Charting the Tractability Frontier of Mixed Multi-Unit Combinatorial Auctions

Valeria Fionda
Dip. di Elettronica, Informatica e Sistemistica
Università della Calabria
I-87036 Rende, Italy
fionda@deis.unical.it

Gianluigi Greco
Dip. di Matematica
Università della Calabria
I-87036 Rende, Italy
ggreco@mat.unical.it

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 MMUCA 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 automating *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 $\langle \mathcal{I}, \mathcal{O}, p \rangle$ where $\mathcal{I} \in \mathbb{N}^{|G|}$ (resp., $\mathcal{O} \in \mathbb{N}^{|G|}$) 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 $p \in \mathbb{R}$ is the payment the bidder is willing to make in return for being allocated the transformation[1]. Then, a mixed multi-unit combinatorial auction instance is a tuple $\langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out} \rangle$ where $\mathcal{T}$ is a multi-set of transformations over G, and $\mathcal{U}_{in} \in \mathbb{N}^{|G|}$ (resp., $\mathcal{U}_{out} \in \mathbb{N}^{|G|}$) is a vector denoting the quantities of goods the auctioneer holds to begin with (resp., expects to end up with). Solving a MMUCA instance $\langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out} \rangle$ amounts to finding 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).

The above problem has been intensively 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 MMUCA 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] If $p < 0$, then the auctioneer must actually pay $-p$ to the bidder in order for she to implement the transformation.

(1) In the case where each bidder submits a multi-set of transformations and accepts 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.

(2) In the case where each bidder accepts at most one transformation from the multi-set of hers submitted transformations (short: XOR-language), we show that FEASIBILITY is tractable *if and only if* every type of good is required as input by one transformation at most.

Then, we turn to consider *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:

(3) We show that FEASIBILITY is hard on (nearly)acyclic instances too. In particular, this is the case for two natural ways of encoding bidder interactions, namely for *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 *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 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:

(4) On the one hand, WINNER-DETERMINATION emerges to be intractable under most kinds of qualitative restrictions, for it inherits all the intractability results that hold with FEASIBILITY as well as with classical CAs.

(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 FEASIBILITY under qualitative and structural restrictions is discussed in Section 3 and Section 4, respectively. Tractability islands for the 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 $\mathcal{W} \in \mathbb{N}^{|G|}$, we denote by $\mathcal{W}(g)$ the element of $\mathcal{W}$ referring to any good $g \in G$. Sometimes, $\mathcal{W}$ will be viewed as a multi-set over G.

Let $\mathcal{A} = \langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out} \rangle$ be a MMUCA instance. Solving $\mathcal{A}$ amounts to deciding which transformations have to be accepted, and in which order to implement them.

Formally, consider a sequence of transformations $\sigma = \langle \mathcal{I}_1, \mathcal{O}_1, p_1 \rangle, ..., \langle \mathcal{I}_k, \mathcal{O}_k, p_k \rangle$ such that $\langle \mathcal{I}_i, \mathcal{O}_i, p_i \rangle \in \mathcal{T}$, for each $i \in \{1, ..., k\}$. Let $\mathcal{M}_0 = \mathcal{U}_{in}$ denote the quantities of goods initially hold by the auctioneer, and let $\mathcal{M}_i \in \mathbb{N}^{|G|}$ be the quantities owned after the i-th transformation, i.e., $\forall g \in G, \mathcal{M}_i(g) = \mathcal{M}_{i-1}(g) + \mathcal{O}_i(g) - \mathcal{I}_i(g)$.

Then, σ is *legal* w.r.t. $\mathcal{A}$ if $\mathcal{M}_{i-1}(g) \geq \mathcal{I}_i(g)$, for each $i \in \{1, ..., k\}$ and $g \in G$. The *revenue* of the auctioneer with σ 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 σ is a *solution* to $\mathcal{A}$ if $\forall g \in G, \mathcal{M}_k(g) \geq \mathcal{U}_{out}(g)$. 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.

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 multi-sets of transformations under conditions of two kinds.

An OR-*condition* on a multi-set $S \subseteq \mathcal{T}$ states that the bidder may accept any transformations in 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 $\mathcal{A}$ with a set $\mathcal{L}$ of c-conditions (with $c \in \{\text{OR}, \text{XOR}\}$, $\bigcup_{S \in \mathcal{L}} S = \mathcal{T}$, and $S \cap S' = \emptyset, \forall S, S' \in \mathcal{L}$) is denoted by $\mathcal{A}_{[\mathcal{L},c]}$.

3 FEASIBILITY and Qualitative Restrictions

In this section, we start the analysis of the complexity of 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:

- $\textit{in-var}(\mathcal{A}) = \max_{\langle \mathcal{I}, \mathcal{O}, p \rangle \in \mathcal{T}} |\{\mathcal{I}(g) \mid g \in G, \mathcal{I}(g) > 0\}|$ is the *input variety* of $\mathcal{A}$, i.e., the maximum number of types of goods required as input over all transformations. Symmetrically, the *output variety* is the value $\textit{out-var}(\mathcal{A}) = \max_{\langle \mathcal{I}, \mathcal{O}, p \rangle \in \mathcal{T}} |\{\mathcal{O}(g) \mid g \in G, \mathcal{O}(g) > 0\}|$.
- $\textit{in-mul}(\mathcal{A}) = \max_{\langle \mathcal{I}, \mathcal{O}, p \rangle \in \mathcal{T}, g \in G} \mathcal{I}(g)$ is the *input multiplicity* of $\mathcal{A}$, i.e., the maximum quantity of any good required as input over all transformations. And, symmetrically, the *output multiplicity* is the value $\textit{out-mul}(\mathcal{A}) = \max_{\langle \mathcal{I}, \mathcal{O}, p \rangle \in \mathcal{T}, g \in G} \mathcal{O}(g)$.
- $\textit{in-deg}(\mathcal{A}) = \max_{g \in G} |\{\langle \mathcal{I}, \mathcal{O}, p \rangle \in \mathcal{T} \mid \mathcal{O}(g) > 0\}|$ is the *input degree* of $\mathcal{A}$, i.e., the maximum number of transformations producing a given good over all goods. Symmetrically, the *output degree* of $\mathcal{A}$ is the value $\textit{out-deg}(\mathcal{A}) = \max_{g \in G} |\{\langle \mathcal{I}, \mathcal{O}, p \rangle \in \mathcal{T} \mid \mathcal{I}(g) > 0\}|$.

Below, $\mathcal{C}(\texttt{iv}, \texttt{ov}, \texttt{im}, \texttt{om}, \texttt{id}, \texttt{od})$ will denote the class of instances $\mathcal{A}$ such that: $\textit{in-var}(\mathcal{A}) \leq \texttt{iv}$, $\textit{out-var}(\mathcal{A}) \leq \texttt{ov}$, $\textit{in-mul}(\mathcal{A}) \leq \texttt{im}$, $\textit{out-mul}(\mathcal{A}) \leq \texttt{om}$, $\textit{in-deg}(\mathcal{A}) \leq \texttt{id}$, and $\textit{out-deg}(\mathcal{A}) \leq \texttt{od}$. Also, the symbol ∞ is used to denote that no bound is issued on some given parameter.

iv	ov	im	om	id	od	**Result (OR)**
1	1	1	1	∞	∞	in P
∞	∞	∞	∞	1	∞	in P
∞	∞	∞	∞	∞	1	in P
2	1	1	1	2	2	NP-complete
1	2	1	1	2	2	NP-complete
1	1	2	1	2	2	NP-complete
1	1	1	2	2	2	NP-complete

iv	ov	im	om	id	od	**Result (XOR)**
∞	∞	∞	∞	1	∞	in P
1	1	1	1	2	1	NP-complete

Figure 1: FEASIBILITY and Qualitative Restrictions.

Results. A summary of our analysis is reported in Figure 1: Our results clearly depict a tractability frontier, since relaxing any condition in a tractable scenario leads to intractability and since FEASIBILITY trivializes when turning some parameter from 1 to 0 (details on these trivial cases are omitted). Note also that the expressiveness of the XOR-language [Cerquides *et al.*, 2007] is payed in terms of a smaller tractability island.

In the remainder, we overview the proofs of these results, by starting with OR-conditions. In particular, we shall consider a "normal form" for MMUCAs where for each good $g \in G$ such that $\mathcal{U}_{out}(g) > 0$ (resp., $\mathcal{U}_{in}(g) > 0$), there is no transformation $\langle \mathcal{I}, \mathcal{O}, p\rangle \in \mathcal{T}$ such that $\mathcal{I}(g) > 0$ (resp., $\mathcal{O}(g) > 0$) holds—indeed, any instance can be modified in polynomial time as to meet this requirement.

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 $\mathcal{A} = \langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out}\rangle_{[\mathcal{L},\text{OR}]}$ be a MMUCA such that $\mathcal{A} \in \mathcal{C}(1,1,1,1,\infty,\infty)$. Then,* FEASIBILITY *can be solved in time $O\left(|G|^3\right)$.*

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 $\langle \mathcal{I}_h, \mathcal{O}_h, p_h\rangle \in \mathcal{T}$ such that $\mathcal{I}_h(g) > 0$ and $\mathcal{O}_h(g') > 0$; 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., [Kleinberg and Tardos, 2005]) where:

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

(ii) each node g has demand $d(g) = \mathcal{U}_{out}(g) - \mathcal{U}_{in}(g)$, while n has demand $d(n) = \sum_{g\in G}(\mathcal{U}_{in}(g) - \mathcal{U}_{out}(g))$.

It can be shown that $\mathcal{A}$ has a solution if and only if there is a *circulation* on (N, E), i.e., a function $f : E \mapsto \mathbb{R}$ such that $f(v, v') \leq u(v, v')$ for each $(v, v') \in E$, and $\sum_{s\in N} f(s, v) - \sum_{t\in N} f(v, t) = d(v)$, for each $v \in N$.

Eventually, the existence of a circulation can be checked in $O(|G|^3)$, where $(|G| + 1)$ is the number of nodes in N. In fact, note that OR-conditions do not play any role here. □

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 the input degree is unitary at most. Moreover, let $\Delta = \max_g(\mathcal{U}_{in}(g) + \sum_{\langle \mathcal{I},\mathcal{O},p\rangle\in\mathcal{T}} \mathcal{O}(g))$ be an upper bound on the number of any good produced over all solutions.

Theorem 3.2 *Let $\mathcal{A} = \langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out}\rangle_{[\mathcal{L},\text{OR}]}$ be a MMUCA such that $\mathcal{A} \in \mathcal{C}(\infty,\infty,\infty,\infty,1,\infty)$. Then,* FEASIBILITY *can be solved in time $O(|\mathcal{T}|^2 \times |G| \times \log\Delta)$.*

Proof. *(Sketch).* Let $\mathcal{T}_r \subseteq \mathcal{T}$ denote the set of all the *necessary* transformations w.r.t. some good g with $\mathcal{U}_{out}(g) > 0$, built as follows. Initially, $\mathcal{T}_r$ includes all the transformations that may produce output goods only; then, $\mathcal{T}_r$ is iteratively updated (until the fixed point is reached) by including those transformations that may produce goods required in input by some transformation already in $\mathcal{T}_r$. Since $\textit{in-deg}(\mathcal{A}) \leq 1$, $\mathcal{A}$ admits a solution if and only if it is possible to execute *all* the transformations in $\mathcal{T}_r$. Thus, we may simple start applying transformations in $\mathcal{T}_r$ till a step k is reached such all the transformations in $\mathcal{T}_r$ are applied (thereby witnessing that there is a solution to $\mathcal{A}$), or there is some transformation in $\mathcal{T}_r$ that cannot be applied (so that there is no solution to $\mathcal{A}$). In particular, note that the application of any transformation does not interfere with the applicability of other transformations (since $\textit{in-deg}(\mathcal{A}) \leq 1$). Hence, the execution order of applicable transformations is not relevant. Eventually, $O(|\mathcal{T}|)$ steps are required, each one feasible in $O(|\mathcal{T}| \times |G| \times \log\Delta)$, where in particular the logarithmic factor accounts for the complexity of executing algebraic manipulations over the numbers involved in the computation. □

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 $\mathcal{A} = \langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out}\rangle_{[\mathcal{L},\text{OR}]}$ be a MMUCA such that $\mathcal{A} \in \mathcal{C}(\infty,\infty,\infty,\infty,\infty,1)$. Then,* FEASIBILITY *can be solved in time $O(|\mathcal{T}|^2 \times |G| \times \log\Delta)$.*

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 $\Phi = c_1 \wedge \ldots \wedge c_m$ over the variables $X_1, \ldots, X_n$ is satisfiable is NP-hard. Below, we state the intractability of a specific class of Boolean formulas.

Lemma 3.4 SATISFIABILITY *is* NP*-hard, even if each variable occurs positively in at most two clauses and negatively in at most one other clause, and if each clause contains three variables at most.*

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

Proof. *(Sketch).* Membership in 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 Φ be a formula satisfying the conditions in Lemma 3.4, and $\mathcal{A}(\Phi) = \langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out}\rangle$ be such that: $G = \bigcup_{i=1}^{n}\{X_i, X_i^T, X_i^F\} \cup \{c_1, ..., c_m\} \cup \{c_j^i \mid X_i \text{ occurs in } c_j\}$, $\mathcal{U}_{in} = \{X_1, ..., X_n\}$, $\mathcal{U}_{out} = \{c_1, ..., c_m\}$, and $\mathcal{T} = \bigcup_{i=1}^{n}\mathcal{T}_i$, where $\mathcal{T}_i$ (for $1 \leq i \leq n$) is defined next.

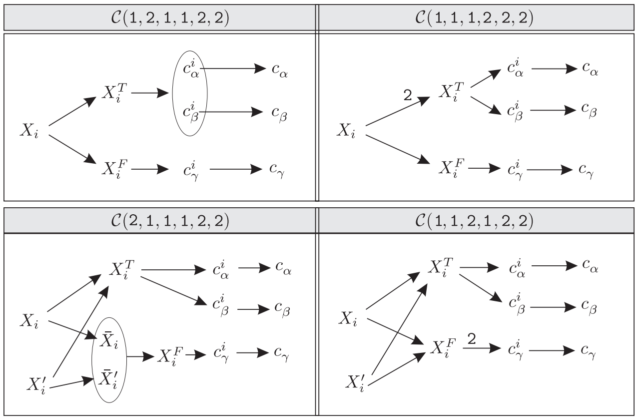

Figure 2: Hardness results: $\mathcal{T}_i$ in the various reductions.

For each variable X_i occurring positively in the clauses c_α and c_β while occurring negatively in c_γ, the set $\mathcal{T}_i$ consists of the transformations[2] graphically depicted in Figure 2—adaptations to the cases where X_i does not negatively occur in a clause and positively in two further clauses are trivial and, hence, omitted. Observe that two transformations occur in $\mathcal{T}_i$ requiring X_i as input, one that produces X_i^T and another that produces X_i^F. 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 $\{c_1,..,c_m\}$ if and only if all the various selections encode a satisfying assignment to Φ. Thus, Φ *is satisfiable* $\Leftrightarrow \mathcal{A}(\Phi)$ *has a solution.*

Actually, observe that $\mathcal{A}(\Phi)$ belongs to $\mathcal{C}(1,2,1,1,3,2)$, but not necessarily to $\mathcal{C}(1,2,1,1,2,2)$ since a clause c_j may contain three variables $X_i, X_{i'}$, and $X_{i''}$. In order to face this case, we may add one further good $\bar{c}_j$, and replace the two transformations $\langle\{c_j^{i'}\},\{c_j\}\rangle$ and $\langle\{c_j^{i''}\},\{c_j\}\rangle$ with the transformations: $\langle\{c_j^{i'}\},\{\bar{c}_j\}\rangle$, $\langle\{c_j^{i''}\},\{\bar{c}_j\}\rangle$, and $\langle\{\bar{c}_j\},\{c_j\}\rangle$. Clearly, the resulting auction has input degree equals to 2, while keeping unchanged the values of all the other parameters and the properties of the reduction. □

The other hardness results can be shown with a similar line of reasoning, provided the modifications on the auction $\mathcal{A}(\Phi)$ that are graphically illustrated in Figure 2. Thus, the following holds, whose proof is omitted due to space constraints.

Theorem 3.6 FEASIBILITY *is* NP*-complete, even under atomic bids and restricted on the classes* $\mathcal{C}(1,1,1,2,2,2)$*,* $\mathcal{C}(1,1,2,1,2,2)$*, and* $\mathcal{C}(2,1,1,1,2,2)$.

3.3 Results for the XOR-language

We now conclude the analysis by considering the case of XOR-conditions. Firstly, we observe that the proof of Theorem 3.2 can easily be adapted to deal with XOR-conditions, based on the fact that executing necessary transformations is mandatory as to solve the instance at hand; thus, if some necessary transformation cannot be executed because of some given XOR-condition (whose check is a source of additional complexity), then no solution exists.

Theorem 3.7 *Let* $\mathcal{A} = \langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out}\rangle_{[\mathcal{L},\text{XOR}]}$ *be a MMUCA such that* $\mathcal{A} \in \mathcal{C}(\infty,\infty,\infty,\infty,1,\infty)$*. Then,* FEASIBILITY *can be solved in time* $O(|\mathcal{T}|^2 \times (|G| \times \log\Delta + |\mathcal{T}|))$.

Interestingly, the above tractability cannot be extended on the dual class $\mathcal{C}(\infty,\infty,\infty,\infty,\infty,1)$, since XOR conditions may be used to simulate scenarios where every type of good is required as input by more than one transformation.

Theorem 3.8 FEASIBILITY *is* NP*-complete under XOR-conditions, even restricted on the class* $\mathcal{C}(1,1,1,1,2,1)$.

4 Structural Restrictions (Alone) Do Not Help

Many NP-hard problems can efficiently be solved when restricted to instances that can be modeled via (nearly)acyclic graphs. Thus, one may expect that these structural restrictions are also beneficial to isolate tractable MMUCAs. This is next investigated, by modeling interactions among bidders in an instance $\mathcal{A}$ via the transformations and the goods graph of $\mathcal{A}$, denoted by $\mathsf{TG}(\mathcal{A})$ and $\mathsf{GG}(\mathcal{A})$, respectively—see Section 1. In addition, we shall consider their undirected versions, denoted by $\overline{\mathsf{TG}}(\mathcal{A})$ and $\overline{\mathsf{GG}}(\mathcal{A})$, respectively.

Beforehand, note that some of the qualitative restrictions studied in Section 3 induce structural restrictions on the transformations and goods graphs. Indeed, it is easily checked that unitary input (or output) degree instances are guaranteed to be associated with graphs that are basically acyclic—cycles there cannot contribute to producing goods. In fact, these cases are tractable (cf. Theorem 3.2 and Theorem 3.3). In general, however, acyclicity does not guarantee tractability. Indeed, the undirected transformations graph associated with the MMUCA 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* $\{\mathcal{A} \mid \overline{\mathsf{TG}}(\mathcal{A})$ *is acyclic*$\}$ *(and, hence, on the class* $\{\mathcal{A} \mid \mathsf{TG}(\mathcal{A})$ *is acyclic*$\}$*) and under atomic bids.*

In addition, the (directed) goods graph in the proof of Theorem 3.5 is also acyclic. Thus:

Corollary 4.2 FEASIBILITY *is* NP*-complete, even restricted on the class* $\{\mathcal{A} \mid \mathsf{GG}(\mathcal{A})$ *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. [Ottens and Endriss, 2008]). Thus, transformation processes emerge to be as hard as arbitrary trades and exchanges of goods.

In order to complete our analysis, we shall show that FEASIBILITY remains NP-hard on nearly-acyclic undirected goods graphs—formally, graphs of bounded treewidth. To this end, we recall here that a *tree decomposition* of a graph $G = (V, E)$ is a pair $\langle T, \chi\rangle$, where $T = (N, F)$ is a tree, and χ 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.

[2]Given our interest in the FEASIBILITY problem, we shall omit the indication of the payments in the various transformations.

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* $\{\mathcal{A} \mid tw(\overline{\mathsf{GG}}(\mathcal{A})) = 2\}$ *and under atomic bids.*

Proof. *(Sketch).* Deciding whether there is a way to partition a multi-set $S = \{s_1, s_2, ..., s_n\}$ of integers into two multi-sets 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 a well-known NP-hard problem.

Let $m = \sum_{i=1}^{n} s_i$. Consider the auction $\mathcal{A}(S) = \langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out} \rangle$ such that: $G = \{g_1, ..., g_n, c_1, c_2\}$, $\mathcal{U}_{in} = \{g_1, ..., g_n\}$, $\mathcal{U}_{out} = \bigcup_{i=1}^{\frac{m}{2}}\{c_1\} \cup \bigcup_{i=1}^{\frac{m}{2}}\{c_2\}$, and $\mathcal{T} = \bigcup_{i=1}^{n} \mathcal{T}_i$. In particular, for each number s_i, the set $\mathcal{T}_i$ consists of the transformations: $\langle \{g_i\}, \bigcup_{i=1}^{s_i}\{c_1\} \rangle$, $\langle \{g_i\}, \bigcup_{i=1}^{s_i}\{c_2\} \rangle$.

Since the transformation $\langle \{g_i\}, \bigcup_{i=1}^{s_i}\{c_1\} \rangle$ is alternative to $\langle \{g_i\}, \bigcup_{i=1}^{s_i}\{c_2\} \rangle$ (where c_1 and c_2 are meant to encode S_1 and S_2, respectively), *a partition of* S *exists* $\Leftrightarrow$ $\mathcal{A}(S)$ *has a solution.* Eventually, we also note that the undirected goods graph associated with $\mathcal{A}(S)$ has treewidth equals to 2, which is witnessed by the tree decomposition T whose root r is such that $\chi(r) = \{c_1, c_2\}$, and where for each good s_i, exactly one (leaf) child ℓ_i of r is in T with $\chi(\ell_i) = \{c_1, c_2, g_i\}$. □

We leave the section by noticing that XOR-conditions may be used to state in the above proof that $\langle \{g_i\}, \bigcup_{i=1}^{s_i}\{c_1\} \rangle$ is alternative to $\langle \{g_i'\}, \bigcup_{i=1}^{s_i}\{c_2\} \rangle$, 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* $\{\mathcal{A} \mid \overline{\mathsf{GG}}(\mathcal{A})$ *is acyclic*$\}$.

5 Tractable WINNER-DETERMINATION

When turning to WINNER-DETERMINATION, we firstly observe that this problem inherits all the hardness results derived for FEASIBILITY. 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* $\{\mathcal{A} \mid \textit{in-deg}(\mathcal{A}) = 0\}$ *and* $\{\mathcal{A} \mid \textit{out-deg}(\mathcal{A}) = 0\}$*, and under atomic bids.*

The reason for the above intractability is that interactions among bidders may be too complex to be analyzed. To bound these interactions, we next consider a structural restriction based on a hypergraph encoding, which is more general than graph (near)acyclicity. In particular, we define the *auction hypergraph* $\mathsf{AH}(\mathcal{A}) = (\mathcal{T}, H)$ as the hypergraph whose nodes are in one-to-one correspondence with the transformations in $\mathcal{A}$, and where for each good $g \in G$, there is a hyperedge in H such that: $h_g = \{\langle \mathcal{I}, \mathcal{O}, p \rangle \in \mathcal{T} \mid \mathcal{I}(g) + \mathcal{O}(g) > 0\}$—in the following, for any hypergraph $\mathcal{H}$, the set of its nodes (resp., edges) is denoted by $\mathcal{N}(\mathcal{H})$ (resp., $\mathcal{E}(\mathcal{H})$). On such hypergraphs, the idea is to consider the *hypertree decomposition* [Gottlob *et al.*, 2002] approach to isolate nearly acyclic auction hypergraphs.

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

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

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

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

Computing a k-width decomposition of $\mathcal{H}$ (or stating that it does not exist) is feasible in $O(|\mathcal{E}(\mathcal{H})|^{2k} \times |\mathcal{N}(\mathcal{H})|^2)$. □

Actually, we have noticed in Section 4 that structural restrictions does not suffice to single out tractable classes of instances. In particular, acyclic instances may still require complex reasoning tasks on the transformations that have to be activated, because of the large number of goods in the scenario. Thus, in addition to considering (generalizations of) acyclicity, we must also define a bound 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 $\mathcal{A} = \langle G, \mathcal{T}, \mathcal{U}_{in}, \mathcal{U}_{out} \rangle$, that is: $\max_{g \in G}(|\{\langle \mathcal{I}, \mathcal{O}, p \rangle \mid \mathcal{O}(g) > 0\}| + \min\{\mathcal{U}_{in}(g) + \sum_{\langle \mathcal{I}, \mathcal{O}, p \rangle \in \mathcal{T}} \mathcal{O}(g),\ |\{\langle \mathcal{I}, \mathcal{O}, p \rangle \mid \mathcal{I}(g) > 0\}|\})$.

Putting It All Together. Let $\mathcal{C}^*(h, k)$ 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 $\mathcal{C}^*(h, k)$, by means of the algorithm SolveMMUCA$_{h,k}$ shown in Figure 3. In the algorithm, any sequence σ is encoded as a set of pairs of the form (t, i) where $t \in \mathcal{T}$ and i is the step where t is executed, by assuming w.l.o.g. that transformations can be applied in parallel ($i = 0$ means that t is not applied in σ). Moreover, we say that σ is *legal over a good* $g \in G$ if $\mathcal{M}_{i-1}(g) \geq \mathcal{I}_i(g)$, for each $(\langle \mathcal{I}, \mathcal{O}, p \rangle, i) \in \sigma$ with $i > 0$, and if $\mathcal{M}_k(g) \geq \mathcal{U}_{out}(g)$ where k is the last step in σ. And, finally, for each $\sigma' \subseteq \sigma$, we define $pay(\sigma') = \sum_{(\langle \mathcal{I}, \mathcal{O}, p \rangle, i) \in \sigma', i > 0} p$.

The algorithm receives a k-width hypertree decomposition $HD = \langle T{=}(N, E), \chi, \lambda \rangle$ of $\mathsf{AH}(\mathcal{A})$. Firstly, for each vertex $v \in N$, it computes the set H_v of all the v-*solutions*, where $\{(t, i) \mid (t, i) \in \sigma \wedge t \in \chi(v)\}$ is a v-solution if there is a sequence σ that is legal over each good g such that $h_g \in \lambda(v)$.

Input: A k-width decomposition $HD = \langle T{=}(N, E), \chi, \lambda \rangle$ of $\mathsf{AH}(\mathcal{A})$;
Output: An optimal solution to WINNER-DETERMINATION on $\mathcal{A}$;
var σ^* : optimal solution;
  $\ell^v_{\sigma_v}$: real number, for each v-solution $\sigma_v \in H_v$;
  $\sigma_{\sigma_v, c}$: c-solution in H_c, for each v-solution σ_v and $(v, c) \in E$;

$H_v :=$ the set of all the v-solutions, for each $v \in N$;
$Done :=$ the set of all the leaves of T;
while $\exists v \in T \setminus Done$ such that $\{c \mid c$ is child of $v\} \subseteq Done$ **do**
  $H_v := H_v - \{\sigma_v \mid \exists (v, c) \in E \wedge \nexists \sigma_c \in H_c$ s.t. $\sigma_v \approx \sigma_c\}$;
  for each $\sigma_v \in H_v$ **do**
    $\ell^v_{\sigma_v} := pay(\sigma_v)$;
    for each $(v, c) \in E$ **do**
      $\bar{\sigma}_c := \arg\max_{\sigma_c \in H_c \mid \sigma_v \approx \sigma_c} \left(\ell^c_{\sigma_c} - pay(\sigma_c \cap \sigma_v) \right)$;
      $\sigma_{\sigma_v, c} := \bar{\sigma}_c$; $\ell^v_{\sigma_v} := \ell^v_{\sigma_v} + \ell^c_{\bar{\sigma}_c} - pay(\bar{\sigma}_c \cap \sigma_v)$;
    end for
  end for
  $Done := Done \cup \{v\}$;
end while
let r be the root of T; **if** $H_r = \emptyset$ **then** HALT answering 'no solution';
$\bar{\sigma}_r := \arg\max_{\sigma_r \in H_r} \ell^r_{\sigma_r}$, and $\sigma^* := \bar{\sigma}_r$;
$TopDown(r, \sigma_r)$;
return σ^*;

Procedure $TopDown(v$: vertex of $N, \bar{\sigma}_v \in H_v)$;
begin
  for each $c \in N$ s.t. $(v, c) \in E$ **do**
    $\bar{\sigma}_c := \sigma_{\bar{\sigma}_v, c}$; $\sigma^* := \sigma^* \cup \bar{\sigma}_c$;
    $TopDown(c, \bar{\sigma}_c)$;
  end for
end;

Figure 3: **Algorithm** SolveMMUCA$_{h,k}$.

Then, it manipulates v-solutions, at each vertex v, by looking for their "conformance" with c-solutions in H_c, for each child c of v in T, where $\sigma_v \in H_v$ *conforms* with $\sigma_c \in H_c$ ($\sigma_v \approx \sigma_c$) if $\forall t \in \chi(v) \cap \chi(c)$, $(t, i) \in \sigma_v \Leftrightarrow (t, i) \in \sigma_c$.

In particular, in the first phase, vertices of T are processed from the leaves to the root r as to filter v-solutions that do not conform with any solution at some child of v, and as to update the weight $\ell^v_{\sigma_v}$ of each remaining v-solution σ_v. Intuitively, $\ell^v_{\sigma_v}$ stores the maximum revenue over all potential solutions restricted on transformations in $\chi(T_v)$. Indeed, if v is a leaf, then $\ell^v_{\sigma_v} = pay(\sigma_v)$. Otherwise, for each child c of v in T, $\ell^v_{\sigma_v}$ is incremented by the maximum of $\ell^c_{\sigma_c} - pay(\sigma_c \cap \sigma_v)$ over all c-solutions conforming with σ_v. The c-solution over which the maximum is achieved is stored in $\sigma_{\sigma_v, c}$. In a second phase, T is processed starting from the root: The solution σ^* is defined as the r-solution in H_r with the maximum payoff, and procedure $TopDown$ extends σ^*, at each vertex v, with the c-solution $\sigma_{\sigma_v, c}$ for each $(v, c) \in E$.

As to analyze SolveMMUCA$_{h,k}$, the crucial observation is that for each vertex v, $|H_v| \leq (|\mathcal{T}| + 1)^{2 \times h \times k}$.

Theorem 5.3 WINNER-DETERMINATION *can be solved in* $O(|G|^{2k} \times (|\mathcal{T}| + 1)^{4 \times h \times k + 2} \times \log \Delta)$ *under OR-conditions and on* $\mathcal{C}^*(h, k)$*—with the uniform cost model for payoffs.*

As a final remark, observe that SolveMMUCA$_{h,k}$ can be used even under XOR-conditions, after encoding them in terms of transformations. For instance, 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. Moreover, the restriction of SolveMMUCA$_{h,k}$ on $h = 1$ can be used to solve the tractable class of CA instances (where one item of each type is available at most, and no good can be produced) identified 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 may 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 $\mathcal{U}_{out}$), to consider more involved kinds of bidding languages, for instance where bidders preferences are taken into account, and to look for approximability results for the various NP-hard scenarios we have singled out, in the spirit of the thorough analysis recently carried for combinatorial exchange problems [Babaioff *et al.*, 2008].

References

[Babaioff *et al.*, 2008] M. Babaioffs, P. Briest, and P. Krysta. On the Approximability of Combinatorial Exchange Problems. In *Proc. of SAGT'08*, pp. 83–94, 2008.

[Cerquides *et al.*, 2007] J. Cerquides, U. Endriss, A. Giovannucci, and J. A. Rodriguez-Aguilar. Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In *Proc. of IJCAI'07*, pp. 1221–1226, 2007.

[Conitzer *et al.*, 2004] V. Conitzer, J. Derryberry, and T. Sandholm. Combinatorial auctions with structured item graphs. In *Proc. of AAAI'04*, pp. 212–218, 2004.

[Giovannucci *et al.*, 2007] A. Giovannucci, J. A. Rodriguez-Aguilar, M. Vinyals, J. Cerquides, and U. Endriss. Mixed multi-unit combinatorial auctions for supply chain management. *SIGecom Exch.*, 7(1):58–60, 2007.

[Gottlob and Greco, 2007] G. Gottlob and G. Greco. On the complexity of combinatorial auctions: structured item graphs and hypertree decomposition. In *Proc. of EC'07*, pp. 152–161, 2007.

[Gottlob *et al.*, 2002] G. Gottlob, N. Leone, and S. Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 63(3):579–627, 2002.

[Kleinberg and Tardos, 2005] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison Wesley, 2005.

[Lehmann *et al.*, 2006] D. Lehmann, R. Müller, and T. Sandholm. The Winner Determination Problem. *Combinatorial Auctions*. MIT Press, 2006.

[Ottens and Endriss, 2008] B. Ottens and U. Endriss. Comparing winner determination algorithms for mixed multi-unit combinatorial auctions. In *Proc. of AAMAS'08*, pp. 1601–1604, 2008.

[Walsh and Wellman, 2003] W.E. Walsh and M.P. Wellman. Decentralized supply chain formation: A market protocol and competititve equilibrium analysis. *Journal of Aritificial Intelligence Research*, 19:513–567, 2003.