Abstract

Mixed-multi unit combinatorial auctions (MMUCAs) are extensions of classical combinatorial auctions (CAs) where bidders trade transformations of goods rather than just sets of goods. Solving MMUCAs, i.e., determining the sequences of bids to be accepted by the auctioneer, is computationally intractable in general. However, differently from CAs, little was known about whether polynomial-time solvable classes of MMUCAs can be singled out based on constraining their characteristics.

The paper precisely fills this gap, by depicting a clear picture of the “tractability frontier” for MMUCAs instances under both structural and qualitative restrictions, which characterize interactions among bidders and types of bids involved in the various transformations, respectively. By analyzing these restrictions, a sharp frontier is charted based on various dichotomy results. In particular, tractability islands resulting from the investigation generalize on MMUCAs the largest class of tractable CAs emerging from the literature.

1 Introduction

Mixed-multi unit combinatorial auctions (MMUCAs) are extensions of classical combinatorial auctions (CAs) where bidders trade transformations of goods rather than just simple goods [Cerquides et al., 2007]. These mechanisms are particularly useful in the context of automatizing supply chain formation, where production processes often emerge as the result of complex interactions among producers and consumers (cf. [Walsh and Wellman, 2003]).

Formally, a transformation over a set $G$ of types of goods is a tuple $⟨I, O, p⟩$ where $I ∈ \mathbb{N}^{|G|}$ (resp., $O ∈ \mathbb{N}^{|G|}$) is a vector of natural numbers denoting the quantities of the goods that are required (resp., produced) for the transformation to take place (resp., as a result of the transformation), and $p ∈ \mathbb{R}$ is the payment the bidder is willing to make in return for being allocated the transformation. Then, a mixed multi-unit combinatorial auction instance is a tuple $⟨G, T, \mathcal{U}_{in}, \mathcal{U}_{out}⟩$ where $T$ is a bag of transformations over $G$, and $\mathcal{U}_{in} ∈ \mathbb{N}^{|G|}$ (resp., $\mathcal{U}_{out} ∈ \mathbb{N}^{|G|}$) is a vector denoting the quantities of goods the auctioneer holds to begin with (resp., expects to end up with).

1If $p < 0$, then the auctioneer must actually pay $-p$ to the bidder in order for she to implement the transformation.

Solving a MMUCA instance $⟨G, T, \mathcal{U}_{in}, \mathcal{U}_{out}⟩$ amounts to find a sequence of transformations such that, based on the input goods in $\mathcal{U}_{in}$, the auctioneer may end up with the desired goods in $\mathcal{U}_{out}$ with the maximum possible revenue (short: WINNER-DETERMINATION problem). This problem has intensively been studied in recent years, by extending to MMUCAs several results originally conceived for classical CAs. In particular, languages have been defined and analyzed that allow bidders to compose (atomic) bids in a natural and intuitive way [Cerquides et al., 2007]; and, motivated by their intractability (formally, NP-hardness), solution approaches have been proposed (see, e.g., [Giovannucci et al., 2007]) that well-behave on realistic scenarios [Ottens and Endriss, 2008].

Differently from classical CAs, however, little was known about whether polynomial-time solvable classes of MMUCAs can be singled out based on the structural and topological properties of the instances at hand. As a matter of fact, by focusing on the kinds of interactions among bidders that are likely to occur in practice, classes of instances over which WINNER-DETERMINATION is tractable—called “islands of tractability” in the literature—have been identified for classical CAs (such as structured item graphs [Conitzer et al., 2004] or bounded hypertree-width dual hypergraphs [Gottlob and Greco, 2007]). However, none of these results had a counterpart in the case of MMUCAs.

The aim of this paper is precisely to fill this gap, by depicting a clear and complete picture of the frontier of tractability for MMUCAs instances. In particular, since the existence of a solution is not guaranteed in the case of MMUCAs (unlike classical CAs), attention is focused not only on the WINNER-DETERMINATION but also on the FEASIBILITY problem of deciding whether a given instance admits a solution at all; indeed, an important and peculiar source of complexity for MMUCAs lays hidden in this latter problem.

In more detail, in the first part of the paper, we show two interesting dichotomy results pertaining FEASIBILITY, which precisely determine the frontier of tractability under qualitative restrictions, i.e., under restrictions characterizing the types of bids in terms of the variety and quantity of goods involved in the various transformations. In fact:

(1) In the case where bidders submit sets of transformations and accept any combination of them for the sum of their prizes (short: OR-language), we show that FEASIBILITY is tractable if and only if every transformation requires and produces one item of one single good at most, or every type of good is required (or produced) as input (resp., output) by one transformation at most.
In the case where each bidder accepts at most one transformation from the set of hers submitted transformations (short: XOR-language), we show that \textsc{Feasibility} is tractable \textit{if and only if} every type of good is required as input by one transformation at most.

Then, we turn to consider \textit{structural} properties of the networks originating from bidder interactions, motivated by the fact that many NP-hard problems in different application areas are known to be efficiently solvable when restricted to instances that can be modeled via (nearly)acyclic graphs. Surprisingly, bad news emerged from our investigation. Indeed:

We show that \textsc{Feasibility} is hard on (nearly)acyclic instances too. In particular, this is the case for two natural ways of encoding bidder interactions, namely for \textit{transformations graphs} (where nodes are in one-to-one correspondence with transformations and an edge indicates that one transformation produces a good required by the other), and for \textit{goods graphs} (where nodes are in one-to-one correspondence with goods and an edge indicates the possibility of transforming a good into another).

Eventually, in the final part of the paper, we focus on the \textsc{Winner-Determination} problem, in order to single out tractable classes of MMUCAs complementing those defined in [Conitzer et al., 2004; Gottlob and Greco, 2007] for CAS:

\textbf{(4)} On the one hand, \textsc{Winner-Determination} emerges to be intractable under most kinds of qualitative restrictions, for it inherits all the intractability results that hold with \textsc{Feasibility} as well as with classical CAS.

\textbf{(5)} However, on the other hand, we show that if qualitative restrictions are combined with suitable structural restrictions (on a hypergraph encoding the interactions), then a tractable class of instances can be identified, which truly generalizes on MMUCAs the largest class of tractable CAS singled out in the literature [Gottlob and Greco, 2007]. In particular, for this class, a polynomial-time solution algorithm is proposed and its properties are analyzed.

The rest of the paper is organized as follows. Section 2 reports a few preliminaries on MMUCAs. The complexity of \textsc{Feasibility} under qualitative and structural restrictions is discussed in Section 3 and Section 4, respectively. Tractability islands for the \textsc{Winner-Determination} problem are isolated in Section 5, and conclusions are drawn in Section 6.

---

### 2 Mixed Multi-Unit Combinatorial Auctions

Let $G$ be a set of types of goods. For each vector $W \in \mathbb{N}^{|G|}$, we denote by $W(g)$ the element of $W$ referring to any good $g \in G$. In some cases, $W$ will be viewed as a bag over $G$.

\textbf{Atomic Bids.} Let $A = \langle G, T, U_{in}, U_{out} \rangle$ be a MMUCA instance. Solving $A$ amounts to decide which transformations have to be accepted, and in which order to implement them.

Formally, consider a sequence of transformations $\sigma = (I_1, O_1, p_1), \ldots, (I_k, O_k, p_k)$ such that $(I_i, O_i, p_i) \in T$, for each $i \in \{1, \ldots, k\}$. Let $M_0 = U_{in}$ denote the quantities of goods initially hold by the auctioneer, and let $M_i \in \mathbb{N}^{|G|}$ be the quantities owned after the $i$-th transformation, i.e., $\forall g \in G, M_i(g) = M_{i-1}(g) + O_i(g) - I_i(g)$.

Then, $\sigma$ is legal w.r.t. $A$ if $M_{i-1}(g) \geq I_i(g)$, for each $i \in \{1, \ldots, k\}$ and $g \in G$. The revenue of the auctioneer with $\sigma$ is the sum of the payments associated with each transformation in it, i.e., the value $\sum_{i=1}^{k} p_i$. Under the free disposal assumption, a legal sequence of transformations $\sigma$ is a solution to $A$ if $\forall g \in G, M_{k}(g) \geq U_{out}(g)$. An optimal solution is a solution providing the auctioneer with the maximum revenue over all the possible solutions.

Throughout the paper we shall always look for (optimal) solutions under the free disposal assumption.

\textbf{Bidding Languages.} In the basic setting above, there is a one-to-one correspondence between bidders and transformations. However, in many practical cases, bidders may want to exploit more expressive languages to submit bids, in place of submitting atomic transformations only. Thus, as commonly done in the literature, we assume that bidders may submit sets of transformations under conditions of two kinds.

An OR-condition on a bag $S \subseteq T$ of transformations states that the bidder may accept any sub-bag of $S$ at the sum of the respective prizes; instead, a XOR-condition states that she is prepared to accept at most one of them. Equipping $A$ with a set $E$ of c-conditions (with $c \in \{\text{or}, \text{xor}\}$, $\bigcup_{S \in E} S = T$, and $S \cap S' = \emptyset, \forall S, S' \in E$) is denoted by $A|_{c,E}$.

### 3 Feasibility and Qualitative Restrictions

In this section, we start the analysis of the complexity of \textsc{Feasibility}, by taking into account various qualitative properties of the underlying instances, as for they can formally be measured in terms of the following parameters:

- \textit{in-var}(A) = $\max_{(I, O, p) \in T} \{|I(g)| \mid g \in G, I(g) > 0\}$ is the input variety of $A$, i.e., the maximum number of types of goods required as input over all transformations. Symmetrically, the output variety is the value \textit{out-var}(A) = $\max_{(I, O, p) \in T} \{|O(g)| \mid g \in G, O(g) > 0\}$.

- \textit{in-mul}(A) = $\max_{(I, O, p) \in T, g \in G} I(g)$ is the input multiplicity of $A$, i.e., the maximum quantity of any good required as input over all transformations. And, symmetrically, the output multiplicity is the value \textit{out-mul}(A) = $\max_{(I, O, p) \in T, g \in G} O(g)$.

- \textit{in-deg}(A) = $\max_{g \in G} \{|I(g, p) \in T \mid O(g) > 0\}$ is the input degree of $A$, i.e., the maximum number of transformations producing a given good over all goods. Symmetrically, the output degree of $A$ is the value \textit{out-deg}(A) = $\max_{g \in G} \{|I(g, p) \in T \mid I(g) > 0\}$.

Below, $\mathcal{C}(\text{i}v, \text{o}v, \text{i}m, \text{o}m, \text{i}d, \text{o}d)$ will denote the class of instances $A$ such that: \textit{in-var}(A) $\leq \text{i}v$, \textit{out-var}(A) $\leq \text{o}v$, \textit{in-mul}(A) $\leq \text{i}m$, \textit{out-mul}(A) $\leq \text{o}m$, \textit{in-deg}(A) $\leq \text{i}d$, and \textit{out-deg}(A) $\leq \text{o}d$. Also, the symbol $\infty$ is used to denote that no bound is issued on some given parameter.

\textbf{Results.} A summary of the complexity figures emerging from our analysis is reported in Figure 1, for both the case of OR and XOR conditions. Note that these results precisely depict the tractability frontier, since relaxing any condition in a tractable scenario immediately leads to intractability. Interestingly, the expressiveness of the XOR-language [Cerquides et al., 2007] is payed in terms of smaller tractability islands.
**3.1 Tractable Instances (OR-conditions)**

We next illustrate the good news on the tractability of **Feasibility**. We start by considering the case where every transformation requires and produces an item of one good at most.

**Theorem 3.1** Let \( \mathcal{A} = (G, T, U, \mathcal{Un}) \) be a MMUCA such that \( \mathcal{A} \in \mathcal{C}(1, 1, 1, \infty, \infty, \infty) \). Then, **Feasibility can be solved in time** \( O(|T|^2 \times |G| \times \log \Delta) \).

**Proof. (Sketch).** Based on \( \mathcal{A} \), consider the directed graph \((N, E)\) built as follows. The set \( N \) of the nodes contains a node \( g \) for each good \( g \in G \), plus the distinguished node \( n \). There is an edge \((g, g')\) from \( g \) to \( g' \) in \( E \) if there is a transformation \((T_h, O_h, p_h) \in T\) such that \( I_h(g) > 0 \) and \( O_h(g') > 0 \); and, there is an edge \((g, n)\) in \( E \) for each \( g \in G \).

The idea is now to consider \((N, E)\) as a demand flow network (see, e.g., [Jensen and Bard, 2002]) where:

(i) each edge \((g, g')\) has capacity \( u(g, g') = 1 \), and each edge \((g, n)\) has capacity \( u(g, n) = \mathcal{Un}(g)\); and,

(ii) each node \( g \) has demand \( d(g) = \mathcal{Un}(g) - \mathcal{In}(g) \), while \( n \) has demand \( d(n) = \sum_{g \in G} \mathcal{Un}(g) - \sum_{g \in G} \mathcal{In}(g) \).

It can be shown that \( \mathcal{A} \) has a solution if and only if there is a circulation on \((N, E)\), that is a function \( f : E \to \mathbb{R}^+ \) such that \( f(e) \leq u(e) \) for each \( e \in E \), and \( \sum_{v \in N} f(v, t) = d(v) \) for each \( v \in N \).

Eventually, the existence of a circulation can be checked in \( O(|G|^3 \times \log(|G| \times \max_g \mathcal{In}(g))) \), where \( (|G| + 1) \) is the number of nodes in \( N \). In addition, note that OR-conditions do not play any role as far as Feasibility is concerned. \( \square \)

Let us now turn to the scenario where every type of good is produced by one transformation at most, i.e., let us assume that input degrees are unitary at most. Also, let \( \Delta = \max_g \mathcal{Un}(g) + \sum_{(I, O, p) \in T} \mathcal{O}(g) \) be an upper bound on the number of any good produced in some solution.

**Theorem 3.2** Let \( \mathcal{A} = (G, T, U, \mathcal{Un}, \mathcal{Out}) \) be a MMUCA such that \( \mathcal{A} \in \mathcal{C}(\infty, \infty, \infty, \infty, 1, \infty) \). Then, **Feasibility can be solved in time** \( O(|T|^2 \times |G| \times \log \Delta) \).

**Proof. (Sketch).** Define a transformation \((T, O, p)\) as necessary w.r.t. a good \( g \) if (a) \( O(g) > 0 \); or, (b) there is a transformation \((T', O', p')\) such that \( O'(g) > 0 \), and \((T, O, p)\) is necessary w.r.t. some good \( g' \) such that \( T'(g') > 0 \). Let \( T_r \subseteq T \) denote the set of all the necessary transformations w.r.t. some good \( g \) with \( \mathcal{Out}(g) > 0 \). Since \( \text{in-deg}(A) \leq 1 \), \( \mathcal{A} \) admits a solution if and only if it is possible to execute all the transformations in \( T_r \). Thus, we may simple start applying transformations in \( T_r \) till a step \( k \) is reached such all the transformations in \( T_r \) are applied (thereby witnessing that there is a solution to \( A \)), or there is some transformation in \( T_r \) that cannot be applied (so that there is no solution to \( A \)).

Eventually, \([T]\) steps at most have to be performed, each one being feasible in \( O(|T| \times |G| \times \log \Delta) \). \( \square \)

A complementary reasoning (i.e., applying transformations from input goods till there is an applicable transformation) shows the tractability of unitary output degree instances.

**Theorem 3.3** Let \( \mathcal{A} = (G, T, U, \mathcal{Out}) \) be a MMUCA such that \( \mathcal{A} \in \mathcal{C}(\infty, \infty, \infty, \infty, \infty, 1) \). Then, **Feasibility can be solved in time** \( O(|T|^2 \times |G| \times \log \Delta) \).

**3.2 Hard Instances (OR-conditions)**

Hardness results are next provided as reductions from the Satisfiability of Boolean formulas in conjunctive normal form. In particular, recall that deciding whether a Boolean formula in conjunctive normal form \( \Phi = c_1 \land \ldots \land c_m \) over the variables \( X_1, \ldots, X_n \) is satisfiable is \( \text{NP-hard} \).

In fact, to our ends, we find useful to state the intractability of a specific class of Boolean formulas.

**Lemma 3.4** Satisfiability is \( \text{NP-hard} \), even if each variable occurs positively in two clauses and negatively in another, and if each clause contains three variables at most.

**Theorem 3.5** Feasibility is \( \text{NP-complete} \), even under atomic bids and restricted on the class \( \mathcal{C}(1, 2, 1, 2, 1, 2) \).

**Proof. (Sketch).** Membership in \( \text{NP} \) was show in [Cerquides et al., 2007] for the whole class \( \mathcal{C}(\infty, \infty, \infty, \infty, \infty, \infty) \). As for the hardness, let \( \Phi \) be a formula satisfying the conditions in Lemma 3.4, and \( \mathcal{A}(\Phi) = (G, T, U, \mathcal{Out}) \) be such that \( G = \bigcup_{i=1}^{m} (X_i, T_i, X_i^c) \cup \{c_1, \ldots, c_m\} \cup \{c_j \mid X_j \text{ occurs in } c_j\} \cup \mathcal{Un} \cup \mathcal{Out} = \{c_1, \ldots, c_m\} \), and \( T = \bigcup_{i=1}^{m} T_i \).

In particular, for each variable \( X_i \) occurring positively in the clauses \( c_i \) and \( c_j \) while occurring negatively in \( c_i \), the set \( T_i \) consists of the transformations graphically depicted in Figure 2. Observe that two transformations occur in \( T_i \) requiring \( X_i \) as input, one that produces \( X_i^c \) and another that produces \( X_i^t \). These transformations are meant to encode the selection of a truth value assignment to the variable \( X_i \), and, in fact, they are mutually exclusive in any solution, since there is just one copy of \( X_i \) in \( \mathcal{A}(\Phi) \). Eventually, one may check that it is possible to produce the output goods \( \{c_1, \ldots, c_m\} \) if and only if all the various selections encode a satisfying assignment to \( \Phi \). Thus, \( \Phi \) is satisfiable \( \Leftrightarrow \mathcal{A}(\Phi) \) has a solution.

---

\( ^{2}\)Given our interest in the **Feasibility** problem, we shall omit the indication of the payments in the various transformations.
For each variable $X_i$ that occurs positively in $c_\alpha$ and $c_\beta$ while occurring negatively in $c_\gamma$, $T_i$ consists of $\langle \{X_1, \{X_1^\prime, X_1^\prime\}\}, \{X_1^\prime, X_1^\prime\}\rangle$, $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, and $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$. In addition, $\mathcal{L}$ contains the two XOR-conditions on $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, and $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, respectively—see Section 1. In addition, we shall consider their undirected versions, denoted by $\overline{\mathcal{T}}(G)$ and $\overline{\mathcal{G}}(A)$, respectively.

Our first result is very bad news on transformations graphs. Indeed, the undirected transformations graph associated with the MMUC instance built in the proof of Theorem 3.5 is acyclic, while still encoding an NP-complete problem. Thus:

**Corollary 4.1** Feasibility is NP-complete, even restricted on the class $\{A \mid \overline{\mathcal{T}}(A)$ is acyclic$\}$ (and, hence, on the class $\{A \mid T(G(A)$ is acyclic$\}$) and under atomic bids.

Similarly, the (directed) graphs graph associated with the instance in the proof of Theorem 3.5 is also acyclic. Thus:

**Corollary 4.2** Feasibility is NP-complete, even restricted on the class $\{A \mid \mathcal{T}(A)$ is acyclic$\}$ and under atomic bids.

In particular, the above result is rather interesting in the light that instances with (directed) acyclic goods graphs correspond to natural transformation processes (cf. [Oetens and Endriss, 2008]). Thus, transformation processes emerge to be as hard as arbitrary trades and exchanges of goods.

In order to conclude our analysis, we shall show that Feasibility remains NP-hard on nearly-acyclic undirected goods graphs. To this end, we need to recall the notion of tree decomposition, which is the most powerful generalization of graph acyclicity. Formally, a tree decomposition of a graph $G = (V,E)$ is a pair $(T, \chi)$, where $T = (N,F)$ is a tree, and $\chi$ is a labeling function assigning to each vertex $p \in N$ a set of vertices $\chi(p) \subseteq V$, such that the following conditions are satisfied: (1) for each node $b$ of $G$, there exists $p \in N$ such that $b \in \chi(p)$; (2) for each edge $(b,d) \in E$, there exists $p \in N$ such that $\{b,d\} \subseteq \chi(p)$; and, (3) for each node $b$ of $G$, the set $\{p \in N \mid b \in \chi(p)\}$ induces a connected subtree.

### 4 Structural Restrictions (Alone) Do Not Help

Many NP-hard problems in different application areas, ranging, e.g., from Constraint Satisfaction to Database Theory, are known to be efficiently solvable when restricted to instances that can be modeled via (nearly)acyclic graphs. Therefore, one may naturally expect that these structural restrictions are also beneficial to isolate tractable MMUCAs. This is next investigated, by modeling interactions among bidders in an instance $A$ via the transformations and the goods graph of $A$, denoted by $\mathcal{T}(G(A)$ and $\mathcal{G}(A)$, respectively—see Section 1. In addition, we shall consider their undirected versions, denoted by $\overline{\mathcal{T}}(G(A)$ and $\overline{\mathcal{G}}(A)$, respectively.

For each variable $X_i$ that occurs positively in $c_\alpha$ and $c_\beta$ while occurring negatively in $c_\gamma$, $T_i$ consists of $\langle \{X_1, \{X_1^\prime, X_1^\prime\}\}, \{X_1^\prime, X_1^\prime\}\rangle$, $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, and $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$. In addition, $\mathcal{L}$ contains the two XOR-conditions on $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, and $\langle \{X_1^\prime, X_1^\prime\}, \{X_1^\prime, X_1^\prime\}\rangle$, respectively—see Section 1. In addition, we shall consider their undirected versions, denoted by $\overline{\mathcal{T}}(G(A)$ and $\overline{\mathcal{G}}(A)$, respectively.

Our first result is very bad news on transformations graphs. Indeed, the undirected transformations graph associated with the MMUC instance built in the proof of Theorem 3.5 is acyclic, while still encoding an NP-complete problem. Thus:

**Corollary 4.1** Feasibility is NP-complete, even restricted on the class $\{A \mid \overline{\mathcal{T}}(A)$ is acyclic$\}$ (and, hence, on the class $\{A \mid \mathcal{T}(G(A)$ is acyclic$\}$) and under atomic bids.

Similarly, the (directed) graphs graph associated with the instance in the proof of Theorem 3.5 is also acyclic. Thus:

**Corollary 4.2** Feasibility is NP-complete, even restricted on the class $\{A \mid \mathcal{T}(G(A)$ is acyclic$\}$ and under atomic bids.

In particular, the above result is rather interesting in the light that instances with (directed) acyclic goods graphs correspond to natural transformation processes (cf. [Oetens and Endriss, 2008]). Thus, transformation processes emerge to be as hard as arbitrary trades and exchanges of goods.

In order to conclude our analysis, we shall show that Feasibility remains NP-hard on nearly-acyclic undirected goods graphs. To this end, we need to recall the notion of tree decomposition, which is the most powerful generalization of graph acyclicity. Formally, a tree decomposition of a graph $G = (V,E)$ is a pair $(T, \chi)$, where $T = (N,F)$ is a tree, and $\chi$ is a labeling function assigning to each vertex $p \in N$ a set of vertices $\chi(p) \subseteq V$, such that the following conditions are satisfied: (1) for each node $b$ of $G$, there exists $p \in N$ such that $b \in \chi(p)$; (2) for each edge $(b,d) \in E$, there exists $p \in N$ such that $\{b,d\} \subseteq \chi(p)$; and, (3) for each node $b$ of $G$, the set $\{p \in N \mid b \in \chi(p)\}$ induces a connected subtree.

### 4 Structural Restrictions (Alone) Do Not Help

Many NP-hard problems in different application areas, ranging, e.g., from Constraint Satisfaction to Database Theory, are known to be efficiently solvable when restricted to instances that can be modeled via (nearly)acyclic graphs. Therefore, one may naturally expect that these structural restrictions are also beneficial to isolate tractable MMUCAs. This is next investigated, by modeling interactions among bidders in an instance $A$ via the transformations and the goods graph of $A$, denoted by $\mathcal{T}(G(A)$ and $\mathcal{G}(A)$, respectively—see Section 1. In addition, we shall consider their undirected versions, denoted by $\overline{\mathcal{T}}(G(A)$ and $\overline{\mathcal{G}}(A)$, respectively.

Our first result is very bad news on transformations graphs. Indeed, the undirected transformations graph associated with the MMUC instance built in the proof of Theorem 3.5 is acyclic, while still encoding an NP-complete problem. Thus:

**Corollary 4.1** Feasibility is NP-complete, even restricted on the class $\{A \mid \overline{\mathcal{T}}(A)$ is acyclic$\}$ (and, hence, on the class $\{A \mid \mathcal{T}(G(A)$ is acyclic$\}$) and under atomic bids.

Similarly, the (directed) graphs graph associated with the instance in the proof of Theorem 3.5 is also acyclic. Thus:

**Corollary 4.2** Feasibility is NP-complete, even restricted on the class $\{A \mid \mathcal{T}(G(A)$ is acyclic$\}$ and under atomic bids.

In particular, the above result is rather interesting in the light that instances with (directed) acyclic goods graphs correspond to natural transformation processes (cf. [Oetens and Endriss, 2008]). Thus, transformation processes emerge to be as hard as arbitrary trades and exchanges of goods.

In order to conclude our analysis, we shall show that Feasibility remains NP-hard on nearly-acyclic undirected goods graphs. To this end, we need to recall the notion of tree decomposition, which is the most powerful generalization of graph acyclicity. Formally, a tree decomposition of a graph $G = (V,E)$ is a pair $(T, \chi)$, where $T = (N,F)$ is a tree, and $\chi$ is a labeling function assigning to each vertex $p \in N$ a set of vertices $\chi(p) \subseteq V$, such that the following conditions are satisfied: (1) for each node $b$ of $G$, there exists $p \in N$ such that $b \in \chi(p)$; (2) for each edge $(b,d) \in E$, there exists $p \in N$ such that $\{b,d\} \subseteq \chi(p)$; and, (3) for each node $b$ of $G$, the set $\{p \in N \mid b \in \chi(p)\}$ induces a connected subtree.
The width of $\langle T, \chi \rangle$ is the number $\max_{p \in N}(|\chi(p)| - 1)$. The treewidth of $G$, denoted by $tw(G)$, is the minimum width over all its tree decompositions. It is well-known that an undirected graph $G$ is acyclic if and only if $tw(G) = 1$.

**Theorem 4.3** Feasibility is NP-complete, even restricted on the class $\{A \mid tw(\GG(A)) = 2\}$ and under atomic bids.

**Proof.** (Sketch) Deciding whether there is a way to partition a bag $S = \{s_1, s_2, \ldots, s_n\}$ of integers into two bags $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ equals the sum of the numbers in $S_2$ is an NP-hard problem.

Let $m = \sum_{i=1}^{n} s_i$. Consider the auction $\mathcal{A}(S) = \langle G, T, U_{in}, U_{out} \rangle$ such that: $G = \{g_1, \ldots, g_n, c_1, c_2\}$, $U_{in} = \{g_1, \ldots, g_m\}$, $U_{out} = \bigcup_{i=1}^{m} \{c_1\} \cup \bigcup_{i=1}^{m} \{c_2\}$, and $T = \bigcup_{i=1}^{m} T_i$. In particular, for each number $s_i$, the set $T_i$ consists of the transformations: $\{(g_i, U_{in}^i \cup \{c_1\}), (g_i, U_{in}^i \cup \{c_2\})\}$. Observe that the transformation $\langle\{g_i, U_{in}^i \cup \{c_1\}\}\rangle$ is alternative to $\langle\{g_i, U_{in}^i \cup \{c_2\}\}\rangle$ (where $c_1$ and $c_2$ are meant to encode $S_1$ and $S_2$, respectively). Thus, each number can be assigned to one subset at most. Moreover, all the numbers must be assigned to a partition for the auctioneer to end up with $m/2$ copies of $c_1$ and $m/2$ copies of $c_2$. Therefore, a partition of $\mathcal{A}(S)$ exists $\iff \mathcal{A}(S)$ has a solution.

Eventually, the undirected goods graph associated with $\mathcal{A}(S)$ has treewidth 1. This is witnessed by the tree decomposition $T$ whose root $r$ is such that $\chi(r) = \{c_1, c_2\}$, and such that for each good $s_i (1 \leq i \leq n)$, exactly one (leaf) child $\ell_i$ of $r$ is in $T$ with $\chi(\ell_i) = \{c_1, c_2, g_i\}$. $\square$

For completeness, we note that the above result is tight, for it can be shown that Feasibility is tractable on the class $\{A \mid tw(\GG(A)) = 1\}$ (of acyclic undirected goods graphs). This class does not appear to model interesting problems yet, and hence the solution algorithm is omitted here.

We leave the section by noticing that XOR-conditions may be used to state in the above proof that $\langle\{g_i, U_{in}^i \cup \{c_1\}\}\rangle$ is alternative to $\langle\{g_i, U_{in}^i \cup \{c_2\}\}\rangle$, where $g_i$ is a copy of $g_i$, also available in input. Thus, one may easily derive the following:

**Theorem 4.4** Feasibility is NP-complete under XOR-conditions, even on the class $\{A \mid \GG(A)\}$ is acyclic.

## 5 Tractable Winner-Determination

Now that the complexity of the basic Feasibility problem has been analyzed, we can turn to isolate tractable classes for the Winner-Determination problem. Clearly enough, all the hardness results we have derived for Feasibility are inherited by Winner-Determination. In addition, computing the optimal solution remains hard even if there is no transformation producing (or requiring) goods. In fact, the result below follows by adapting classical results on combinatorial auctions (see, e.g., [Lehmann et al., 2006]).

**Theorem 5.1** Winner-Determination is NP-hard, even restricted on the classes $\{A \mid in-deg(A) = 0\}$ and $\{A \mid out-deg(A) = 0\}$, and under atomic bids.

### 5.1 Bounded Intricacy and Hypergraph Encodings

By summarizing the lessons learned from the NP-hardness results, we may note that there are two basic sources of complexities in MMUCAs, namely the fact that:

(a) goods may not suffice to “activate” all the transformations requiring them; and, that

(b) interactions among bidders may be complex to be analyzed.

To isolate tractable classes of Winner-Determination, we have hence to issue bounds on both these aspects.

Point (a) suggest to issue a bound on the number of transformations producing a given good $g$ (as to easily control its availability in any solution); plus either a bound on the maximum quantity of $g$ that can be produced over solutions, or a bound on the number of transformations requiring $g$ (as to easily control the consumption of $g$). This is formalized via the measure of intricacy of an instance $\langle G, T, U_{in}, U_{out} \rangle$, denoted by $intr(A)$, which is the value:

$$\max_{p \in \mathcal{P}} \left( |\{I, O, p \mid O(A) > 0\}| + \min \left\{ U_{in}(g) + \sum_{(S, O, p) \in T} O(g), \{\{I, O, p \mid I(A) > 0\} \right\} \right).$$

With respect to (b), we consider instead a structural restriction that is even more general than graph acyclicity, and which permits a suitable hypergraph representation of the interactions in $A$. In particular, we define the auction hypergraph $AH(A) = (T, H)$ as the hypergraph whose nodes are in one-to-one correspondence with the transformations in $A$, and where for each good $g \in G$, there is a hyperedge in $H$ such that: $h_g = \{\{I, O, p \mid I(g) > 0\} \in H\}$. The set of its nodes (resp., edges) is denoted by $N(H)$ (resp., $E(H)$). And, we consider the hypertree decomposition [Gottlob et al., 2002] approach to isolate nearly acyclic hypergraphs.

Formally, a hypertree for a hypergraph $H$ is a triple $\langle T, \chi, \lambda \rangle$, where $T = (N, E)$ is a rooted tree, and $\chi$ and $\lambda$ are labeling functions, which associate each vertex $p \in N$ with two sets $\chi(p) \subseteq N(H)$ and $\lambda(p) \subseteq E(H)$. For a set of edges $H \subseteq E(H)$, $N(H)$ denotes the set $\bigcup_{p \in H} h_g$. $T'$ is a subtree in $T$, we define $\chi(T') = \bigcup_{p \in N' \subseteq N} \chi(p)$. We denote the set of vertices of $T$ by $\text{vertices}(T)$. Moreover, for any $p \in N$, $T_p$ denotes the subtree of $T$ rooted at $p$.

**Definition 5.2** ([Gottlob et al., 2002]) A (complete) hypertree decomposition of a hypergraph $H$ is a hypertree $HD = \langle T, \chi, \lambda \rangle$ for $H$ satisfying the following conditions:

1. for each edge $h \in E(H)$, there exists $p \in \text{vertices}(T)$ such that $h \subseteq \chi(p)$, and $h \in \lambda(p)$;
2. for each node $Y \in N(H)$, the set $\{p \in \text{vertices}(T) \mid Y \in \chi(p)\}$ induces a (connected) subtree of $T$;
3. for each $p \in \text{vertices}(T)$, $\chi(p) \subseteq N(H)$;
4. for each $p \in \text{vertices}(T)$, $N(\chi(p)) \cap \chi(T_p) \subseteq \chi(p)$.

The width of a hypertree decomposition $\langle T, \chi, \lambda \rangle$ is $\max_{p \in \text{vertices}(T)} |\lambda(p)|$. The hypertree width $\text{hw}(H)$ of $H$ is the minimum width over all its hypertree decompositions. $\square$

Given a hypergraph $H$, deciding the existence of a $k$-width hypertree decomposition and computing one (if any) are feasible in time $O(|\text{edges}(H)|^k \times |N(H)|^2)$.
Let $C^*(h, k)$ denote the class of instances whose intricacy is bounded by $h$ and whose auction hypergraphs have hypertree width bounded by $k$. We next show that WINNER-DETERMINATION is tractable on $C^*(h, k)$, by means of the algorithm SolveMMUCA$_{h,k}$ shown in Figure 3.

In the algorithm, any solution $\sigma$ is encoded as a set of $|T|$ pairs of the form $(t, i)$ where $t \in T$ and $i$ is the step where $t$ is executed ($i = 0$ means that $t$ is not applied in $\sigma$). Moreover, for each vertex $v$ of a hypertree decomposition, a $v$-solution (w.r.t. $\sigma$) is defined as the set $\{(t, i) \mid (t, i) \in \sigma \land t \in \chi(v)\}$. The set of all the possible $v$-solutions is denoted by $H_v$. And, for any $\sigma' \subseteq \sigma$, we let $\nu(\sigma) = \sum_{i}(t, i, p) \in \sigma, i > 0 \ p$.

The algorithm receives as input a $h$-width hypertree decomposition $HD = (T=(N, E), \chi, \lambda)$ of $\mathbf{AH}(A)$, and it manipulates $v$-solutions, for each vertex $v$, by looking for their “conformance” with $c$-solutions in $H_v$, for each children $c$ of $v$ in $T$: $v \in H_v$ conforms with $c \in H_c$, denoted by $\sigma_c$ if: for each $t \in \chi(v) \cap \chi(c), (t, i) \in \sigma_v \Leftrightarrow (t, i) \in \sigma_c$. The $c$-solution $\sigma_c$ on which the maximum is achieved is stored in $\sigma_c$.

In the second phase, the tree $T$ is processed starting from the root. Firstly, the solution $\sigma^*$ is defined as the $v$-solution in $H_v$ with the maximum payoff. Then, procedure $\text{TopDown}$ extends $\sigma^*$, at each vertex $v$, with the $c$-solution $\sigma_{c,v,c}$, for each child $c$ of $v$.

For each vertex $v$, a rough upper bound on the size of $H_v$ is $|T|^{2h \times k}$. Hence, the following can be obtained.

**Theorem 5.3** Winner-Determination can be solved in time $O(|G| \times |T|^{4h \times k})$ under OR-conditions and on the class $C^*(h, k)$.

In fact, SolveMMUCA$_{h,k}$ can be used even under XOR-conditions, after encoding them in terms of transformations. E.g., if $t_1$ and $t_2$ are alternative because of a XOR-condition, we may simply add one good $g_{1,2}$ and force $t_1$ and $t_2$ to require $g_{1,2}$ for their application.

We conclude by noticing that with of classical CAs (where one item of each type of good is available, and no good can be produced), $\text{intr}(A) = 1$ holds. In fact, SolveMMUCA$_{h,k}$ degenerates to the algorithm in [Gottlob and Greco, 2007].

### 6 Conclusion

The problem of identifying tractability islands for MMUCAs has been faced, by complementing tractability results that were derived for classical CAs. Our results paves the way for the implementation of solution algorithms for arbitrary MMUCA instances, which can take advantage of structural and qualitative properties of some of their portions. In addition, interesting avenues of further research are to chart the tractability frontier under lack of free disposal (i.e., when the auctioneer wants to end up exactly with the goods in $U_{out}$), and to consider more involved kinds of bidding languages, where for instance bidders preferences are taken into account.

### References


