula φ is equal to $\exists c \exists z \exists x_0 \exists x_1 \exists y_1 \exists a_1 \exists b_1 \dots \exists x_k \exists y_k \exists a_k \exists b_k$ $\forall x \forall y \forall a \forall b \left(\psi^k \vee \bigvee_{0 \le k' < k} \psi^{k'} \wedge \chi^{k'}\right)$ with

$$\psi^{k'} \equiv \bigwedge_{i=1}^{k'} \left(E(x_i, y_i) \wedge \mathrm{o}_i(x_i, a_i) \wedge \mathrm{o}_i(y_i, b_i) \wedge b_i \underset{x_i}{\succ} a_i \wedge a_i \underset{y_i}{\succ} b_i\right)$$
$$\wedge\ A(x_{k'}) \wedge \mathrm{o}_{k+1}(x_{k'}, c) \wedge Y(z) \wedge z \underset{x_{k'}}{\succ} c$$
$$\chi^{k'} \equiv \left(E(x, y) \wedge \mathrm{o}_{k'+1}(x, a) \wedge \mathrm{o}_{k'+1}(y, b)\right) \rightarrow \left(a \underset{x}{\succ} b \ \vee b \underset{y}{\succ} a\right)$$

where for all i, $\mathrm{o}_i(.,.)$ is defined as in Thm. 8.

It is easy to see that formula φ expresses the reachability of an object c for agent A such that A prefers y to c, either in a stable allocation or within at most k swaps. Hence, co-GLS-sum is correctly translated. The reasoning for co-GLS-makespan is the same, based on the formula of the proof in Prop. 9. □