The Parameterized Complexity of Manipulating Top Trading Cycles

William Phan*[1] and Christopher Purcell†[2]

[1]Department of Economics, North Carolina State University, Raleigh, NC, USA

[2]Department of Mathematics, University of West Bohemia, Pilsen, Czech Republic

December 20, 2021

Abstract

We study the problem of exchange when agents are endowed with heterogeneous indivisible objects, and there is no money. No rule satisfies *Pareto-efficiency*, *individual rationality*, and *strategy-proofness*; there is no consensus in the literature on satisfactory second-best mechanisms. A natural generalization of the ubiquitious Top Trading Cycles (TTC) satisfies the first two properties on the lexicographic domain, rendering it manipulable. We characterize the computational complexity of manipulating this mechanism; we show that it is **W**[**P**]-hard by reduction from MONOTONE WEIGHTED CIRCUIT SATISFIABILITY. We provide a matching upper bound for a wide range of preference domains. We further show that manipulation by groups (when parameterized by group size) is **W**[**P**]-hard. This provides support for TTC as a second-best mechanism. Lastly, our results are of independent interest to complexity theorists: there are few natural **W**[**P**]-complete problems and, as far as we are aware, this is the first such problem arising from the social sciences.

1 Introduction

Gale's Top Trading Cycles (TTC) is ubiquitous [36]. It is used *extensively* as a key building block for the design of mechanisms in real-life applications including kidney exchange, school choice, airplane arrival slots exchange, probabilistic assignment, and mixed-ownership

*wphan@ncsu.edu

†purcell@ntis.zcu.cz

economies.[1,2] For example, in the design of kidney exchange mechanisms, the proposed Top Trading Cycles and Chains mechanism satisfies the usual desirable axiomatic properties and features substantial welfare gains in terms of both number and quality of transplants compared to other mechanisms [34]. The mechanism loses its incentive compatibility properties when moving to more general environments, though, and this is the setting of our paper.

Our precise setting is the problem of exchange between agents endowed with heterogeneous indivisible objects when there is no money. When each agent owns and eventually consumes one object, TTC is the unique mechanism satisfying *Pareto-efficiency*, *individual rationality*, and *strategy-proofness*, and selects the core allocation at each profile of preferences and endowments [33, 23, 38, 43, 25].[3]

Consider now the more general case in which each agent owns and consumes possibly multiple objects. There are numerous applications to this type of model: Dual-donor kidney exchange, liver exchange, employee shift trading, and dynamic assignment mechanisms (i.e. a stream of assignments is a bundle indexed by time).[4] Unfortunately, no mechanism satisfies all three properties [38], and there is no consensus on satisfactory second-best mechanisms. The frontier of viable mechanisms remains a relevant and open question with welfare implications.

Our paper contributes to this discussion. A natural generalization of TTC (which we will refer to also as TTC) satisfies *Pareto-efficiency* and *individual rationality* on the lexicographic domain of preferences, rendering it manipulable. In this generalization, agents attempt to obtain their next most preferred available object and trade objects in cycles—repeatedly until no objects are remaining.

We characterize the computational complexity of manipulating TTC. We argue that a parameterized complexity approach is appropriate: The size of endowments is a natural parameter that may remain bounded or grow much more slowly than the number of participants. In general, the complexity of computational problems may depend heavily on parameters other than the total size of the input. Parameterized complexity thus offers a finer comparison of computational problems and reflects real-world constraints.[5]

Our main result is that the problem of manipulating TTC is $\mathbf{W}[\mathbf{P}]$-hard when parame-

[1] In school choice, students have priorities at various schools instead of ownership, and TTC is the most fair *strategy-proof* and *Pareto-efficient* mechanism [2, 27]. The FAA's mechanism for the exchange of airplane landing slots is improved upon by a variant of TTC [35]. For the more general "mixed-ownership" economies when some objects can be collectively owned, the three properties help characterize a TTC variant [39, 40]. Dropping *individual rationality*, the full class of *Pareto-efficient* and *strategy-proof* mechanisms features agents trading objects in cycles [29, 32].

[2] It also coincides with other well known solution concepts. In probabilistic assignment problems, randomly endowing agents with objects and running TTC is equivalent to Random Serial Dictatorship [1]. For school choice, it coincides with a notion of competitive equilibrium [13].

[3] An allocation is in the core if no group of agents would rather secede and trade amongst themselves. *Strategy-proofness* rules out beneficial manipulation by means of an agent misrepresenting their preference over objects.

[4] See [5, 7, 14, 15, 19, 24]. These models typically add structure or constraints to the problem depending on the application at hand, but in essence feature agents exchanging multiple objects.

[5] For example, see [16] for a parameterized complexity analysis of manipulation of sequential allocation when there are no endowments; and [6] and [45] regarding manipulation of elections through voter control and bribery.

terized by the size of the endowments (Theorem 3.1). Furthermore, we provide a matching upper bound that holds for a large class of preference domains, including the additive domain (Theorem 4.1). Intuitively, as the market grows, even if agents' endowments are below some finite bound, our result shows that manipulation of TTC is computationally difficult. We also consider *group* manipulability, wherein a number of agents can attempt to benefit by making a joint misreport to the mechanism. We show that TTC is **W**[**P**]-hard to manipulate by groups, when parameterized by the size of the group (Theorem 5.1). This does not follow directly from the previous result, but the proof technique is similar. These results give support to TTC as a candidate second-best mechanism satisfying *Pareto-efficiency* and *individual rationality* (on the lexicographic domain).

The result is also of independent interest in computer science. To the best of our knowledge, there are no known problems derived from social science applications known to be **W**[**P**]-complete. Natural problems that are complete for a given class are useful for reductions and help us understand the class. Populating parameterized complexity classes with problems from applications demonstrates the significance of parameterized complexity for practitioners.

We also remark that while the result is encouraging for TTC, complexity of manipulation would be complementary to—as opposed to substituting for—other oft-considered large market properties such as *asymptotic strategy-proofness* where the *benefit* from manipulation vanishes as the market grows [4, 22].

Related Literature

Responding to the incompatibility of *Pareto-efficiency*, *individual rationality*, and *strategy-proofness* as shown by Sönmez [38], the literature has developed in several different directions. Closest to ours, Fujita et al. [17, 18] show that in the conditionally lexicographic domain of preferences TTC selects from the core, implying *Pareto-efficiency* and *individual rationality*, and is **NP**-*hard to manipulate*. When *Pareto-efficiency* is weakened to *range-efficiency*, Pápai [31] provides a characterization of the entire class of mechanisms (the "fixed-deal exchange rules" where possible exchanges are predetermined). Alternatively, Sonoda et al. [41] and Sun et al. [42] cover several interesting domains where agents may have "tops-only", m-chotomous, or correlated (referred to as asymmetric) preferences, and show the extent of compatibility as well as implications for single-valuedness of the core. If agents are restricted a priori to particular sets of trading partners, then compatibility may also be recovered [30, 44]. Finally, other authors also study tradeoffs in an environment where objects have types [20, 21, 26, 37].

2 Model

Let $\mathcal{N}$ be a finite set of *agents*, and $\mathcal{O}$ be a finite set of *objects*. Each agent $i \in \mathcal{N}$ has an *endowment* of objects $\omega_i \subset \mathcal{O}$. An *endowment profile* $\omega = (\omega_i)_{i\in\mathcal{N}} \in (2^{\mathcal{O}})^{\mathcal{N}}$ is a list of endowments for the agents such that $\bigcup_{i\in\mathcal{N}} \omega_i = \mathcal{O}$, and for each $i, j \in \mathcal{N}$ such that $i \neq j$, it holds that $\omega_i \cap \omega_j = \emptyset$. If $\alpha \in \omega_i$, then we say that agent i is the *owner* of α and that $ow(\alpha) = i$. Each agent $i \in \mathcal{N}$ has a *preference relation* R_i over subsets of objects; that is, R_i is a complete, transitive, and anti-symmetric binary relation over $2^{\mathcal{O}}$. Let $\mathcal{R}$ be the set of all such preference relations. We denote the strict component of R_i by P_i, i.e. for each $X, Y \subseteq \mathcal{O}$, it holds that $X\ P_i\ Y$ if and only if $X\ R_i\ Y$ and $X \neq Y$. A *preference profile* $R = (R_i)_{i\in\mathcal{N}} \in \mathcal{R}^{\mathcal{N}}$ is a list of preference relations for the agents.

An *economy* is a tuple $\mathcal{E} = (\mathcal{N}, \mathcal{O}, \omega, R)$. An *allocation* $z = (z_i)_{i\in\mathcal{N}} \in (2^{\mathcal{O}})^{\mathcal{N}}$ for economy $\mathcal{E}$ is a list specifying for each agent a subset of objects, such that $\bigcup_{i\in\mathcal{N}} z_i = \mathcal{O}$, and for each $i, j \in \mathcal{N}$ such that $i \neq j$, it holds that $z_i \cap z_j = \emptyset$. Note that the endowment is an allocation. A *rule* ϕ maps each economy $\mathcal{E}$ to an allocation for $\mathcal{E}$. We denote by $\phi_i(\mathcal{E})$ the allocation of agent i under ϕ at $\mathcal{E}$; if $\phi(\mathcal{E}) = z$, then $\phi_i(\mathcal{E}) = z_i$. We also refer to z_i as an agent's allocation.

Preference domains. We define two natural subdomains of preference relations. An agent may prescribe for each object a utility level; subsequently, for each set of objects, their total utility is the summation of utilities over objects in the set. Formally, a preference relation R_i is *additive* if there is $u_i : \mathcal{O} \to \mathbb{R}$ such that for each $X, Y \subseteq \mathcal{O}$, it holds that $X\ R_i\ Y$ if and only if $\sum_{\alpha\in X} u_i(\alpha) \geq \sum_{\alpha\in Y} u_i(\alpha)$. It is also possible that higher-ranked objects are worth significantly more than lower-ranked objects. In this case, in comparing two sets of objects, an agent first compares the respective top-ranked objects in each set, then the second... and so on, until one set contains a more preferred object. A special case of additivity is thus when R_i is *lexicographic*: for each $X, Y \subseteq \mathcal{O}$, it holds that $X\ P_i\ Y$ if and only if there is $\beta \in \mathcal{O}$ such that 1) $\beta \in X \backslash Y$, and 2) for each $\alpha \in X$ with $\alpha\ P_i\ \beta$, we have $\alpha \in Y$.[6]

For the remainder of the paper, we assume that all agents have lexicographic preferences. Note that in such preferences, the ordering over objects uniquely determines the ordering over sets of obects.[7] Furthermore, no two orderings over objects induces the same ordering over sets of objects. Thus it is sufficient to focus on agents' preferences over just objects. We write $R_i = (a, b, c, \square)$ to mean that within objects, R_i prefers a above each other object, followed by b, followed by c, followed (in arbitrary order) by objects in $\omega_i \setminus \{a, b, c\}$, followed (in arbitrary order) by objects in $\mathcal{O} \setminus (\omega_i \cup \{a, b, c\})$.[8] We write $R_i = (a, b, c)$ to similarly mean that within objects R_i prefers a, b, c, followed by all other objects. Finally, if $\alpha\ R_i\ \beta$

[6] To see that any lexicographic R_i is additive, for each $\alpha \in \mathcal{O}$, let $u_i(\alpha) = 2^{k(\alpha)}$ where $k(\alpha) = |\{\beta \in \mathcal{O} :\ \alpha\ R_i\ \beta\}|$.

[7] For example, $a\ P_i\ b\ P_i\ c$ implies that $\{a, b, c\}\ P_i\ \{a, b\}\ P_i\ \{a, c\}\ P_i\ \{a\}\ P_i\ \{b, c\}\ P_i\ \{b\}\ P_i\ \{c\}$.

[8] Formally, 1) $a\ P_i\ b\ P_i\ c$, 2) for each $x \in \omega_i \setminus \{a, b, c\}$, it holds that $c\ P_i\ x$, and 3) for each each $x \in \omega_i$, and each $y \in \mathcal{O} \setminus (\omega_i \cup \{a, b, c\})$, it holds that $x\ P_i\ y$.

for all β in some subset of objects $\mathcal{O}' \subseteq \mathcal{O}$, we say that agent i *topranks* α in $\mathcal{O}'$ (if $\mathcal{O}' = \mathcal{O}$ we say simply that i topranks α).

Properties of rules. Let $\mathcal{E}$ be an economy. Following standard notation we write (R'_i, R_{-i}) to be the preference profile obtained from R by replacing R_i with $R'_i \neq R_i$. We say that R'_i is a *misreport* for agent i. Let $\mathcal{E}' = (\mathcal{N}, \mathcal{O}, \omega, R'_i, R_{-i})$; a misreport R'_i is *beneficial* under ϕ at $\mathcal{E}$ if $\phi_i(\mathcal{E}')\ P_i\ \phi_i(\mathcal{E})$. A rule ϕ is *strategy-proof* if for each economy $\mathcal{E}$, no agent has a beneficial misreport under ϕ at $\mathcal{E}$. We also consider manipulation by groups. For each $\mathcal{M} \subseteq \mathcal{N}$, let $R'_{\mathcal{M}} = (R'_i)_{i \in \mathcal{M}}$ be a *group misreport* for group $\mathcal{M}$, where for some $i \in \mathcal{M}$, $R'_i \neq R_i$. Let $\mathcal{E}' = (\mathcal{N}, \mathcal{O}, \omega, R'_{\mathcal{M}}, R_{-\mathcal{M}})$. A group misreport $R'_{\mathcal{M}}$ is beneficial under ϕ at $\mathcal{E}$ if for each $i \in \mathcal{M}$, it holds that $\phi_i(\mathcal{E}')\ R_i\ \phi_i(\mathcal{E})$, and for some $j \in \mathcal{M}$, it holds that $\phi_j(\mathcal{E}')\ P_j\ \phi_j(\mathcal{E})$. A rule ϕ is *group strategy-proof* if for each economy $\mathcal{E}$, there is no group $\mathcal{M} \subseteq N$ with a beneficial group misreport under ϕ at $\mathcal{E}$. These properties imply that no agent(s) have incentive to lie about their preferences even if they have full information about the preferences of the other agents. One trivial example of a *group strategy-proof* (and thus *strategy-proof*) rule is the "No Deal" rule—for each $\mathcal{E}$, let $\phi(\mathcal{E}) = \omega$—but this rule is suboptimal in terms of welfare as agents may benefit from trade. We say that an allocation z is *Pareto-dominated* at $\mathcal{E}$ if there is an allocation z' for $\mathcal{E}$ such that for each $i \in \mathcal{N}$, it holds that $z'_i\ R_i\ z_i$, and for some $j \in \mathcal{N}$, it holds that $z'_j\ P_j\ z_j$. We say that an allocation is *Pareto-efficient* at $\mathcal{E}$ if it is not Pareto-dominated at $\mathcal{E}$, and that a rule is *Pareto-efficient* if for each economy $\mathcal{E}$, it recommends an allocation that is *Pareto-efficient* at $\mathcal{E}$. If an agent is worse off after the trade according to their own preference relation, then there is no incentive to take part. A rule is said to be *individually rational* if for each economy $\mathcal{E}$, and each agent $i \in \mathcal{N}$, it holds that $\phi_i(\mathcal{E})\ R_i\ \omega_i$.

We give some examples of rules. The No Deal rule is *individually rational* and (*group*) *strategy-proof*, but not *Pareto-efficient*. The rule where for each economy, we give all of the objects to agent 1 is *Pareto-efficient* and (*group*) *strategy-proof*, but not *individually rational*. Finally, TTC is a rule that is *Pareto-efficient* and *individually rational*, but not (*group*) *strategy-proof*.

Graph theory. A (directed) graph is a set V of vertices and a set E of edges which are ordered pairs of vertices. The edge (u, v) is said to start at u and end at v. If there is such an edge, then we say that u is a *predecessor* of v and that v is a *successor* of u. The *in-degree* of a vertex u, denoted $in(u)$, is the number of edges that end at u; the *out-degree*, denoted $out(u)$, is the number of edges that start at u. A *source* is a vertex of in-degree 0; a *sink* is a vertex of out-degree 0. A (directed) *walk* in a graph is an ordered multiset $(v_1, \ldots, v_j)$ where v_i is a vertex and (v_i, v_{i+1}) is an edge for $1 \leq i < j$. A *path* is a walk where no vertex is repeated. A cycle is a path plus an edge from v_k to v_1.

Top Trading Cycles. We define the rule central to our study by means of an algorithm. In the first step, each agent points to their most preferred singleton object. If there is a cycle, then agents trade along that cycle. The process is repeated with the remaining agents and objects, until no agents remain. For the special case when each agent owns one object, the rule coincides with Gale's TTC of [36].

Top Trading Cycles Algorithm
Input: An economy $\mathcal{E} = (\mathcal{N}, \mathcal{O}, \omega, R)$.
Output: An allocation z for $\mathcal{E}$.

Initialize $V_1 = \mathcal{O}$. For each $t \geq 1$:

Step *t*:

1. Let H_t be the directed graph on V_t with an edge (α, β) in E_t if and only if $ow(\alpha)$ topranks β in V_t.
2. If V_t is empty, then stop.
3. Otherwise, select an arbitrary cycle $(\gamma_1, \gamma_2, \ldots, \gamma_j)$ in H_t.
   (a) Add γ_1 to $z_{ow(\gamma_j)}$, and add γ_{i+1} to $z_{ow(\gamma_i)}$ for $1 \leq i < j$.
   (b) Let $V_{t+1} = V_t \setminus \{\gamma_1, \ldots, \gamma_j\}$.

We refer to Step t.3 in which the objects in the cycle $(\gamma_1, \ldots, \gamma_j)$ are allocated and removed, and the preference graph updated, as *resolving* the cycle $(\gamma_1, \ldots, \gamma_j)$. Figure 1 gives an example of an economy with three agents, and shows the result of resolving one of its trading cycles. In this economy, $N = \{1, 2, 3\}$, $\omega_1 = \{\gamma, \delta\}$, $\omega_2 = \{\alpha\}$, $\omega_3 = \{\beta\}$, $R_1 = (\alpha, \beta, \square)$, $R_2 = (\gamma, \delta, \beta, \square)$, and $R_3 = (\gamma, \square)$. We adopt the convention throughout the paper that a directed edge $\longrightarrow$ between an object α and an object β indicates that, amongst the remaining objects in the economy, the owner of α prefers the object β to any other object. Similarly, $\cdots\cdots\rightarrow$ denotes second preference, $- - - -\rightarrow$ denotes third preference, $-\cdot-\cdot-\rightarrow$ denotes fourth preference, and $-\cdot\cdot-\cdot\cdot-\rightarrow$ denotes fifth preference. We only depict the preferences of an agent up to their endowment and we may omit a *loop* (i.e. a directed edge from a object α to itself) for clarity unless confusion may arise. We have included several examples of economies depicted in this way, and full descriptions of the result of applying TTC, in the Appendix.

When each agent owns only one object, TTC is *strategy-proof*; however, when agents own multiple objects, it is not. In Figure 1, when agent 1 reports the lie $R_1' = (\beta, \alpha, \square)$, TTC assigns them $\{\alpha, \beta\}$. Since β is ranked higher than δ, agent 1 is better off and thus has a beneficial misreport.

In Figure 2 we provide an example of a particular beneficial misreport. The intuition of the example will carry over into the proof of our main result. The idea is as follows: an

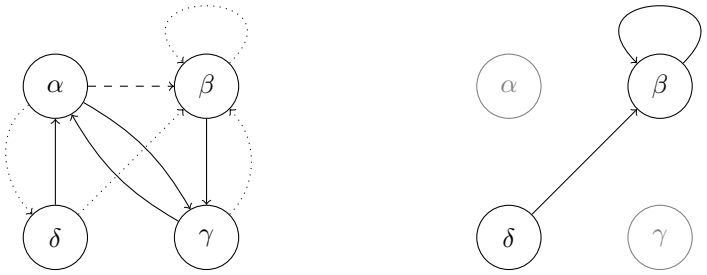

Figure 1: In the economy on the left, the owner of α top ranks γ, whose owner topranks α, so (α, γ) is a trading cycle. On the right we show the result of resolving the cycle.

agent ranks several "undesired" objects first in order to acquire more preferred objects later. There are two forces at play: Agent 1 has multiple opportunities to get α because γ ranks *two* objects in the endowment of agent 1, so getting α can be delayed. Getting x_1 and x_2 first then "unlocks" the opportunity to get β (the path $(\beta, y_1, y_2, e_\beta)$).

Since manipulation of TTC is possible, we next examine if computing a beneficial misreport is computationally tractable. We also note that TTC is *Pareto-efficient* on the lexicographic domain, but not on the additive domain. To see this, consider an economy with two agents 1 and 2 in which $\omega_1 = \{a\}$ and $\omega_2 = \{b, c\}$. Suppose $u_1(a) = 4$, $u_1(b) = 3$, $u_1(c) = 2$, $u_2(a) = 10$, $u_2(b) = 2$, $u_2(c) = 1$. In this economy, TTC allocates each agent their own endowment, but swapping bundles makes both better off. TTC is *individually rational* on both domains.

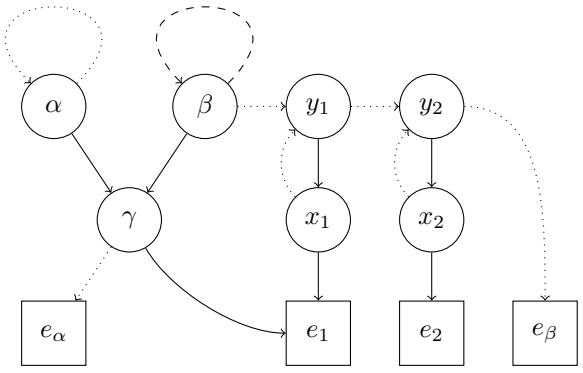

Figure 2: In this economy, agent 1 has a beneficial misreport. Agent 1 owns e_1, e_2, e_α, and e_β. Their true preference relation, not shown, is $(\alpha, \beta, e_\alpha, e_\beta, e_1, e_2)$. If agent 1 reports the truth, their allocation is $\{\alpha, e_\alpha, e_\beta, e_2\}$. If they report $(x_1, x_2, \alpha, \beta)$, then their allocation is $\{\alpha, \beta, x_1, x_2\}$.

Computational complexity. We are interested in the complexity of the following problem:

BENEFICIAL MISREPORT (BM)
INPUT: An economy $\mathcal{E}$.
QUESTION: Does agent 1 have a beneficial misreport under TTC at $\mathcal{E}$?

For simplicity's sake, we always assume that agent 1 is the would-be liar. Since we are mainly interested in proving (conditional) lower bounds on the complexity of manipulating TTC, we focus on the decision version of the problem, which is **NP**-hard as shown by Fujita et al. [17]. Since the outcome of TTC depends only on the reported preferences over singletons, we may assume that the input to BM includes only a total ordering of the objects for each agent apart from agent 1. It is necessary to compare bundles in the consumption space according to the true preference of agent 1 in order to solve BM, and we assume that the preference of agent 1 comes in the form of an algorithm that takes $f(k)\cdot|\mathcal{E}|^{O(1)}$ computational steps to compare two bundles of size k, where $|\mathcal{E}|$ is the number of bits needed to represent $\mathcal{E}$ and f is some computable function. We argue that this is a reasonable assumption. Choosing between two bundles of size k does not depend on the existence of alternative bundles, which implies the existence of an algorithm to compare bundles of size k whose runtime does not depend on the total size of the economy.

Parameterized complexity. For a full treatment of the topic of parameterized complexity, we refer the reader to the textbook by Downey and Fellows [12]. For completeness, we recall some important terminology and definitions.

We define a *parameterized language* to be a subset of $\Sigma^* \times \mathbb{N}$ for some alphabet Σ. If (s, k) is a member of a parameterized language L we say that k is the *parameter*. An algorithm that decides whether (s, k) is a member of L in time $f(k) \cdot |s|^{O(1)}$ for some computable function f is said to be an fpt-algorithm for L. The class of languages decidable by an fpt-algorithm is called **FPT**. We call the decision problem associated with a parameterized language a *parameterized problem* (these terms are commonly used interchangeably).

A *reduction* between (classical) decision problems Π_1 and Π_2 is an algorithm which takes a string s_1 and produces a string s_2 such that s_1 is a yes-instance of Π_1 if and only if s_2 is a yes-instance of Π_2. A problem Π is **NP**-hard if there is a special kind of resource-bounded reduction from any problem in **NP** to Π. There are several choices for the definition of this reduction; the reader unfamiliar with this concept is directed to, e.g., [28]. Proving that Π is **NP**-hard is a *conditional lower bound* for the run time of an algorithm that solves Π. An analogous notion of a reduction and a corresponding conditional lower bound exists in the parameterized setting. An *fpt-reduction* between parameterized languages L_1 and L_2 is an algorithm which takes (s_1, k_1) as input and, in time at most $f(k) \cdot |s|^{O(1)}$ for some computable function f, produces $(s_2, g(k_1))$ as output for some computable function g, such

that $(s_1, k_1) \in L_1$ if and only if $(s_2, g(k_1)) \in L_2$.

The class $\mathbf{W}[1]$ contains $\mathbf{FPT}$ and can be defined as the set of parameterized languages that can be reduced, by an fpt-reduction, to the language associated with the following decision problem.

SHORT NONDETERMINISTIC TURING MACHINE HALTING
INPUT: A nondeterministic Turing machine M.
PARAMETER: An integer k.
QUESTION: Is it possible for M to reach a halting state in at most k steps?

It is considered extremely unlikely that $\mathbf{FPT} = \mathbf{W}[1]$. A parameterized problem Π is $\mathbf{W}[1]$-hard if every problem in $\mathbf{W}[1]$ can be reduced, by an fpt-reduction, to Π. The $\mathbf{W}[1]$-hardness of a given problem is evidence of its intractability. For example, the problem of deciding whether a given graph contains a clique of size k (that is, k vertices every pair of which is adjacent), parameterized by k, is $\mathbf{W}[1]$-complete [11].

The class $\mathbf{W}[1]$ is the first level of the so-called $\mathbf{W}$-hierarchy: $\mathbf{FPT} \subseteq \mathbf{W}[1] \subseteq \mathbf{W}[2] \subseteq \ldots \subseteq \mathbf{W}[t] \subseteq \ldots \mathbf{W}[\mathbf{P}]$. We omit the technical definition of each level of the $\mathbf{W}$-hierarchy but, for example, deciding the existence of a dominating set of size k, parameterized by the size of the solution, is a $\mathbf{W}[2]$-complete problem. We are particularly interested in the class $\mathbf{W}[\mathbf{P}]$, which contains the class $W[t]$ for all t. The following theorem gives a characterization of $\mathbf{W}[\mathbf{P}]$ that we will make use of later.

Theorem 2.1. *([10] but see [8, 9]) A parameterized problem Π is in $\mathbf{W}[\mathbf{P}]$ if and only if there exists a Turing machine M that decides Π such that, on input (x, k), M performs at most $g(k)q(|x|)$ steps and at most $g(k)\log_2|x|$ nondeterministic steps, where g is a computable function and q is a polynomial.*

To state the defining problem of $\mathbf{W}[\mathbf{P}]$, and the problem we will use to prove that BM is $\mathbf{W}[\mathbf{P}]$-hard when parameterized by the size of the endowment to agent 1, we introduce some notions from circuit complexity.

Boolean circuits. A Boolean circuit $\mathcal{C}$ is a directed acyclic graph with one sink. We call the vertices of $\mathcal{C}$ *gates* and the edges *wires*. The sources of $\mathcal{C}$, are labelled with Boolean variables $x_1, \ldots, x_n$ (or $g_1, \ldots, g_n$); we call these *input gates*. The sink, g_N, is called the *output* gate. The remaining gates, denoted $g_{n+1}, \ldots, g_{N-1}$ are called the *internal* gates. Each of the gates $g_{n+1}, \ldots, g_N$ is labeled $\vee, \wedge$ or $\neg$. A $\neg$-gate must have in-degree 1, otherwise the degrees are unrestricted. For brevity, we omit the word "Boolean" and refer simply to "circuits". An *assignment* to a circuit assigns Boolean values to the input gates. We denote Boolean truth by the symbol 1 and falsehood by 0. It is convenient to refer to an assignment as a subset X of the input gates, namely the subset that is assigned 1. We *evaluate* a circuit $\mathcal{C}$ at an assignment X by assigning values to all of the gates recursively, according to their la-

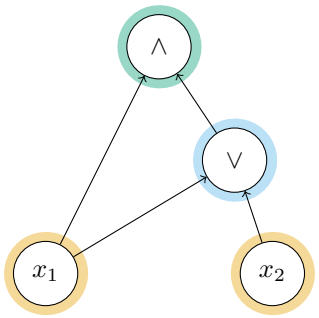

Figure 3: A monotone circuit with two input gates.

bels and to the assigned values of their predecessors. A ¬-gate whose predecessor is assigned the value b is assigned the negation of b, denoted by $\neg b$. An ∨-gate whose predecessors are assigned the values $b_1, \ldots, b_j$ is assigned the logical disjunction of these values. An ∧-gate whose predecessors are assigned the values $b_1, \ldots, b_j$ is assigned the logical conjunction of these values. The value assigned to g_N is the output of $\mathcal{C}$ given the assignment X, denoted $\mathcal{C}(X)$. If $\mathcal{C}(X) = 1$ we say that X is *satisfying* and that X *satisfies* $\mathcal{C}$. The size of X is called its *weight*.

WEIGHTED CIRCUIT SATISFIABILITY (WCSAT)
INPUT: A circuit $\mathcal{C}$.
PARAMETER: An integer k.
QUESTION: Does $\mathcal{C}$ have a satisfying assignment of weight k?

A circuit with no ¬-gates is said to be *monotone*. The WCSAT problem is $\mathbf{W}[\mathbf{P}]$-complete even when the input is restricted to monotone circuits [3]. We call this restricted problem MWCSAT.

Observation 2.2. *If a monotone circuit has a satisfying assignment of weight at most k, then it has a satisfying assignment of weight exactly k.*

We introduce the following terminology and a useful corollary of the previous observation. Let x_i be an input gate of a monotone circuit $\mathcal{C}$ and let g_j be a successor of x_i. Let X be an assignment to $\mathcal{C}$. We define an evaluation of $\mathcal{C}$ at X that is *faulty* on the wire (x_i, g_j) in the sense that g_j is assigned a value as if x_i were assigned 0, regardless of X. Formally, if g_j is an ∨-gate, then the value it is assigned is the logical disjunction of the values of its predecessors *except* that of x_i; if g_j is an ∧-gate, it is assigned 0. We then proceed with the evaluation in the normal way, and denote the result by $\mathcal{C}_{i,j}(X)$. We may extend the definition of a faulty evaluation to an arbitrary subset W of the wires starting at input gates, and denote the result by $\mathcal{C}_W(X)$.

Observation 2.3. *Let $\mathcal{C}$ be a monotone circuit, W be a subset of the wires starting at the input gates of $\mathcal{C}$, and X be an assignment for $\mathcal{C}$. Then, $\mathcal{C}_W(X) = 1$ implies $\mathcal{C}(X) = 1$.*

3 The Main Result

In this section, we state and prove our main result: BM is $\mathbf{W}[\mathbf{P}]$-hard.

Theorem 3.1. *BM, parameterized by the size of the endowment, is $\mathbf{W}[\mathbf{P}]$-hard.*

We prove the theorem by a reduction from MWCSAT. Let $\mathcal{C}$ be a monotone circuit and recall that we call its input gates $x_1, \ldots, x_n$, its internal gates $g_{n+1}, \ldots, g_{N-1}$, and its output gate g_N. We construct an economy $\mathcal{E}_\mathcal{C}$ with agent 1 having k+2 objects, such that agent 1 has a beneficial misreport if and only if $\mathcal{C}$ has a satisfying assignment of weight k.

In order to clarify the proof, we describe the construction in two stages. First, in Section 3.1, we show how to construct an economy $\mathcal{E}'_\mathcal{C}$ such that a weight-k satisfying assignment for $\mathcal{C}$ can be converted into a beneficial misreport for agent 1. Then, in Section 3.2 we show how to modify $\mathcal{E}'_\mathcal{C}$ to get the economy $\mathcal{E}_\mathcal{C}$ in which the converse holds.

3.1 Constructing Economy $\mathcal{E}'_\mathcal{C}$

In both of our economies, agent 1 is endowed with $k + 2$ objects and every other agent is endowed with exactly one object. To simplify our notation, we identify each such agent with the object they own, whenever this can be done unambiguously. For example, when we say that α topranks γ, we will mean that the owner of the object α topranks the object γ. We represent the preference relations of all agents as a list of singletons.

We now describe the set of objects and agents in $\mathcal{E}'_\mathcal{C}$. Firstly, we include agent 1 and the objects $e_1, \ldots, e_k, e_\alpha, e_\beta$ which are each in the endowment to agent 1. For each input gate x_i we add objects $x_i^1, \ldots, x_i^{out(i)}$, and we say these objects *represent* x_i; to simplify our notation, we will write x_i^* instead of $x_i^{out(i)}$. For each gate g_p with $n < p < N$ we add objects $g_p^1, \ldots, g_p^{out(p)}$ and also $h_p^1, \ldots, h_p^{in(p)}$. Only the objects $g_p^1, \ldots, g_p^{out(p)}$ are said to represent g_p; the remaining objects are called the *auxiliary* objects of g_p. For the output gate g_N, we add a special object y which represents g_N in this case. We add auxiliary objects $h_N^1, \ldots, h_N^{in(N)}$ for g_N just as for the other gates. Finally, we add objects $\alpha, \beta, \gamma, \gamma_1, \ldots, \gamma_k$.

We will construct $\mathcal{E}'_\mathcal{C}$ so that TTC allocates agent 1 the bundle $\{\alpha, e_\alpha, e_\beta, e_k, \ldots, e_2\}$. Furthermore, agent 1 will have preference relation $R_1 = (\alpha, \beta, e_\alpha, e_\beta, e_k, e_{k-1}, \ldots, e_1)$. Note that each bundle that agent 1 prefers to the allocation they receive when reporting their true preferences contains both α and β by the definition of lexicographic preferences.

The key intuitive idea behind our construction is that if (say) $\{x_{i_1}, \ldots, x_{i_k}\}$ is a satisfying assignment to $\mathcal{C}$, then agent 1 can benefit by reporting $R'_1 = (x^*_{i_1}, \ldots, x^*_{i_k}, \alpha, \beta, \square)$. If $\{x_{i_1}, \ldots, x_{i_k}\}$ is not a satisfying assignment, and agent 1 reports R'_1, then we want y and β to ultimately form a trading cycle when we run TTC; that is, agent 1 is unable to obtain

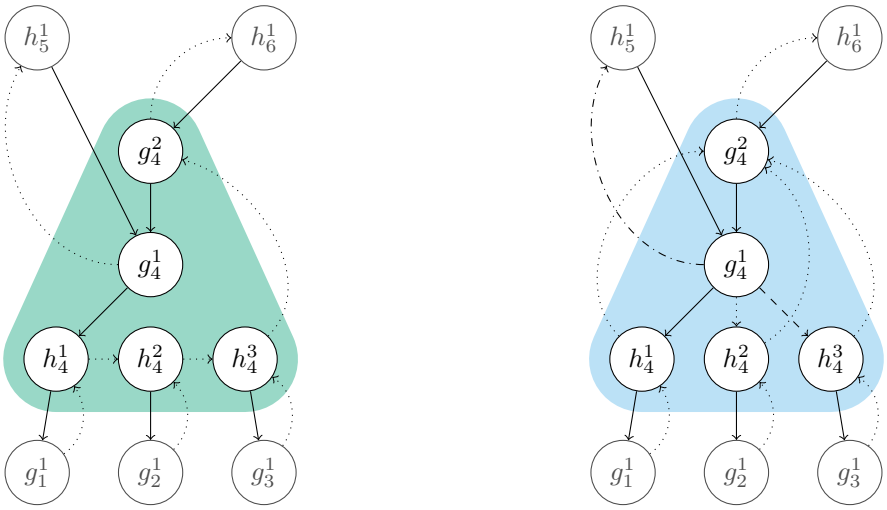

Figure 4: Left: an $\wedge$-gate gadget; Right: an $\vee$-gate gadget. In both cases the gate is g_4; its predecessors are g_1, g_2, and g_3; its successors are g_5 and g_6. It happens that g_4 is the first successor of each of its predecessors, and also the first predecessor of each of its successors.

β. On the other hand, if it is a satisfying assignment, we want that $\gamma_1, \ldots, \gamma_k$ and y are in trading cycles *without* β, so that the allocation of agent 1 will be $\{\alpha, \beta, x^*_{i_1}, \ldots, x^*_{i_k}\}$. We will show that if there exists a beneficial misreport for agent 1, then there exists one of the form R'_1.

We now describe the preference relations of the other agents. In order to keep our notation simple, we refer to the agents by the name of their object. Since each agent, except agent 1, owns one object this will be unambiguous. Let

$$\begin{aligned} R_\alpha &= (\gamma, \square), \\ R_\beta &= (\gamma, \gamma_1, \ldots, \gamma_k, y, e_\beta, \square), \\ R_\gamma &= (e_1, \ldots, e_k, e_\alpha, \square), \text{ and} \\ R_{\gamma_i} &= (e_i, \beta, \square) \end{aligned}$$

for each $i \in \{1, \ldots, k\}$. These preference relations imply that the allocation to agent 1 when they report the truth R_1 is as stated above: Observe that (α, γ, e_1) is a trading cycle. When this cycle is resolved, β topranks γ_1 and vice versa, creating a trading cycle; so β is not in the allocation for agent 1.

We now define the preferences of the agents owning objects representing x_i. In the definitions below, g_p is a successor of x_i. According to some arbitrary ordering over the predecessors and successors of each gate, g_p is the jth successor of x_i and, in turn, x_i is the qth predecessor of g_p. Let

$$R_{x_i^j} = (\gamma_1, \ldots, \gamma_k, h_p^q, \square) \quad \text{if } j = 1 \text{ and}$$
$$R_{x_i^j} = (x_i^{j-1}, h_p^q, \square) \qquad \text{if } j > 1.$$

We now describe the preferences of agents that represent internal gates and their auxiliary agents. See Figure 4 for an example of the construction.

In the following definitions, g_p is an $\wedge$-gate with a predecessor g_i (which may possibly be an input gate x_i) and a successor g_w. The gate g_i is the qth predecessor of g_p, and g_p is the jth successor of g_i. The gate g_w is the uth successor of g_p, and g_p is the vth predecessor of g_w. Let

$$R_{h_p^q} = (g_i^j, h_p^{q+1}, \square) \qquad \text{if } q < in(p),$$
$$R_{h_p^q} = (g_i^j, g_p^{out(p)}, \square) \quad \text{if } q = in(p),$$
$$R_{g_p^u} = (h_p^1, h_w^v, \square) \qquad \text{if } u = 1, \text{ and}$$
$$R_{g_p^u} = (g_p^{u-1}, h_w^v, \square) \qquad \text{if } 1 < u \leq out(p).$$

The preference relations of agents representing $\vee$-gates are similar. Now let g_p be an $\vee$-gate with a predecessor g_i (which may possibly be an input gate x_i) and a successor g_w. As before, g_i is the qth predecessor of g_p, g_p is the jth successor of g_i, g_w is the uth successor of g_p and g_p is the vth predecessor of g_w. Let

$$R_{h_p^q} = (g_i^j, g_p^{out(p)}, \square),$$
$$R_{g_p^u} = (h_p^1, \ldots, h_p^{in(p)}, h_w^v, \square) \quad \text{if } u = 1, \text{ and}$$
$$R_{g_p^u} = (g_p^{u-1}, h_w^v, \square) \qquad \text{if } 1 < u \leq out(p).$$

Finally we deal with the output gate g_N. The preference relations of the auxiliary agents for g_N are similar to those above, simply replace g_N^1 with y in the definitions (think of g_N as a gate with one output, and y as a relabeling of g_N^1). The preference relation of y is simply $(h_N^1, \beta, \square)$ if g_N is an $\wedge$-gate or $(h_N^1, \ldots, h_N^{in(N)}, \beta, \square)$ if g_N is an $\vee$-gate.

Figure 5 gives a full example of the construction in the case $k = 1$.[9] The circuit corresponding to the economy in the figure is the circuit with two input gates x_1 and x_2, and an output gate which is an $\wedge$-gate. Agent 1 has $k + 2$ objects denoted by e_α, e_β and e_1, which are depicted as squares in the figure. If agent 1 tells the truth, then they are allocated $\{\alpha, e_\alpha, e_\beta\}$. The preferences of agent 1 are not shown in the figure, though, as the goal of agent 1 is to find a beneficial misreport if one exists. Note that the circuit has no satisfying assignment of weight 1, and that the misreports $R_1' = (x_1^*, \alpha, \beta, \square)$ and $R_1'' = (x_2^*, \alpha, \beta, \square)$ are not beneficial. As $x_1^* = x_1^1$ and $x_2^* = x_2^1$, these misreports give agent 1 $\{x_1^1, \alpha, e_\beta\}$ and $\{x_2^1, \alpha, e_\beta\}$, respectively. We detail the steps of TTC for misreport R_1' in Appendix A.2.[10]

[9] We also provide an additional example of this construction for a circuit with one internal gate in Appendix A.1.

[10] We also provide steps of TTC for a similar construction when the output gate is an $\vee$-gate, instead, in Appendix A.3.

Note that there is a beneficial misreport for agent 1 in the depicted economy (and indeed any economy constructed as we have described). If agent 1 reports $R_1''' = (y, \alpha, \beta, \square)$, then they are allocated $\{y, \alpha, \beta\}$. We return to this issue in the next section. For now, we prove that the strategy of obtaining a set of objects corresponding to an assignment produces a beneficial misreport if and only if the assignment is satisfying.

Proposition 3.2. *Let $X = \{x_{i_1}, \ldots, x_{i_k}\}$ be an assignment to $\mathcal{C}$ and let $R_1' = (x_{i_1}^*, \ldots, x_{i_k}^*, \alpha, \beta, \square)$ be a misreport. Then R_1' is beneficial if and only if X is satisfying.*

Proof. We run TTC on $\mathcal{E}_{\mathcal{C}}'$ with agent 1 misreporting R_1'. Intuitively, a particular (non-output) gate in $\mathcal{C}$ has value 1 in the evaluation of the assignment if and only if the objects representing the gate are traded in the same cycle. Once all the objects representing a gate are traded we say that the gate is *fixed*. If all the objects representing a gate were traded in the same cycle, we will say that the gate is fixed *with value* 1, and with value 0 otherwise. Recall that we may choose the order in which to resolve cycles. We first resolve cycles including objects representing input gates. After dealing with the input gates, we deal with gates whose predecessors are all fixed.

In the initial preference graph, there is just one trading cycle $(e_1, x_{i_1}^*, \ldots, x_{i_1}^1, \gamma_1)$. We resolve this cycle, and fix x_{i_1} with value 1. The updated preference graph contains several trading cycles; we proceed by resolving the cycle $(e_j, x_{i_j}^*, \ldots, x_{i_j}^1, \gamma_j)$ and fixing x_{i_j} with value 1 for $j = 2, \ldots, k$. At the end of this process, each gate in X is fixed with value 1.

Now, for each gate x_i that is not in X, there is a trading cycle (x_i^1, h_p^q), where g_p is the 1st successor of x_i, and x_i is the qth predecessor of g_p. We resolve this cycle, and update the preference graph. There is now a trading cycle $(x_i^2, h_{p'}^{q'})$, where $g_{p'}$ is the 2nd successor of x_i, and x_i is the q'th predecessor of $g_{p'}$. Resolving this cycle and repeating for the remaining successors of x_i, we then fix x_i with value 0. We do the same for all the input gates outside of X, and so all of the input gates are fixed.

We continue by fixing any $\wedge$-gate whose predecessors were all fixed. Let g_p be such a gate; we show that g_p is fixed with value 1 if and only if all its predecessors were fixed with value 1. Suppose first that each predecessor was fixed with value 1. Let the qth predecessor of g_p be g_j; since it was fixed with value 1 the objects representing it were traded in the same cycle. In particular, if g_p is the ℓth successor of g_j, the object g_j^ℓ was traded in the cycle with $g_j^1, \ldots, g_j^{out(j)}$ and not with h_p^q. Therefore, h_p^q is in the current preference graph, and currently topranks h_p^{q+1}, unless $q = in(p)$, in which case it topranks $g_p^{out(p)}$. Thus there is a trading cycle $(g_p^{out(p)}, \ldots, g_p^1, h_p^1, \ldots, h_p^{in(p)})$. We resolve this cycle, fixing g_p with value 1 as required. Now suppose that j' is the smallest integer such that the predecessor $g_{j'}$ was fixed with value 0, and suppose that $g_{j'}$ is the q'th predecessor of g_p. The the objects representing $g_{j'}$ were not traded in the same cycle. We claim that $(g_{j'}^{\ell'}, h_p^{q'})$ was a trading cycle already resolved in a previous step of TTC. Indeed, if $g_{j'}$ was in fact an input gate, then this follows from the previous paragraph. Since there must be a gate all of whose predecessors are input gates, the claim will follow by induction. Since $h_p^{q'}$ has been traded, $(h_p^{q'-1})$ is a trading cycle

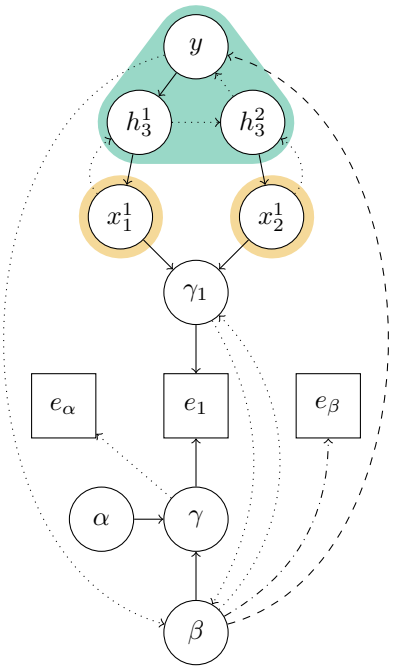

Figure 5: The economy $\mathcal{E}'_{\mathcal{C}}$ where $\mathcal{C}$ is the circuit with one $\wedge$-gate g_3 and two input gates x_1 and x_2, with $k = 1$ (agent 1 has endowment size $k + 2 = 3$). The true preference of agent 1, not shown, is $(\alpha, \beta, e_\alpha, e_\beta, e_1)$.

in the current preference graph, which we resolve. After repeating this step $q' - 1$ times, h_p^1 is traded. Suppose the first successor of g_p is g_r and g_p is the sth predecessor of g_r. Then g_p^1 topranks h_r^s in the current preference graph, since h_p^1 was traded. Since h_r^s topranks g_p^1 by definition, there is a trading cycle (g_p^1, h_r^s). Resolving this cycle completes the induction to give the claim. Thus, the objects representing g_p are not traded in the same cycle, which is what we wanted: g_p is fixed with value 0.

The procedure is similar for $\vee$-gates. Let g_p be such a gate; we show that g_p is fixed with value 0 if and only if all its predecessors were fixed with value 0. Suppose first that each predecessor was fixed with value 0. Let the qth predecessor of g_p be g_j, and suppose g_p is the ℓth sucessor of g_j. By the same inductive reasoning as in the previous paragraph, we may assume that (g_j^ℓ, h_p^q) was a trading cycle resolved in a previous step. Let the first successor of g_p be g_r and suppose that g_p is the sth predecessor of g_r. Then g_p^1 topranks h_r^s in the current preference graph, since $h_p^1, \ldots, h_p^{in(p)}$ were removed. Since h_r^s topranks g_p^1 by definition, there is a trading cycle (g_p^1, h_r^s). After resolving this cycle, we see that the objects representing g_p are not traded in the same cycle, which is what we wanted: g_p is fixed with value 0. Now let j' be the smallest integer such that the predecessor $g_{j'}$ was fixed with value 1, and suppose that $g_{j'}$ is the q'th predecessor of g_p. The the objects representing $g_{j'}$ were traded in the same cycle. As in the previous paragraph, $h_p^{q'}$ is in the current preference graph, but since

g_p is an $\vee$-gate, $h_p^{q'}$ topranks $g_p^{out(p)}$. Thus there is a trading cycle $(g_p^{out(p)}, \ldots, g_p^1, h_p^{q'})$. We resolve this cycle, fixing g_p with value 1 as required.

After repeating the above for all gates, we may observe that, if X is satisfying, then g_N will be fixed with value 1, and the object y was in a cycle with its auxiliary objects and not in the remaining preference graph. Otherwise, y will be fixed with value 0, and there will be a trading cycle (y, β), which will prevent agent 1 from getting β.

To complete the proof, we consider α and β. Since each of the objects in the endowment to agent 1 has been traded, γ topranks e_α. Agent 1 now topranks α, so there is a trading cycle $(\alpha, \gamma, e_\alpha)$. We resolve this cycle, which adds α to the allocation to agent 1 as required. Consider the current preference graph. Since γ was traded when the previous cycle was resolved, β topranks e_β if and only if y was already traded. Therefore, if X is satisfying, we will have a trading cycle (β, e_β) and resolving it will add β to the allocation to agent 1. Otherwise, we will have a trading cycle (β, y) and the allocation of agent 1 does not include β. This completes the proof that R_1' is beneficial if and only if X is satisfying. □

We have proved that if $\mathcal{C}$ has a satisfying assignment of weight k then agent 1 has a beneficial misreport in $\mathcal{E}_\mathcal{C}'$. However, the converse may not hold. Indeed $R_1'' = (y, \gamma_2, \ldots, \gamma_k, \alpha, \beta, \square)$ is a beneficial misreport for $\mathcal{E}_\mathcal{C}'$ for any $\mathcal{C}$ and thus the main theorem is not yet proved. Before we overcome this difficulty, we make the following observation.

Observation 3.3. *Let $R_1' = (\eta_1, \ldots, \eta_k, \alpha, \beta, \square)$ be such that each η_j is either γ_j or an object $x_{i_j}^\ell$ representing an input gate. Then if R_1' is beneficial so is $R_1'' = (x_{i_1}^*, \ldots, x_{i_k}^*, \alpha, \beta, \square)$.*

Proof. By Proposition 3.2, running TTC with R_1'' is equivalent to evaluating $\mathcal{C}$ at the assignment $X = \{x_{i_1}, \ldots, x_{i_k}\}$. Running TTC with R_1' is equivalent to a faulty evaluation of $\mathcal{C}$ at X, with some faulty wires starting at input gates. The claim follows from Observation 2.3. □

3.2 Constructing Economy $\mathcal{E}_\mathcal{C}$

In order to complete the proof of Theorem 3.1, we modify $\mathcal{E}_\mathcal{C}'$ to obtain the final construction $\mathcal{E}_\mathcal{C}$. Let the *circuit structure* of $\mathcal{E}_\mathcal{C}'$ be the set of objects representing gates, the set of auxiliary objects, and the preference relations of their respective owners. In the previous section, we observed that it is possible for agent 1 to destroy the circuit structure in $\mathcal{E}_\mathcal{C}'$, e.g. by top ranking y, so that running TTC is not equivalent to evaluating some assignment to $\mathcal{C}$. We will show how to prevent this outcome by duplicating the circuit structure in $\mathcal{E}_\mathcal{C}'$ k+1 times.

For each object in the circuit structure of $\mathcal{E}_\mathcal{C}'$ we include k+1 copies in $\mathcal{E}_\mathcal{C}$. We distinguish these copies by adding a label from 1 to k+1 to their subscripts. For instance, the copies of g_i^j are denoted $g_{i,1}^j, \ldots, g_{i,k+1}^j$. The copies of the output gates are denoted $y_1, \ldots, y_{k+1}$. We will use x_i^* to denote only the last copy of $x_i^{out(i)}$; that is, $x_i^* = x_{i,k+1}^{out(i)}$.

We now describe the preference relations of the copies. For the ℓth copy of the objects representing an internal gate or any auxiliary object (including those of the output gate), we

obtain a preference relation from that of its original in $\mathcal{E}'_{\mathcal{C}}$ by adding ℓ to the subscript and putting all objects that are not in the ℓth copies after itself. Similarly, $R_{y_\ell} = (h^1_{N,\ell}, \beta, \square)$ if g_N is an $\wedge$-gate and $R_{y_\ell} = (h^1_{N,\ell}, \ldots, h^{in(N)}_{N,\ell}, \beta, \square)$ if g_N is an $\vee$-gate.

We treat the copies of objects representing input gates differently. The first copy of x^1_i has a similar preference relation to its original, but the ℓth copy top ranks the $\ell-1$th copy of $x_i^{out(i)}$. The copies of x^j_i for $j > 1$ have similar preference relations to their originals, the effect of which is that all the copies of all the objects representing the input gate x_i form a directed path in the graph of first preferences. As in the original definitions, in what follows g_p is a successor of x_i. In particular, it is the jth successor of x_i and x_i is the qth predecessor of g_p. Let

$$\begin{aligned}
R_{x^j_{i,1}} &= (\gamma_1, \ldots, \gamma_k, h^q_{p,1}, \square) && \text{if } j = 1,\\
R_{x^1_{i,\ell}} &= (x^{out(i)}_{i,\ell-1}, h^q_{p,\ell}, \square) && \text{if } 1 < \ell \leq k+1, \text{ and}\\
R_{x^j_{i,\ell}} &= (x^{j-1}_{i,\ell}, h^q_{p,\ell}, \square) && \text{if } j > 1 \ \ \& \ \ 1 \leq \ell \leq k+1.
\end{aligned}$$

An example of the result of this duplication process is given in Figure 6.

The final modification that we make is to change the preference relation of β to $R_\beta = (\gamma, \gamma_1, \ldots, \gamma_k, y_1, \ldots, y_{k+1}, e_\beta, \beta, \square)$.

Proof of Theorem 3.1. Let $\mathcal{C}$ be a monotone circuit and $\mathcal{E}_{\mathcal{C}}$ be the economy obtained from $\mathcal{E}'_{\mathcal{C}}$ by the modifications described above. We will show that $\mathcal{C}$ has a satisfying assignment of weight k if and only if agent 1 has a beneficial misreport in $\mathcal{E}_{\mathcal{C}}$. Furthermore, $\mathcal{E}_{\mathcal{C}}$ can be constructed in time $g(k) \cdot |\mathcal{C}|$. We will conclude that there is a parameterized reduction from MONOTONE WCSAT to BM, as required.

Let $X = \{x_{i_1}, \ldots, x_{i_k}\}$ be an assignment for $\mathcal{C}$.

Claim 3.4. *$R'_1 = (x^*_{i_1}, \ldots, x^*_{i_k}, \alpha, \beta, \square)$ is a beneficial misreport for agent 1 in $\mathcal{E}_{\mathcal{C}}$ if and only if R'_1 is beneficial for agent 1 in $\mathcal{E}'_{\mathcal{C}}$.*

The above claim follows by construction of $\mathcal{E}_{\mathcal{C}}$. To see this, consider the first step of TTC for this economy. As $x^*_{i_1} = x^{out(i_1)}_{i_1,k+1}$, the only cycle that forms is

$$(e_1, x^{out(i_1)}_{i_1,k+1}, \ldots, x^1_{i_1,k+1}, x^{out(i_1)}_{i_1,k}, x^{out(i_1)-1}_{i_1,k}, \ldots, x^1_{i_1,1}, \gamma_1).$$

Note then that for each copy of the circuit structure, the respective input gate x_{i_1} takes value 1 (i.e. for copy $\ell < k+1$, the objects $\{x^{out(i_1)}_{i_1,\ell}, \ldots, x^1_{i_1,\ell}\}$ are in the cycle as in Proposition 3.2).[11] A symmetric statement holds for each other input gate $\{x_{i_2}, \ldots, x_{i_k}\}$. After we resolve each such trading cycles, we can run TTC in any of the copies of the circuit structure independently of the others in a way that is identical to the proof of Proposition 3.2. As a

[11] For intuition, see Figure 6. There, if agent 1 topranks $x^1_{1,6}$, then $\{x^1_{1,6}, \ldots, x^1_{1,1}\}$ are included in the first trading cycle. Thus, for each of the 6 copies, the respective objects representing input gate x_1 are removed. This indicates that input gate x_1 takes value 1 in each copy.

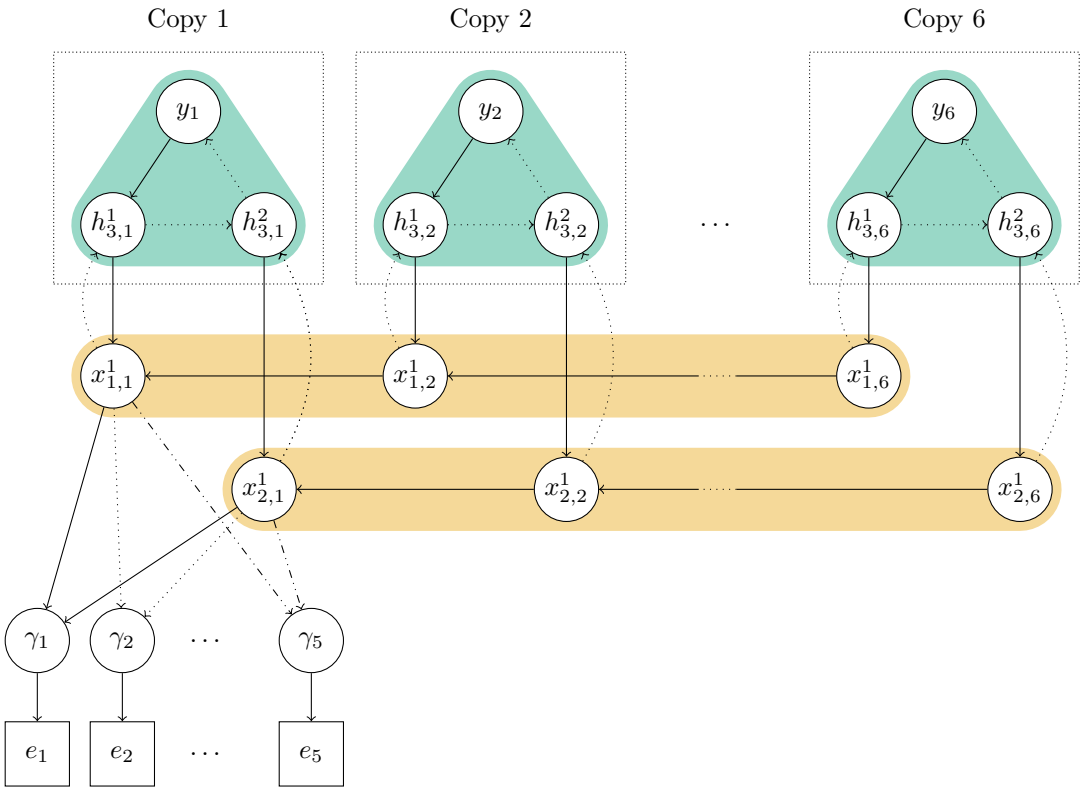

Figure 6: The economy $\mathcal{E}_\mathcal{C}$ where $\mathcal{C} = x_1 \wedge x_2$, as in Figure 5, in which $k = 1$. Here, $k = 5$, chosen to illustrate the general construction. We omit the objects α, β, γ, e_α, and e_β for clarity.

corollary, if $\mathcal{C}$ has a satisfying assignment of weight k then agent 1 has a beneficial misreport in $\mathcal{E}_\mathcal{C}$ as required.

We must show that if there is some beneficial misreport for agent 1, then there is a satisfying assignment for $\mathcal{C}$ of weight k. Let R'_1 be such a misreport, and consider running TTC on $\mathcal{E}_\mathcal{C}$ with agent 1 misreporting R'_1. We define a corresponding assignment for $\mathcal{C}$ later. Let $A = \{\alpha, \beta, \delta_1, \ldots, \delta_k\}$ be the allocation of agent 1 when they report R'_1. We may assume that the first $k+2$ objects in R'_1 coincide with A; therefore, for each object $\epsilon \in \mathcal{O} \setminus A$, and each $\epsilon' \in A$, we have that $\epsilon' \; R'_1 \; \epsilon$. We may also assume, without loss of generality, that $\delta_j \; R'_1 \; \delta_{j+1}$ for all j. We make the following useful observations about A and R'_1:

Observation 3.5. $\gamma \notin A$.

Proof. $\gamma \in A \Rightarrow \alpha \notin A$ by construction. □

Observation 3.6. $\alpha \; R'_1 \; \beta$.

Proof. Since α topranks γ, they are either traded in the same cycle or γ is traded before α. But if γ is traded before α then (α) will be a trading cycle, which is a contradiction. □

Observation 3.7. *δ_i R'_1 α for $1 \leq i \leq k$.*

Proof. Let i be the smallest integer such that α R'_1 δ_i. By construction, the trading cycle containing α must be (α, γ, e_i). After removing this cycle, (β, γ_i) is a trading cycle, which is a contradiction to $\beta \in A$. □

The above observations show that R'_1 takes the form $(\delta_1, \ldots, \delta_k, \alpha, \beta, \square)$ and that each δ_j is in some copy of the circuit structure (or $\delta_j = \gamma_j$, although we will see below that this effectively represents giving a lower weight assignment to $\mathcal{C}$; in this case the result will follow from Observation 2.2).

Intuitively, agent 1 makes k trading cycles each in some copy of the circuit structure of $\mathcal{E}_\mathcal{C}$ representing the circuit, and then tries to get α and β. Since there are k+1 copies of the circuit structure, there is at least one copy where $\delta_1, \ldots, \delta_k$ are not objects in the circuit structure. In this copy, the trading cycles including its objects only interact with input gates. Thus, its circuit structure is preserved, and agent 1 needs to avoid a trading cycle between its copy of y and β. We describe formally how to extract a satisfying assignment of weight k from this interaction below.

We apply TTC according to the following ordering of the trading cycles. For each $j = 1, \ldots, k$, we remove the trading cycle C_j containing δ_j and e_j and update the preference graph. We call these the *initial* cycles. There are at most k input gates x_i such that some copy of some object representing x_i has been removed. We call these $X = \{x_{i_1}, \ldots, x_{i_{k'}}\}$ where $k' \leq k$. Our goal is to prove that X is a satisfying assignment for $\mathcal{C}$. By Observation 2.2, it will follow that $\mathcal{C}$ has a satisfying assignment of weight exactly k.

In order to achieve our goal, we continue our application of TTC in $\mathcal{E}_\mathcal{C}$ with agent 1 reporting R'_1. Observe that there must be some z with $1 \leq z \leq k$+1 such that none of the zth copies of the objects representing internal gates and auxiliary objects have been removed. As we remove trading cycles representing zth copies of objects, we fix the value of the gates similarly to the proof of Proposition 3.2. Let x_i be a gate not in X, and consider the object $x^1_{i,1}$. Since $\gamma_1, \ldots, \gamma_k$ have been removed, and $x^1_{i,1}$ has not, there is a trading cycle $(x^1_{i,1}, h^q_{p,1})$ where the first successor of x_i is the qth predecessor of g_p. If $z \neq 1$, we remove this cycle, and then there is a trading cycle $(x^2_{i,1}, h^{q'}_{p',1})$ which we remove, and so on up to $(x^{out(i)}_{i,1}, h^{q''}_{p'',1})$. Now there is a trading cycle $(x^1_{i,2}, h^q_{p,2})$ which we remove. We continue this process until we reach $x^1_{i,z}$. When we resolve the cycles $(x^1_{i,z}, h^q_{p,z}), \ldots, (x^{out(i)}_{i,z}, h^{q''}_{p'',z})$ we *fix* the gate x_i with value 0 (much as we did in the proof of Proposition 3.2). We repeat this process for each $x_i \notin X$.

Now consider a gate $x_i \in X$ (in other words $i \in \{i_1, \ldots, i_k\}$) and consider the zth copies of the objects representing x_i. There are three cases to consider. If all the objects $x^1_{i,z}, \ldots, x^{out(i)}_{i,z}$ were removed in the same initial cycle, then we can simply fix x_i with value 1

and continue. If none of the objects $x^1_{i,z}, \ldots, x^{out(i)}_{i,z}$ have been traded (in other words, if an initial cycle removed $x^q_{i,\ell}, \ldots, x^1_{i,\ell}, x^{out(i)}_{i,\ell-1}, \ldots, x^1_{i,\ell-1}, \ldots, x^{out(i)}_{i,1}, \ldots, x^1_{i,1}$ for some q and for some $\ell < z$) then we can fix x_i with value 0 and repeat the above process, eventually resolving $(x^1_{i,z}, h^q_p)$ where the first successor of x_i is the qth predecessor of g_p, and so on. In the third case, some of the objects $x^1_{i,z}, \ldots, x^{out(i)}_{i,z}$ have been traded and some have not. Note that this can only happen if $\delta_j = x^r_{i,z}$ for some j and r so that there was an initial cycle $(e_j, x^r_{i,z}, x^{r-1}_{i,z}, \ldots, x^1_{i,z}, x^{out(i)}_{i,z-1}, \ldots, x^1_{i,1}, \gamma_j)$. In this case we fix x_i with value 1, but now we have a trading cycle $(x^{r+1}_{i,z}, h^q_{p,z})$, where g_p is the $r+1$th successor of x_i, and x_i is the qth predecessor of g_p. When we resolve this cycle, we will say that g_p received the value 0 from x_i, and that the wire from x_i to g_p is *faulty*. We repeat this process for each r' from $r+1$ to $out(i)$. It will follow that continuing to run TTC in the zth copy of the circuit structure is equivalent to a (possibly faulty) evaluation of the assignment X.

Consider a gate g_p all of whose predecessors are input gates. Following the proof of Proposition 3.2, if g_p is an $\wedge$-gate, then there will be a trading cycle $(h^1_{p,z}, \ldots, h^{in(p)}_{p,z}, g^{out(p)}_{p,z}, \ldots, g^2_{p,z}, g^1_{p,z})$ only if each predecessor of g_p was assigned the value 1. Similarly, if g_p is an $\vee$-gate, there will be a trading cycle $(h^i_{p,z}, g^{out(p)}_{p,z}, \ldots, g^2_{p,z}, g^1_{p,z})$ only if at least one of the predecessors of g_p was assigned the value 1; namely, its ith predecessor. The rest of the proof is identical to Proposition 3.2. Since R'_1 is beneficial, eventually (β, e_β) is a trading cycle. Therefore, (y_z, β) is never a trading cycle; in other words, the output gate g_N must be fixed with value 1 after we run TTC. We conclude that X is a satisfying assignment for $\mathcal{C}$.

It remains only to justify that the construction of $\mathcal{E}_\mathcal{C}$ can be performed in $g(k) \cdot |\mathcal{C}|^{O(1)}$ steps for some computable function g; that is, the reduction described above is truly an fpt-reduction. For each of the N gates of $\mathcal{C}$, at most N objects and agents are created in $\mathcal{E}'_\mathcal{C}$, each with a preference relation that is a list of objects of length at most $k+2$. We duplicate these $k+2$ times in $\mathcal{E}_\mathcal{C}$. This takes at most $f(k) \cdot O(N^2 \log_2 N)$ time for some computable f. Agent 1 has $k+2$ objects and a preference relation of length $k+4$. There are $k+3$ remaining objects and agents which each have a preference relation of length at most $k+4$. Constructing these objects and agents takes $f'(k) \cdot O(\log_2 N)$ time for some computable f'. Combining the above shows that the reduction takes $g(k) \cdot |\mathcal{C}|^{O(1)}$ steps as required. □

4 The Upper Bound

In this section, we show that BM is a member of $\mathbf{W}[\mathbf{P}]$ and is thus $\mathbf{W}[\mathbf{P}]$-complete.

Theorem 4.1. *BM is* $\mathbf{W}[\mathbf{P}]$*-complete.*

Proof. Let $\mathcal{E} = (\mathcal{N}, \mathcal{O}, \omega, R)$ be an economy. We show that there exists a Turing machine M that decides the existence of a beneficial misreport in $\mathcal{E}$ that performs at most $g(k)q(|\mathcal{E}|)$ steps and at most $g(k)\log_2 |\mathcal{E}|$ non-deterministic steps, for some computable g and polynomial q.

By Theorem 2.1, this gives the upper bound that matches the lower bound of the previous section, and thus the result.

Our proof relies on the fact that a beneficial misreport for an agent with an endowment of size k has a canonical form that takes $O(k \lceil \log_2 |\mathcal{O}| \rceil)$ bits to specify. To prove this claim, it is necessary to fix a precise order in which cycles are removed by the TTC algorithm. In particular, for each economy $\mathcal{E}$ we fix a total ordering over the set of its objects $\mathcal{O}$. We can thus refer to the *first* object in $\mathcal{O}$. Observe that for each object α in V_t there is a unique directed walk with no repeated edges starting at α; we call this the *trading walk* starting at α. Since every element of V_t has outdegree 1, this walk must contain a cycle. We define the cycle chosen to be resolved in Step $t.3$ of the TTC algorithm to be the one contained in the trading walk starting at the first object in V_t. We can now define the trading time $tt_{\mathcal{E}}(\alpha)$ of an object α in a run of TTC on $\mathcal{E}$ to be the least integer t such that $\alpha \in V_t \setminus V_{t+1}$. When the economy is unambiguous, we write $tt(\alpha) = tt_{\mathcal{E}}(\alpha)$. The following observation says that, if an agent i receives an object β at a particular step, then all objects that are preferable to β under R_i must have had an earlier trading time.

Observation 4.2. *Let $\mathcal{E}$ be an economy, $i \in \mathcal{N}$ be an agent, and z be the allocation that TTC recommends for $\mathcal{E}$. For each $\beta \in z_i$, and each $\alpha \in \mathcal{O} \setminus \{\beta\}$, if $\alpha \; P_i \; \beta$, then $tt(\alpha) < tt(\beta)$.*

We are now ready to show that every beneficial misreport is equivalent to one that can be specified efficiently, in the sense that they yield the same allocation. In particular, the following claim shows that it is enough to guess the first k objects of a beneficial misreport.

Claim 4.3. *Suppose the allocation of agent 1 under TTC at (R'_1, R_{-1}) is $z_i = \{\gamma_1, \ldots, \gamma_k\}$, and let $R''_1 = (\gamma_1, \ldots, \gamma_k, \square)$. Then the allocation of agent 1 at (R''_1, R_{-1}) is also z_i.*

Proof. Let $\mathcal{E}'$ and $\mathcal{E}''$ be the (otherwise identical) economies in which agent 1 reports R'_1 and R''_1 respectively. For $t = 1, 2, \ldots$ let H'_t and H''_t be the graphs generated by running TTC on $\mathcal{E}'$ and $\mathcal{E}''$ respectively. Let $tt'(\alpha)$ and $tt''(\alpha)$ be the trade times of α during a run of TTC on $\mathcal{E}'$ and $\mathcal{E}''$ respectively. By Observation 4.2, $tt'(\gamma_1) < tt'(\gamma_i)$ and $tt''(\gamma_1) < tt''(\gamma_i)$ for $1 < i \leq k$. Consider $H'_{tt'(\gamma_1)}$. None of the cycles that were resolved to obtain $H'_2, \ldots, H'_{tt'(\gamma_1)-1}$ have included any of $\gamma_1, \ldots, \gamma_k$. Therefore we may assume without loss of generality that the same cycles are resolved to obtain $H''_2, \ldots, H''_{tt''(\gamma_1)-1}$ and thus $H''_{tt''(\gamma_1)}$ and $H'_{tt'(\gamma_1)}$ are identical. Since γ_1 is allocated to agent 1 in $\mathcal{E}'$, it must also be allocated to agent 1 in $\mathcal{E}''$. The claim follows by induction. □

Observe that TTC runs in polynomial time since in each step we reduce the total number of objects by at least one. In particular, there exists a Turing machine M_{TTC} that runs in time $O(|\mathcal{E}|^c)$ for some constant c, takes as input a binary string representing the list of preferences over singletons of each agent, and returns an allocation by running TTC on the economy given by that string. Our Turing machine M will use M_{TTC} as a subroutine. We omit the details but M has three phases.

Phase 1: In this phase, M prepares an input string for M_{TTC}. First, M non-deterministically writes a binary string representing an ordered list of k objects onto one of its tapes, which is the beginning of R_1'. Since there are $|\mathcal{O}|$ objects in total, this string has length $O(k \lceil \log_2 |\mathcal{O}| \rceil)$. The rest of R_1' can be prepared deterministically by ordering the remaining objects by their representation as a binary string. Finally the rest of the input to M_{TTC} is prepared by writing the preferences of the other agents.

Phase 2: In this phase, M runs M_{TTC} on the input written on the tape in the previous phase, and also on the true preferences. It writes the true allocation A and the allocation A' under R_1' on one of its tapes.

Phase 3: In this phase, M compares A and A' according to R_1. If A' R_1 A, then M accepts. Otherwise, M rejects.

Phase 1 uses $O(k \lceil \log_2 |\mathcal{O}| \rceil)$ non-deterministic steps; in particular, it takes no more than $g(k) \log_2 |\mathcal{E}|$ bits of non-determinism. By Claim 4.3 a beneficial misreport will be guessed in Phase 1 if one exists. Listing the remaining elements to complete the construction of R_1' in the deterministic part of Phase 1 takes linear time. Phase 2 takes polynomial time by the definition of M_{TTC}, and Phase 3 takes polynomial time by the definition of BM. Therefore M fulfils the requirements of Theorem 2.1 as required. □

5 Group Manipulation

Consider now the possibility of manipulation by a group of agents. For intuition, we depict a scenario in Figure 7 that extends upon that of Figure 2. Now, instead of only one agent who must obtain an undesirable object to get a more desirable one, we have two agents who each must obtain an undesirable object for each of them to get their respective more desirable objects. In general, a group misreport may consist of agents in the group who simply state their true preference. Thus, in this scenario, if there is a third agent who benefits even if they state their true preference, then there is also a beneficial group misreport for the three agents.

Formally, we consider the following problem:

GROUP MANIPULATION (GM)
INPUT: An economy $\mathcal{E}$, an integer m.
QUESTION: Do agents $1, \ldots, m$ have a beneficial group misreport under TTC at $\mathcal{E}$?

The natural parameters for GM are the size of the group and the size of the union of the endowments of the group. If we parameterize GM by the latter, then we obtain a **W**[**P**]-hard problem as a corollary of our main result. If the number of manipulating agents is not part

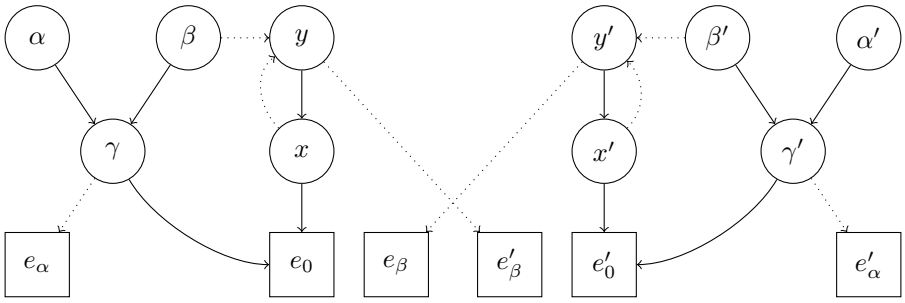

Figure 7: Agent 1 owns $\{e_\alpha, e_0, e_\beta\}$ and agent 2 owns $\{e'_\alpha, e'_0, e'_\beta\}$. Let $R_1 = (\alpha, \beta, e_\alpha, e_\beta, e_0)$, and $R_2 = (\alpha', \beta', e'_\alpha, e'_\beta, e'_0)$. If agent 2 tells the truth, then agent 1 can only get $\{\alpha, e_\alpha, e_\beta\}$ (by telling the truth). If agent 2 lies by reporting $R'_2 = (x', \alpha', \beta')$, then agent 1 can benefit by reporting the lie $R'_1 = (x, \alpha, \beta)$ and receiving $\{x, \alpha, \beta\}$. In this group misreport agents 1 and 2 additionally obtain β and β' (respectively) through the cycle $(e_\beta, \beta, y, e'_\beta, \beta', y', e_\beta)$. Symmetrically, 2 receives $\{x', \alpha', \beta'\}$.

of the parameter, we can choose its size in the definition of the reduction. Thus the group can be of size one, and the exact same construction works.

We now adapt the proof of Theorem 3.1 to show that GM, parameterized by the size of the group k, is $\mathbf{W}[\mathbf{P}]$-hard. In that proof, agent 1 had to obtain k unwanted objects in order to obtain a preferred bundle. Those k choices corresponded to choosing an assignment of weight k to a circuit. We adapt this idea by dividing the k choices among the manipulating agents. The agents each must obtain one unwanted object. Their choices will collectively correspond to an assignment of weight k to a circuit. As in Theorem 3.1, they will benefit if and only if they are able to find a satisfying assignment.

Theorem 5.1. *GM, parameterized by the size of the manipulating group, is* $\mathbf{W}[\mathbf{P}]$*-hard.*

Proof. We reduce from MWCSAT. Let $\mathcal{C}$ be a monotone circuit. Recall the construction of $\mathcal{E}_\mathcal{C}$ from the proof of the main theorem. We adapt this construction to obtain an economy $\mathcal{E}^*_\mathcal{C}$. To construct $\mathcal{E}_\mathcal{C}$, we modified $\mathcal{E}'_\mathcal{C}$ by duplicating the circuit structure k+1 times. We do the same operation but instead we make $2k$ duplications. In order to explain our construction better, we label the copies with the pairs $(0,1), (1,1), \ldots, (0,k), (1,k)$ and refer to the copy with label (b,i) as the (b,i)th copy. In particular, we denote the $2k$ duplications of y by $y^{0,i}, y^{1,i}$ for $1 \leq i \leq k$. We further duplicate the objects $\alpha, \beta, \gamma, e_\alpha, e_\beta$, and label the copies $\alpha^i, \beta^i, \gamma^i, e^i_\alpha, e^i_\beta$ for $i = 1, \ldots, k$. The endowment of manipulating agent i is $\{e_i, e^i_\alpha, e^i_\beta\}$. The preference relations of the manipulating agents and the owners of the duplicated objects over singletons are given below.

$$
\begin{aligned}
R_i &= (\alpha^i, \beta^i, e^i_\alpha, e^i_\beta, \square) \\
R_{\alpha^i} &= (\gamma^i, e^i_\alpha, \square) \\
R_{\beta^i} &= (\gamma^i, \gamma_i, y^{1,i}, y^{2,i}, e^i_\beta, \square)
\end{aligned}
$$

Finally, we change the preferences of the owners of $\gamma^i, y^{0,i}$, and $y^{1,i}$ for each i by replacing β in their preference relation with β^i.

The true allocation of manipulating agent i is $\{\alpha^i, e^i_\alpha, e^i_\beta\}$, and each manipulating agent can only improve upon this allocation by obtaining both α^i and β^i. The proof of the following proposition is analogous to that of Proposition 3.2.

Proposition 5.2. *Let $X = \{x_{i_1}, \ldots, x_{i_k}\}$ be an assignment to $\mathcal{C}$ and let $(R'_j)_{j=1}^k$ be be a group misreport defined by $R'_j = (x^*_{i_j}, \alpha^i, \beta^i)$ for each j. Then, $(R'_j)_{j=1}^k$ is beneficial if and only if X is satisfying.*

It remains to show that if the manipulating agents have a beneficial group manipulation, then there is a satisfying assignment for $\mathcal{C}$. The reasoning follows that of the proof of the main theorem. Let $A = \{A_1, \ldots, A_k\} = \{\{\alpha^1, \beta^1, \delta_1\}, \ldots, \{\alpha^k, \beta^k, \delta_k\}\}$ be the assignments to the manipulating agents. The following observation is completely analogous to Observations 3.5, 3.6, and 3.7.

Observation 5.3. *For $i = 1, \ldots, k$:*

1. $\gamma^i \notin A_i$
2. $\alpha^i \; R'_i \; \beta^i$
3. $\delta_i \; R'_i \; \alpha^i$

As in the proof of the main theorem, the above observation shows that the misreport must take a certain form. For agent i, we have that $R'_i = (\delta_i, \alpha^i, \beta^i)$, where δ_i is part of the circuit structure (again it could be the case that $\delta_i = \gamma_i$ but this will simply result in a lower weight assignment to $\mathcal{C}$).

Let z be the smallest integer such that, for some $b \in \{0, 1\}$, none of the objects $\delta_1, \ldots, \delta_k$ are elements of the (b, z)th copy. The rest of the proof is identical to that of the main theorem. By running TTC we perform a (possibly faulty) evaluation of an assignment to $\mathcal{C}$ of weight k. Since $(y^{b,z}, \beta^z)$ is not a trading cycle, the assignment must be satisfying, as required. □

Acknowledgements

A large part of this research took place while the first author was at the University of Helsinki and the second author was at Aalto University in Finland. The project began with a chance

meeting in the sauna of Töölö Towers; we thank the wonderful staff there. We also thank Jukka Suomela, Patrick Harless, and Vikram Manjunath for many useful discussions.

References

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

[2] A. Abdulkadiroğlu and T. Sönmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003.

[3] K. A. Abrahamson, R. G. Downey, and M. R. Fellows. 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.

[4] E. M. Azevedo and E. Budish. Strategy-proofness in the large. *The Review of Economic Studies*, 86(1):81–116, 2019.

[5] P. Biró, F. Klijn, and S. Pápai. Balanced exchange in a multi-object Shapley-Scarf market. Working Paper, 2019.

[6] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. *Information and Computation*, 251:140–164, 2016.

[7] E. Budish and E. Cantillon. The multi-unit assignment problem: Theory and evidence from course allocation at Harvard. *American Economic Review*, 102(5):2237–71, 2012.

[8] L. Cai, J. Chen, R. Downey, and M. Fellows. On the structure of parameterized problems in NP. *Information and Computation*, 123(1):38–49, 1995.

[9] M. Cesati. The Turing way to parameterized complexity. *Journal of Computer and System Sciences*, 67(4):654–685, 2003.

[10] Y. Chen, J. Flum, and M. Grohe. Machine-based methods in parameterized complexity theory. *Theoretical Computer Science*, 339(2-3):167–199, 2005.

[11] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science*, 141(1):109–131, Apr. 1995.

[12] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[13] U. Dur and T. Morrill. Competitive equilibria in school assignment. *Games and Economic Behavior*, 108:269–274, 2018.

[14] H. Ergin, T. Sönmez, and M. U. Ünver. Dual-donor organ exchange. *Econometrica*, 85(5):1645–1671, 2017.

[15] H. Ergin, T. Sönmez, and M. U. Ünver. Efficient and incentive-compatible liver exchange. *Econometrica*, 88(3):965–1005, 2020.

[16] M. Flammini and H. Gilbert. Parameterized complexity of manipulating sequential allocation. 24th European Conference on Artificial Intelligence (ECAI 2020), 2020.

[17] E. Fujita, J. Lesca, A. Sonoda, T. Todo, and M. Yokoo. A complexity approach for core-selecting exchange with multiple indivisible goods under lexicographic preferences. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 907–913. AAAI Press, 2015.

[18] E. Fujita, J. Lesca, A. Sonoda, T. Todo, and M. Yokoo. A complexity approach for core-selecting exchange under conditionally lexicographic preferences. *Journal of Artificial Intelligence Research*, 63:515–555, 2018.

[19] J. Kennes, D. Monte, and N. Tumennasan. The day care assignment: A dynamic matching problem. *American Economic Journal: Microeconomics*, 6(4):362–406, 2014.

[20] B. Klaus. The coordinate-wise core for multiple-type housing markets is second-best incentive compatible. *Journal of Mathematical Economics*, 44(9-10):919–924, 2008.

[21] H. Konishi, T. Quint, and J. Wako. On the Shapley-Scarf economy: The case of multiple types of indivisible goods. *Journal of Mathematical Economics*, 35(1):1–15, 2001.

[22] Q. Liu and M. Pycia. Ordinal efficiency, fairness, and incentives in large markets. Working Paper, 2016.

[23] J. Ma. Strategy-proofness and the strict core in a market with indivisibilities. *International Journal of Game Theory*, 23(1):75–83, 1994.

[24] V. Majunath and A. Westkamp. Strategy-proof exchange under trichotomous preferences. Working Paper, 2019.

[25] E. Miyagawa. Strategy-proofness and the core in house allocation problems. *Games and Economic Behavior*, 38(2):347–361, 2002.

[26] D. Monte and N. Tumennasan. Centralized allocation in multiple markets. *Journal of Mathematical Economics*, 61:74–85, 2015.

[27] T. Morrill. Making just school assignments. *Games and Economic Behavior*, 92:18–27, 2015.

[28] C. M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[29] S. Pápai. Strategyproof assignment by hierarchical exchange. *Econometrica*, 68(6):1403–1433, 2000.

[30] S. Pápai. Strategyproof exchange of indivisible goods. *Journal of Mathematical Economics*, 39(8):931–959, 2003.

[31] S. Pápai. Exchange in a general market with indivisible goods. *Journal of Economic Theory*, 132(1):208–235, 2007.

[32] M. Pycia and M. U. Ünver. Incentive compatible allocation and exchange of discrete resources. *Theoretical Economics*, 12(1):287–329, 2017.

[33] A. E. Roth and A. Postlewaite. Weak versus strong domination in a market with indivisible goods. *Journal of Mathematical Economics*, 4(2):131–137, 1977.

[34] A. E. Roth, T. Sönmez, and M. U. Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 2004.

[35] J. Schummer and R. V. Vohra. Assignment of arrival slots. *American Economic Journal: Microeconomics*, 5(2):164–85, 2013.

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

[37] S. Sikdar, S. Adali, and L. Xia. Mechanism design for multi-type housing markets. In *AAAI*, pages 684–690, 2017.

[38] T. Sönmez. Strategy-proofness and essentially single-valued cores. *Econometrica*, 67(3):677–689, 1999.

[39] T. Sönmez and M. U. Ünver. House allocation with existing tenants: An equivalence. *Games and Economic Behavior*, 52(1):153–185, 2005.

[40] T. Sönmez and M. U. Ünver. House allocation with existing tenants: A characterization. *Games and Economic Behavior*, 69(2):425–445, 2010.

[41] A. Sonoda, E. Fujita, T. Todo, and M. Yokoo. Two case studies for trading multiple indivisible goods with indifferences. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[42] Z. Sun, H. Hata, T. Todo, and M. Yokoo. Exchange of indivisible objects with asymmetry. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[43] L.-G. Svensson. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4):557–567, 1999.

[44] T. Todo, H. Sun, and M. Yokoo. Strategyproof exchange with multiple private endowments. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 805–811, 2014.

[45] J. Wang, W. Su, M. Yang, J. Guo, Q. Feng, F. Shi, and J. Chen. Parameterized complexity of control and bribery for d-approval elections. *Theoretical Computer Science*, 595:82–91, 2015.

A Additional Examples

In this section we provide several fully worked examples of the constructions defined in the main text. Previously, in Figures 5 and 6, we considered a simple circuit with no internal gates, and depicted the corresponding economies $\mathcal{E}'_{\mathcal{C}}$ and $\mathcal{E}_{\mathcal{C}}$. In Figures 8 and 9 we consider a circuit with one internal gate (Figure 3), and provide the full constructions of the corresponding $\mathcal{E}'_{\mathcal{C}}$ and $\mathcal{E}_{\mathcal{C}}$. In Figures 10 and 11 we detail several steps of TTC for two different circuits and their resulting economy $\mathcal{E}'_{\mathcal{C}}$ constructions.

A.1 A Circuit with One Internal Gate

Consider the circuit from Figure 3 equivalent to the formula $x_1 \wedge (x_1 \vee x_2)$. It is degenerate in a certain sense: it is equivalent to the circuit with a single gate x_1 serving as input and output. Thus there is only one satisfying assignment of weight 1, in which $x_1 = 1$ and $x_2 = 0$. In Figure 8, we depict the corresponding economy $\mathcal{E}'_{\mathcal{C}}$ following the definitions in Section 3.1.

By design, there is a beneficial misreport of a particular form corresponding to the satisfying assignment of weight 1. Consider the possible preference relations that agent 1 can report. Recall that their true preference is $R_1 = (\alpha, \beta, e_\alpha, e_\beta, e_k, e_{k-1}, \ldots, e_1)$. If agent 1 reports the truth, then they receive $\{\alpha, e_\alpha, e_\beta\}$. If agent 1 reports $R'_1 = (x_1^2, \alpha, \beta, \square)$ (which represents the satisfying assignment $x_1 = 1$), then they receive $\{x_1^2, \alpha, \beta\}$ and are better off. If agent 1 reports $R''_1 = (x_2^1, \alpha, \beta, \square)$ (which represents a non-satisfying weight 1 assignment $x_2 = 1$), then they receive $\{x_2^1, \alpha, e_\beta\}$ and are worse off.

Next, we show the result of the duplication process to construct $\mathcal{E}_{\mathcal{C}}$ for $k = 1$ in Figure 9. As in Figure 6, we omit the objects α, β, γ , e_α, and e_β in this diagram for readability.

A.2 Steps of TTC for Economy from Figure 5

The economy from Figure 5 is the construction $\mathcal{E}'_{\mathcal{C}}$ where circuit $\mathcal{C}$ is equivalent to $x_1 \wedge x_2$ and $k = 1$. Recall that the true preference of agent 1 is $R_1 = (\alpha, \beta, e_\alpha, e_\beta, e_1)$, and that they are allocated $\{\alpha, e_\alpha, e_\beta\}$.

In Figure 10, we show the result of applying TTC when agent 1 misreports to $R'_1 = (x_1^*, \alpha, \beta, \square)$. For readability, we omit drawing agent 1's preference except where necessary. In the top left, we depict Step 1 of TTC. Since agent 1 topranks $x_1^* = x_1^1$, we have the cycle (x_1^*, γ_1, e_1). All objects that agent 1 owns point to x_1^* as well, but we omit these directed edges. We resolve this cycle. In the top right of the figure, we depict Step 2. We remove objects that were in a cycle in Step 1 (now greyed) and update all agents' preferences over remaining objects. For example, since e_1 is gone, the owner of γ now topranks e_α. Agent 1 owns e_α which now topranks and points to α. Two cycles appear: (x_2^1, h_3^2) and $(\alpha, \gamma, e_\alpha)$. We select the former to resolve. In Step 3 (bottom left), we update preferences, and since h_1^3 only ranked one object in the previous step, they now toprank their own object. Again there are two cycles: (h_1^3) and $(\alpha, \gamma, e_\alpha)$. We select the former to resolve. In Step 4 (bottom

right), $(\alpha, \gamma, e_\alpha)$ is the only cycle. In Step 5 (not shown), (y, β) form a cycle. Thus, we have that agent 1 is allocated $\{x_1^*, \alpha, e_\beta\}$. This outcome is worse than the outcome if they had told the truth.

Note that in Step 2, removing h_3^2 from the graph corresponds to the second predecessor of the $\wedge$-gate g_3 taking value 0. Thus, the output $\wedge$-gate g_3 takes value 0. The result, by design, is that y cycles with β, taking away the opportunity for agent 1 to receive β.

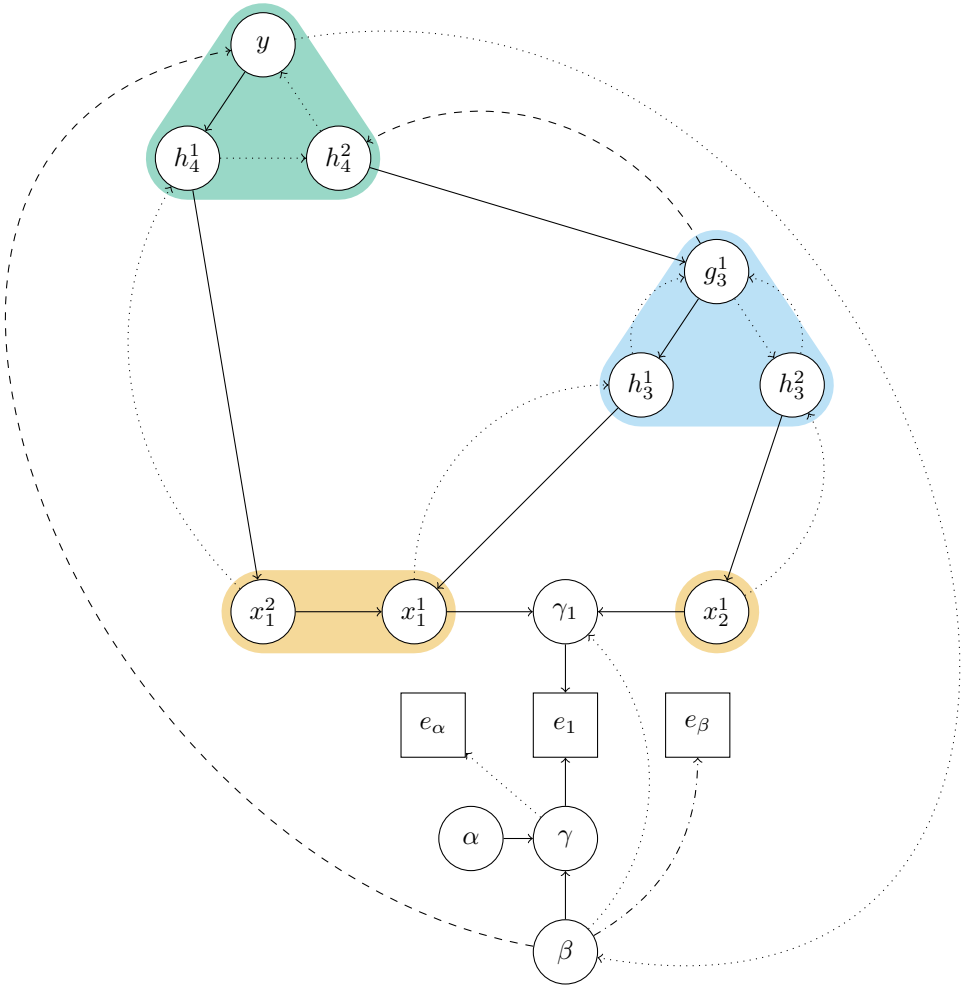

Figure 8: The economy $\mathcal{E}'_{\mathcal{C}}$ for a circuit with one internal gate.

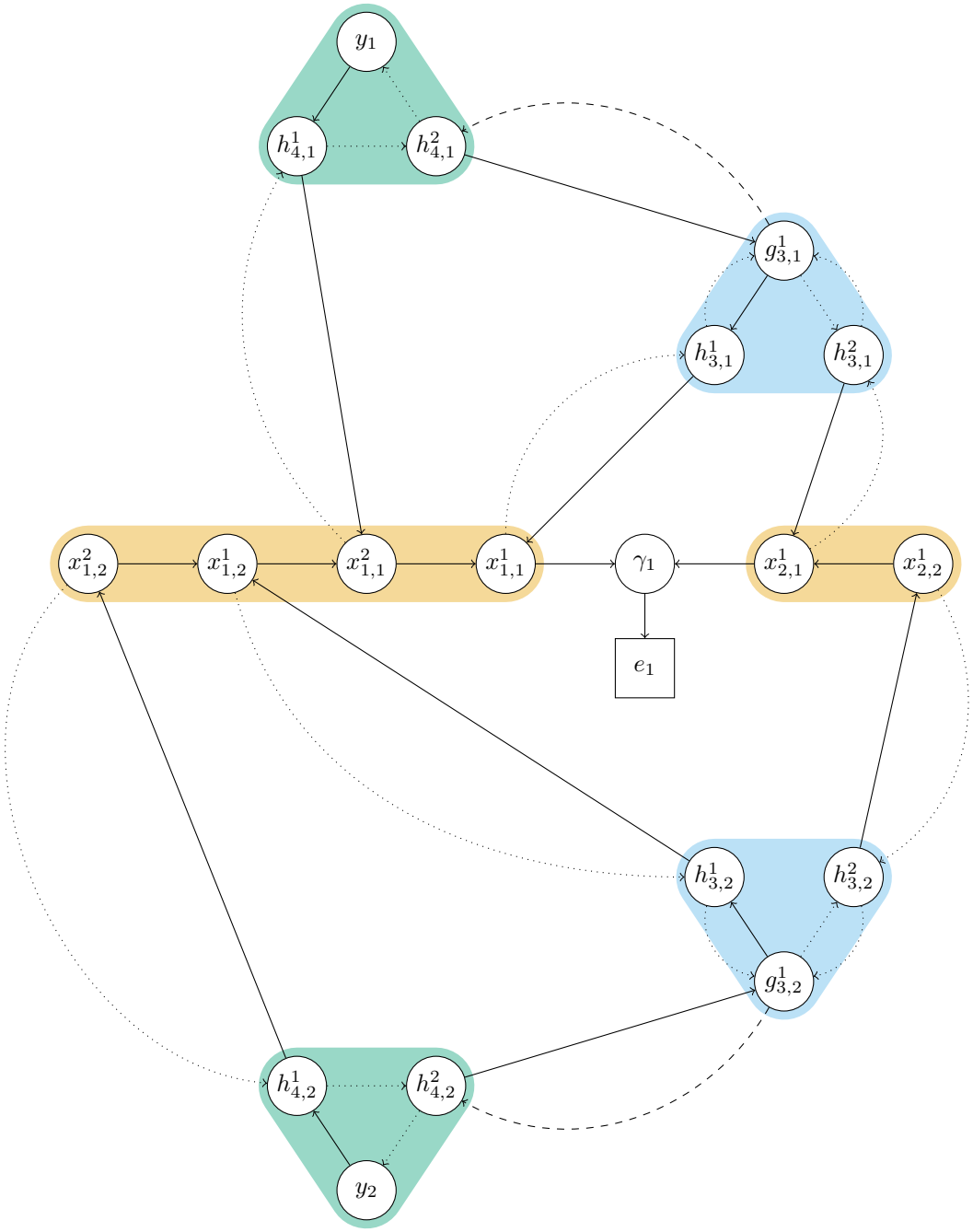

Figure 9: The economy $\mathcal{E}_{\mathcal{C}}$ "duplicates" certain elements of $\mathcal{E}'_{\mathcal{C}}$.

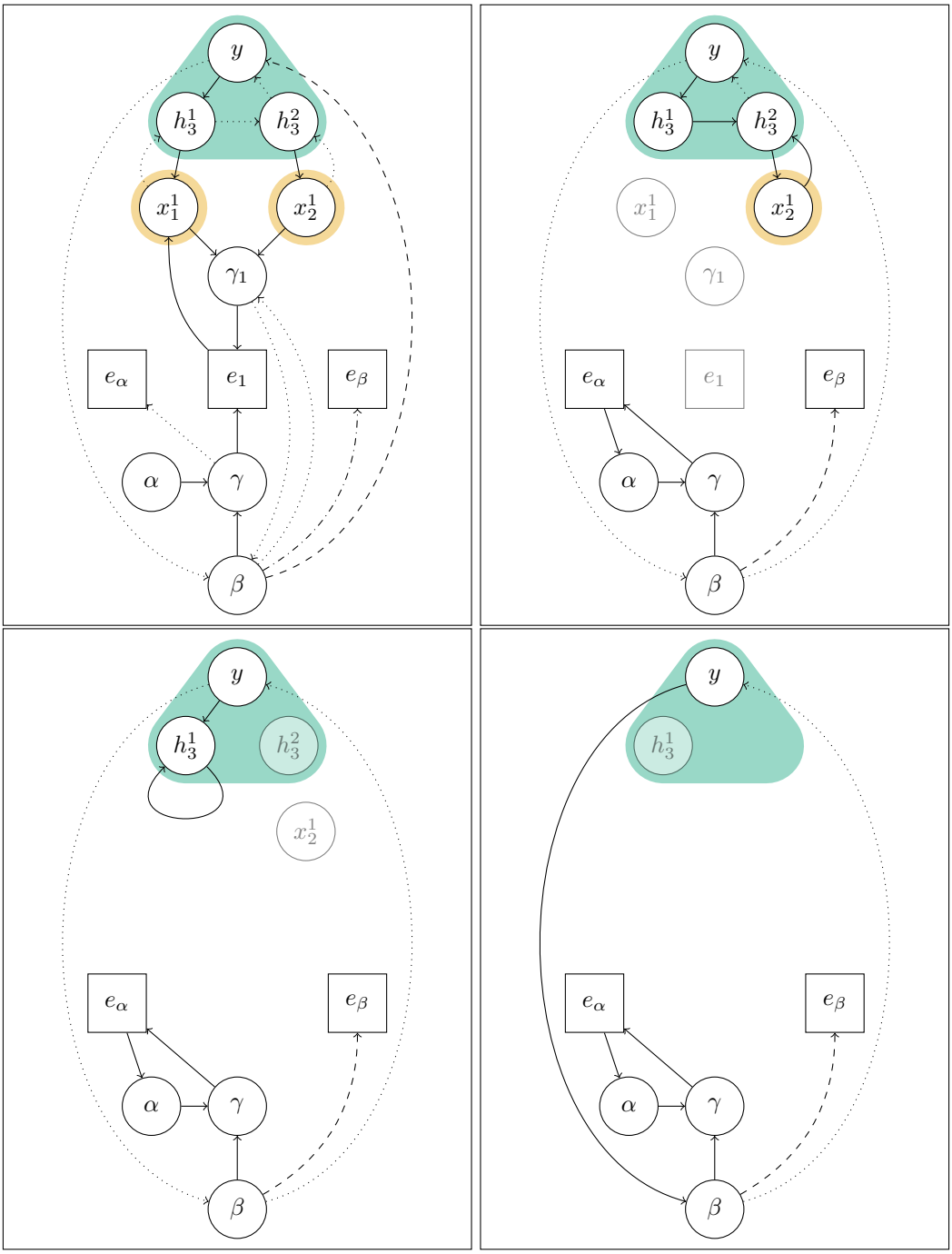

Figure 10: Four steps of TTC for an economy $\mathcal{E}'_{\mathcal{C}}$.

A.3 Steps of TTC for Economy Another Circuit

In Figure 11, we show the construction of $\mathcal{E}'_{\mathcal{C}}$ where $\mathcal{C}$ is the circuit equivalent to $x_1 \vee x_2$ and $k = 1$. The true preference of agent 1 is $R_1 = (\alpha, \beta, e_\alpha, e_\beta, e_1)$, and they are allocated $\{\alpha, e_\alpha, e_\beta\}$. We show the result of applying TTC when agent 1 misreports to $R'_1 = (x^*_1, \alpha, \beta, \square)$. Note that this is a satisfying assignment of weight 1 for the circuit. By design, the misreport $R'_1 = (x^*_1, \alpha, \beta, \square)$ will be beneficial.

In the top left, we depict Step 1 of TTC. Since agent 1 topranks $x^*_1 = x^1_1$, we have the cycle (x^*_1, γ_1, e_1). We resolve this cycle. All objects that agent 1 owns point to x^*_1 as well, but again we omit these directed edges. In Step 2 (top right), given updated preferences, we have three cycles: (h^2_3, x^1_2), (y, h^1_3), and $(\alpha, \gamma, e_\alpha)$. We select the first cycle to resolve. In Step 3 (bottom left), cycles are (y, h^1_3), and $(\alpha, \gamma, e_\alpha)$. We select the first cycle to resolve. In Step 4 (bottom right), $(\alpha, \gamma, e_\alpha)$ is the only cycle. In Step 5 (not shown), (β, e_β) form a cycle. Thus, we have that agent 1 is allocated $\{x^*_1, \alpha, \beta\}$. This outcome is preferred to the one where they told the truth.

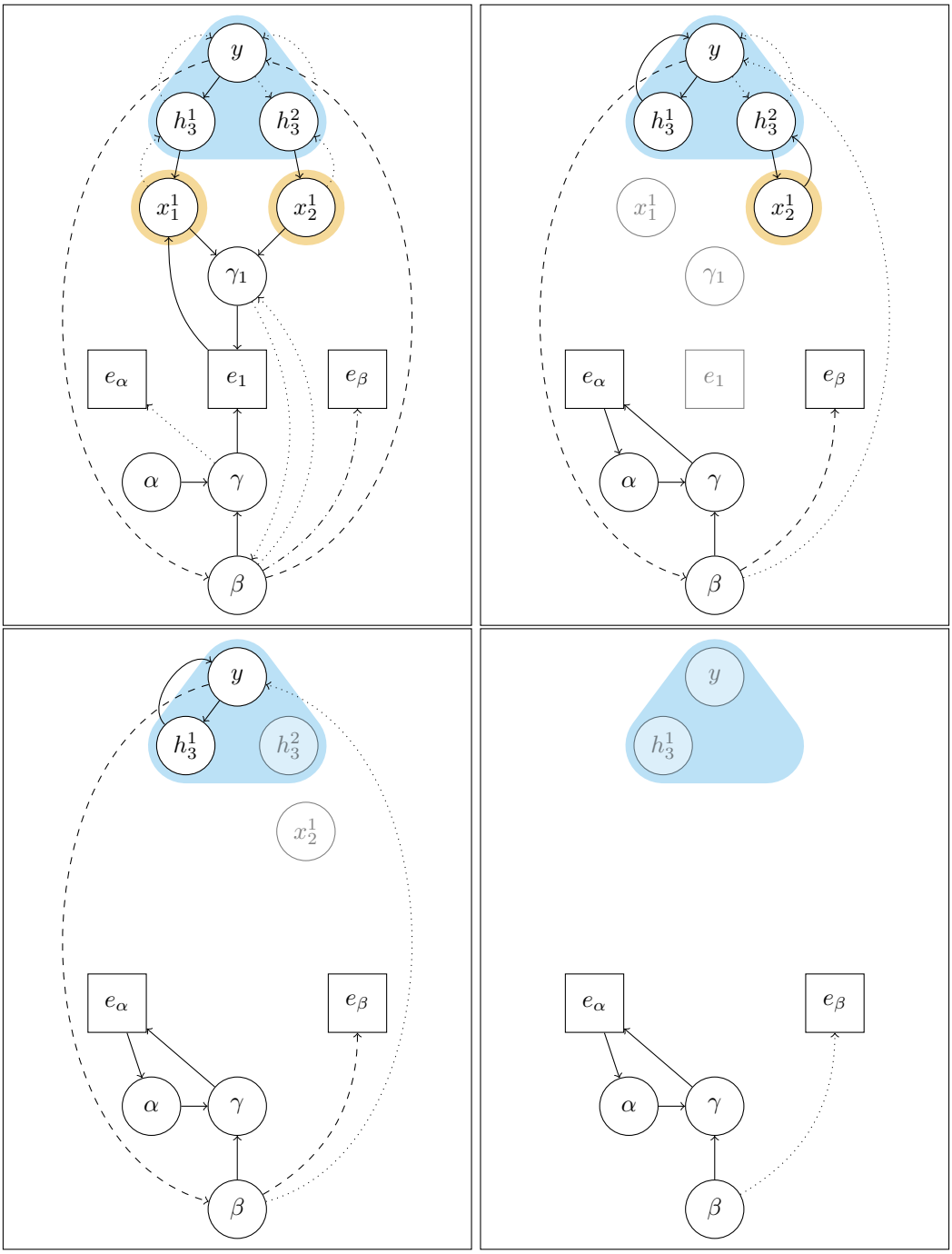

Figure 11: Four steps of TTC for an economy $\mathcal{E}'_{\mathcal{C}}$ with an $\vee$ output gate.