Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)

Modeling and Reasoning about NTU Games via Answer Set Programming

Giovanni Amendola, Gianluigi Greco, Nicola Leone, and **Pierfrancesco Veltri**
DEMACS - University of Calabria, Italy
{amendola,ggreco,leone,veltri}@mat.unical.it

Abstract

A compact representation for *non-transferable utility* games founding on *answer set programming* is proposed. The representation is fully expressive, in that it can capture all games defined over a finite set of alternatives. Moreover, due to the knowledge representation capabilities of answer set programs, it can easily accommodate the definition of games within a wide range of application domains, ranging from scheduling, to routing and planning, just to name a few. The computational complexity of the proposed framework is studied, in particular, by focusing on the core as the prototypical solution concept. A system supporting the basic reasoning tasks arising therein is also made available, and results of experimental activity are discussed.

1 Introduction

Coalitional games are mathematical models which have been proposed to study payoff distribution problems among cooperative agents [von Neumann and Morgenstern, 1944]. In the classical setting where utility can be freely transferred among agents, a game can be defined as a pair $\langle N, v\rangle$, where N is a finite set of agents and where v is a function associating each *coalition* $S \subseteq N$ with the worth $v(S) \in \mathbb{R}$ that agents in S can get by collaborating with each other. A fundamental problem over $\langle N, v\rangle$ is then to single out the most desirable distributions of the total worth $v(N)$, called *solution concepts*, which can be perceived as fair and stable (see, e.g., [Osborne and Rubinstein, 1994]).

Since the nineties, coalitional games gained popularity in the artificial intelligence community where solution concepts have been reconsidered from the computational viewpoint (cf. [Deng and Papadimitriou, 1994]). In particular, moving from the observation that explicitly listing all coalitions with their worths is unfeasible over games involving many agents, significant efforts have been spent to study *compact representations* for coalitional games (see, e.g., [Chalkiadakis *et al.*, 2011; Greco *et al.*, 2011]). In a compact representation, a game $\langle N, v\rangle$ is defined via an encoding, such as a combinatorial structure, a graphical structure, or a logical theory; and, for each coalition $S \subseteq N$, the worth $v(S)$ is computed via an algorithm taking as input that structure and the coalition S.

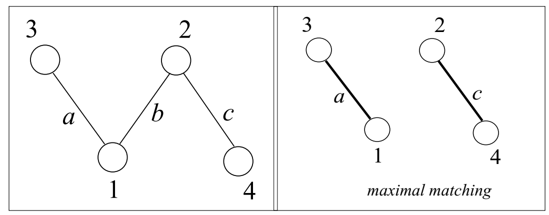

Figure 1: Illustration for Example 1 and Example 2.

Example 1. The graph on the left of Figure 1 can be a compact representation for a game over the set $N = \{1, 2, 3, 4\}$ of agents one-to-one corresponding with its nodes. For instance, for each coalition $S \subseteq N$, we might define $v(S)$ as the number of edges in the subgraph induced over the nodes in S. E.g., we have $v(\{1, 2, 3\}) = 2$ and $v(\{1, 2, 3, 4\}) = 3$. In general, as every graph on $|N|$ nodes has $O(|N|^2)$ edges, the encoding is exponentially more succinct than explicitly listing the worths associated with all $2^{|N|}$ coalitions. $\triangleleft$

Whenever utility cannot be freely transferred, the classical model for coalitional games can be generalized by associating each coalition $S \subseteq N$ with all the possible payoff distributions that are allowed to the agents in S, rather than with just one value $v(S)$. Games of this kind have been firstly studied by Aumann and Peleg [1960] and they are called coalitional games with *non-transferable utility* (short: NTU games).

Formally, for any coalition $S \subseteq N$, let $\mathbb{R}^S$ be the $|S|$-dimensional real coordinate space, whose coordinates are labeled with the members of S; in particular, given a *payoff vector* $x \in \mathbb{R}^S$, x_i denotes the component associated with the agent $i \in S$. Then, an NTU game is a pair $\langle N, V\rangle$, where N is a finite set of agents and V is a function associating each coalition $S \subseteq N$ with a set of payoff vectors $V(S) \subseteq \mathbb{R}^S$, which are also called *consequences*.

Applications of NTU games include frameworks for distributing invisible goods, for service composition, for task assignment, for coordination in wireless networks, and more generally all frameworks where incentives to cooperation are provided via payments taking values from discrete domains (see, e.g., [Greco *et al.*, 2010] and the references therein).

When looking at the specification of NTU games from the computer science perspective, however, we might note that there is no clear way to adapt over them existing encodings defined for games with transferable utility (short: TU games). In fact, for each given coalition $S \subseteq N$, $V(S)$ is not a single real number which can be computed on top of the encoding, but it is actually a set of different alternatives that are available to the agents in S. Accordingly, compact representations for NTU games must provide not only a mechanism to reason about exponentially many coalitions (as with TU games), but also a mechanism that, for each given coalition $S \subseteq N$, allows to succinctly encode the consequences in $V(S)$.

As a matter of fact, very few efforts have been spent in the literature to define compact encodings for NTU games, and a representation that naturally provides the support discussed above was missing, so far. The goal of the paper is to fill this gap, by proposing a representation for NTU games founding on *answer set programming* [Gelfond and Lifschitz, 1991].

Indeed, answer set programming is a declarative programming paradigm where problems are specified in terms of logic programs, and its key feature is that the semantics of each program P is given in terms of a set $\mathsf{AS}(P)$ of *answer sets*, each one describing a possible alterative model for P. Hence, compact representations for NTU games can be naturally based on encoding their alternatives/consequences in terms of answer sets of suitably associated programs.

In the paper, we formalize this observation and we show that every NTU game with a finite set of consequences can be represented as an *answer set game*. Then, we embark on the study of the computational complexity of such games, by considering *stratified* and *normal* programs and by focusing on the *core* as a reference solution concept [Edgeworth, 1881]. For instance, we consider the problems of checking whether a given distribution is in the core and whether the core of a game is empty. Finally, we report results of experimental activity conducted over a system prototype we have implemented to support the above reasoning tasks. Notably, no reasoner for arbitrary NTU games was available, so far.

2 Preliminaries

Answer Set Programming. Let Γ be a set of atoms. A rule r on Γ is an expression having the form

$$a \leftarrow b_1, \ldots, b_m, not\ c_1, \ldots, not\ c_n.$$

where $m + n \geq 0$, and all a, b_j and c_k are atoms from Γ.

The *head* of r is the set $H(r) = \{a\}$, and the *positive* (resp., *negative*) *body* of r is the set $B^+(r) = \{b_1, ..., b_m\}$ (resp., $B^-(r) = \{c_1, ..., c_n\}$). If $B^-(r) = B^+(r) = \emptyset$, then the arrow "$\leftarrow$" will be omitted in the notation.

A *(normal) program* P is a finite set of rules. An *interpretation* of P is any set $I \subseteq \Gamma$ of atoms. We say that I *satisfies* a rule r if $B^+(r) \subseteq I$ and $B^-(r) \cap I = \emptyset$ implies $I \cap H(r) \neq \emptyset$. If I satisfies every rule $r \in P$, then I is a *model* of P. The *reduct* of P w.r.t. an interpretation I is the program P^I such that: *(i)* for each rule $r \in P$ with $B^-(r) \cap I = \emptyset$, P^I contains the rule r' with $H(r') = H(r)$, $B^+(r') = B^+(r)$, and $B^-(r') = \emptyset$; *(ii)* no further rule is in P^I. An interpretation I is an *answer set* of P if it is a *subset-minimal* model of P^I. The set of all answer sets of P is denoted by $\mathsf{AS}(P)$.

A normal program P is *stratified* if the atoms can be partitioned into pairwise disjoint sets $S_1, ..., S_n$ such that, for each $a \in S_i$ and $b \in S_j$, if there is a rule $r \in P$ with $a \in H(r)$ and $b \in B^+(r)$ (resp., $b \in B^-(r)$), then $i \geq j$ (resp., $i > j$).

Theorem 1. *The following computational properties are known to hold (see, e.g.,* [Dantsin *et al.*, 2001]*):*

(1) *Every stratified program has a unique answer set that, moreover, can be computed in polynomial time.*

(2) *For normal programs P, checking whether an interpretation M belongs to $\mathsf{AS}(P)$ is feasible in polynomial time, while deciding whether $\mathsf{AS}(P) \neq \emptyset$ is* **NP**-*complete.*

Imputations and Cores. Let $\mathcal{G} = \langle N, V \rangle$ be an NTU game. A consequence $x \in V(N)$ is an *imputation* of $\mathcal{G}$ if the following two properties hold (see, e.g., [Peleg, 1963]):

- *Efficiency*: for each $y \in V(N)$, there is an agent $i \in N$ such that $x_i \geq y_i$; and
- *Individual Rationality*: for each agent $i \in N$, $x_i \geq \max\{y_i \mid y_i \in V(\{i\})\}$.

Intuitively, the former condition restricts imputations to those consequences where it is impossible to make any agent better off without making at least another agent worse off. Instead, the latter condition filters out all consequences where an agent receives less than it could obtain on its own, i.e., without cooperating with anyone else. In the following, the set of all imputations of $\mathcal{G}$ is denoted by $\mathsf{I}(\mathcal{G})$.

Solutions concepts are defined in the literature as imputations enjoying additional desirable properties. In the paper, we focus on the *core* of $\mathcal{G}$, denoted by $\mathsf{C}(\mathcal{G})$, which is the set of all imputations x for which there is no *objection*, i.e., pair (y, S) such that $y \in V(S)$ and $y_k > x_k$, for each $k \in S$ [Aumann, 1961]. Therefore, this outcome represents a kind of agreement amongst players, in that it is "stable" with respect to the possibility that subsets of players get an incentive to deviate from it, by forming coalitions on their own.

Fact 1. *If $\mathcal{G}=\langle N, V \rangle$ is such that $|N| \leq 2$, then $\mathsf{I}(\mathcal{G}) = \mathsf{C}(\mathcal{G})$.*

3 Formal Framework

In this section, we introduce answer set games and specialize over them the basic concepts about coalitional games.

3.1 Answer Set Games

Let N be a set of agents. An *environment* for N is a tuple $\Omega = \langle N, P, \omega \rangle$, where P is a vector of logic programs on a set Γ of atoms and ω is a vector of *weight* functions mapping interpretations to real numbers. In particular, the components of P and ω are labeled with the members of N.

We assume that $\Gamma \supseteq N$, that is, agents are transparently viewed as atoms. Moreover, for each agent $i \in N$ and rule $r \in P_i$, we require that $H(r) \cap N = \emptyset$, that is, atoms in N occur only in rule bodies. Finally, for each $i \in N$, we require that ω_i is polynomial-time computable. In the following, whenever $\omega_i(I) = \sum_{a \in I} \omega_i(\{a\})$ holds, for each interpretation I, we shall say that ω_i is *additively separable*.

Hereinafter, for each $S \subseteq N$, we define a logic program $P_S = \bigcup_{i \in S}(P_i \cup \{i.\})$ based on the logic programs P_i associated with the agents $i \in S$. In particular, when i is included in S, we add to P_S the program P_i as well as the information that i occurs in S (that is, the fact "$i.$"). This information can be used by other agents to reason about members of S. For instance, the logic program of another agent $j \neq i$ can be defined to model scenarios where j can perform some task only in coalitions where i occurs.

The notions introduced above are now exemplified over an environment where logic programs play the role of modeling *matchings* over an underlying graph—recall that a matching is a set of pairwise non-adjacent edges; thats is, there is no pair of edges sharing some node. The rationale for the chosen exemplification is that matching games constitute one of the most studied class of coalitional games, with applications in matching markets (see, e.g., [Roth and Oliveira Sotomayor, 1990; Shapley and Shubik, 1971]).

Example 2. Consider again the graph shown in Figure 1, and let us define $\Omega = \langle N, P, \omega \rangle$, with $N = \{1, 2, 3, 4\}$, as follows. Each agent $i \in N$ is equipped with a program P_i prescribing that at most one of its incident edges can be "selected". The intuition is that, for each $S \subseteq N$, answer sets of P_S will correspond to the matchings over the subgraph induced on S. Formally, we use atoms a, b, and c to identify the edges selected in the matching, and $\bar{a}$, $\bar{b}$, and $\bar{c}$ to mean that the corresponding edges are not selected. Then, we define:

$$P_1 = \left\{ \begin{array}{l} \bar{b} \leftarrow a. \\ \bar{a} \leftarrow b. \\ a \leftarrow 3, not\ \bar{a}. \\ \bar{a} \leftarrow 3, not\ a. \\ b \leftarrow 2, not\ \bar{b}. \\ \bar{b} \leftarrow 2, not\ b. \end{array} \right\}, \quad P_2 = \left\{ \begin{array}{l} \bar{c} \leftarrow b. \\ \bar{b} \leftarrow c. \\ b \leftarrow 1, not\ \bar{b}. \\ \bar{b} \leftarrow 1, not\ b. \\ c \leftarrow 4, not\ \bar{c}. \\ \bar{c} \leftarrow 4, not\ c. \end{array} \right\},$$

$$P_3 = \{a \leftarrow 1, not\ \bar{a}.\ \ \bar{a} \leftarrow 1, not\ a\},$$
$$P_4 = \{c \leftarrow 2, not\ \bar{c}.\ \ \bar{c} \leftarrow 2, not\ c.\}.$$

Note that the interpretation $M = \{1, 2, 3, 4, a, \bar{b}, c\}$ belongs to $\mathsf{AS}(P_N) = \mathsf{AS}(\bigcup_{i \in \{1,2,3,4\}}(P_i \cup \{i.\}))$, and it encodes a maximal *matching* over the graph—see Figure 1.

Finally, for each agent/node $i \in N$, ω_i is the additively separable function mapping each atom to 0, but the edges incident to i, which are mapped to 1. Hence, for the answer set M, we have $\omega_i(M) = 1$, for each $i \in \{1, 2, 3, 4\}$. ◁

Environments are now used to induce, in a natural way, an NTU game over the agents in N. Formally, the *answer set game* associated with Ω is the game $\mathcal{G}_\Omega = \langle N, V_\Omega \rangle$ where, for each $S \subseteq N$, the set $V_\Omega(S)$ is defined as follows:

$$V_\Omega(S) = \{x \in \mathbb{R}^S \mid \ \exists M \in \mathsf{AS}(P_S) \text{ such that } x_i = \omega_i(M), \forall i \in N\}.$$

Example 3. In Example 2, consider the coalition $S = \{2, 4\}$. We have $V_\Omega(S) = \{\alpha, \beta\}$ with $\alpha_2 = \alpha_4 = 1$ (corresponding to the matching where the edge c is selected) and $\beta_2 = \beta_4 = 0$ (corresponding to the case where no edge is selected). ◁

It is easy to see that any *finite* NTU game $\langle N, V \rangle$, i.e., such that $V(S)$ is a finite set of outcomes, for each $S \subseteq N$, can be defined by using a suitable environment.[1]

Theorem 2. *For each finite* NTU *game* $\langle N, V \rangle$, *there is an environment* Ω *over* N *such that* $V_\Omega(S) = V(S)$, $\forall S \subseteq N$.

[1]See `http://ntu2dlv.altervista.org/`, for details and missing proofs.

In fact, the environments in the above result founds on programs whose rules are associated with all possible coalitions $S \subseteq N$ and with all possible outcomes in $V(S)$. But, more succinct encodings can be obtained in some cases. For instance, by generalizing Example 2 to arbitrary graphs, it is easily seen that all (possibly exponentially-many) matchings can be encoded via programs with polynomially-many rules, only. For instance, for a clique over N, the number of maximal matching is exponential w.r.t. $|N|$, but the number of rules in each program P_i is linearly bounded by the number of such nodes. For a further example of a succinct encoding, we can consider an environment where, for each $i \in N$, we have $P_i = \{a_i \leftarrow i, not\ \bar{a}_i.\} \cup \{\bar{a}_i \leftarrow i, not\ a_i.\}$ and $\omega_i(\{a_i\}) = 1$, with all other atoms mapped to 0. Then, we can check that $|\mathsf{AS}(P_N)| = |V_\Omega(N)| = 2^{|N|}$ holds.

More importantly, the modeling capabilities of answer set programming make the setting expressive enough to deal with a number of real-world applications, for instance, to encode games defined over combinatorial structures [Bilbao, 2012]. Indeed, for each $i \in N$, P_i might encode different combinatorial problems, ranging from scheduling, to routing and planning, just to name a few (see, e.g., [Grasso *et al.*, 2013]).

Example 4. Consider a setting where, in order to perform a task t, a set $\mathcal{S}$ of skills are required, and where each agent $i \in N$ has some skills $s_1, ..., s_{h_i}$ taken from $\mathcal{S}$. Consider a stratified logic program P modeling the conditions under which t is executed. In particular, if $T \subseteq \mathcal{S}$ is a set of available skills, then the unique answer set M of $P \cup \bigcup_{s \in T}\{s.\}$ is such that: $t \in M$, if and only if, $T = \mathcal{S}$. Note that further specific application-oriented constraints can be defined in P.

Now, for each agent $i \in N$, consider the program $P_i = P \cup \{s_1.\} \cup \cdots \cup \{s_{h_i}.\}$ and the weight function ω_i such that $\omega_i(M) = 1$ (resp., $\omega_i(M) = 1$) if $t \in M$ (resp., $t \notin M$). Then, the game $\mathcal{G}_\Omega = \langle N, V_\Omega \rangle$ is such that, for each $S \subseteq N$, $V_\Omega(S) = \{x\}$ where each component of x is 1 (resp., 0) if agents in S have (resp., do not have) the skills to perform t.

Simple extensions of the exemplification include the execution of more than one task, and the definition of a monetary reward that is specific for each task being executed and for the specific skills provided by each agent. ◁

3.2 Solution Concepts

Answer sets games $\mathcal{G}_\Omega$ are standard NTU games and, therefore, we can apply on them well-known notions and solution concepts from coalitional game theory. In fact, such notions and concepts can be recast in terms of the underlying vector of logic programs P, as we discuss below.

Note first that, if x is an imputation in $\mathsf{I}(\mathcal{G}_\Omega)$, then $x \in V_\Omega(N)$ holds and, by definition of of V_Ω, there is an answer set $M \in \mathsf{AS}(P_N)$ such that $x_i = \omega_i(M)$, for each $i \in N$. This is called an *imputation answer set* of P_N w.r.t. $\mathcal{G}_\Omega$, and a useful characterization for it is next provided.

Fact 2. *An interpretation* $M \in \mathsf{AS}(P_N)$ *is an imputation answer set if, and only if, the following properties hold:*

- *"Efficiency": for each* $M' \in \mathsf{AS}(P_N)$, *there is an agent* $i \in N$ *such that* $\omega_i(M) \geq \omega_i(M')$; *and*
- *"Individual Rationality": for each agent* $i \in N$, $\omega_i(M) \geq max\{\omega_i(M') \mid M' \in \mathsf{AS}(P_{\{i\}})\}$.

Example 5. Recall from Example 2 that $M = \{1, 2, 3, 4, a, \bar{b}, c\} \in \mathsf{AS}(P_N)$ and $\omega_1(M) = 1$. This is the best possible value for ω_1; hence, M is efficient. Moreover, $\omega_i(M) = 1 \geq \max\{\omega_i(M') \mid M' \in \mathsf{AS}(P_{\{i\}})\}$, for each $i \in N$. Indeed, $P_{\{i\}}$ has one answer set $\{i.\}$ and $\omega_i(\{i.\}) = 0$. So, M is also individually rational. Thus, it is an imputation answer set. $\triangleleft$

Let $\mathsf{IAS}(P, \mathcal{G}_\Omega)$ denote the set of all imputation answer sets. An interesting property is that, as far as the non-emptiness of $\mathsf{IAS}(P, \mathcal{G}_\Omega)$ is concerned, efficiency in Fact 2 is immaterial.

Theorem 3. *If there is an answer set of P_N that satisfies the individual rationality property, then $\mathsf{IAS}(P, \mathcal{G}_\Omega) \neq \emptyset$.*

In addition to the elements in $\mathsf{IAS}(P, \mathcal{G}_\Omega)$, the paper will also study core answer sets. Formally, an interpretation $M \in \mathsf{AS}(P_N)$ is a *core answer set* of P_N w.r.t. $\mathcal{G}_\Omega$ if $\omega(M) \in \mathsf{C}(\mathcal{G}_\Omega)$. Let $\mathsf{CAS}(P, \mathcal{G}_\Omega)$ be the set of all core answer sets. In order to characterize these elements, given an interpretation M and a coalition $S \subseteq N$, we say that M is *S-feasible* if $M \in \mathsf{AS}(P_S)$. Then, the following can be derived.

Fact 3. *An interpretation $M \in \mathsf{AS}(P_N)$ is a core answer set of P_N w.r.t. $\mathcal{G}_\Omega$ if, and only if, the following properties hold:*

- *M is an imputation answer set (i.e., $M \in \mathsf{IAS}(P, \mathcal{G}_\Omega)$);*
- *"Absence of Objections": there is no S-feasible interpretation M' such that $\omega_k(M') > \omega_k(M)$, $\forall k \in S$.*

Example 6. In our example, $M \in \mathsf{AS}(P_N)$ is a core answer set. This is witnessed by Fact 3 and by observing that, for each coalition $S \subseteq N$ and for each $M' \in \mathsf{AS}(P_S)$, there is always some agent $k \in S$ such that $\omega_k(M') \leq \omega_k(M)$.

Instead, the set $\{1, 2, 3, 4, a, \bar{b}, \bar{c}\} \in \mathsf{IAS}(P, \mathcal{G}_\Omega)$ is not a core answer set. Indeed, by Example 3, we know that coalition $\{2, 4\}$ can object. In fact, for the interpretation $\{2, 4, \bar{b}, c\} \in \mathsf{AS}(P_{\{2,4\}})$, we can derive that $\omega_2(\{2, 4, \bar{b}, c\}) = \omega_4(\{2, 4, \bar{b}, c\}) = 1$, while we have $\omega_2(\{1, 2, 3, 4, a, \bar{b}, \bar{c}\}) = \omega_4(\{1, 2, 3, 4, a, \bar{b}, \bar{c}\}) = 0$.

More generally, it can be checked that an answer set in $\mathsf{AS}(P_N)$ is a core answer set if, and only if, it corresponds to a maximal matching over the underlying graph. $\triangleleft$

4 Computational Complexity Aspects

In this section, we study the computational complexity of the following reasoning tasks, all of them receiving as input an environment Ω and being defined over the game $\mathcal{G}_\Omega$:

I-CHECK: Given a vector x, is $x \in \mathsf{I}(\mathcal{G}_\Omega)$?
C-CHECK: Given a vector x, is $x \in \mathsf{C}(\mathcal{G}_\Omega)$ hold?
I-NONEMPTY: Does $\mathsf{I}(\mathcal{G}_\Omega) \neq \emptyset$ hold?
C-NONEMPTY: Does $\mathsf{C}(\mathcal{G}_\Omega) \neq \emptyset$ hold?

In addition, we study the following problems[2] where solution concepts for $\mathcal{G}_\Omega$ are recast in terms of answer sets:

IAS-CHECK: Given an interpretation M, is $M \in \mathsf{IAS}(P, \mathcal{G}_\Omega)$?
CAS-CHECK: Given an interpretation M, is $M \in \mathsf{CAS}(P, \mathcal{G}_\Omega)$?
IAS-NONEMPTY: Does $\mathsf{IAS}(P, \mathcal{G}_\Omega) \neq \emptyset$ hold?
CAS-NONEMPTY: Does $\mathsf{CAS}(P, \mathcal{G}_\Omega) \neq \emptyset$ hold?

[2]Note that IAS-NONEMPTY and CAS-NONEMPTY are equivalent to I-NONEMPTY and C-NONEMPTY. Indeed, $\mathsf{I}(\mathcal{G}_\Omega) \neq \emptyset$ (resp., $\mathsf{C}(\mathcal{G}_\Omega) \neq \emptyset$) iff $\mathsf{IAS}(P, \mathcal{G}_\Omega) \neq \emptyset$ (resp., $\mathsf{CAS}(P, \mathcal{G}_\Omega) \neq \emptyset$).

Problem	Restriction	Normal	Stratified
IAS-CHECK[h]	$h \in \mathbb{N} \cup \{\infty\}$	**coNP**-c	in **P**
I-CHECK[h]	$h = 1$	**coNP**-c	in **P**
	$h \in \mathbb{N} \setminus \{1\} \cup \{\infty\}$	$\mathbf{D^p}$-c	in **P**
IAS-NONEMPTY[h]	$h = 1$	**NP**-c	in **P**
$\equiv$ I-NONEMPTY[h]	$h \in \mathbb{N} \setminus \{1\} \cup \{\infty\}$	$\mathbf{\Delta_2^p}$-c	in **P**
CAS-CHECK[h]	$h \in \mathbb{N}$	**coNP**-c	in **P**
	$h = \infty$	**coNP**-c	**coNP**-c
C-CHECK[h]	$h = 1$	**coNP**-c	in **P**
	$h \in \mathbb{N} \setminus \{1\}$	$\mathbf{D^p}$-c	in **P**
	$h = \infty$	$\mathbf{D^p}$-c	**coNP**-c
CAS-NONEMPTY[h]	$h = 1$	**NP**-c	in **P**
$\equiv$ C-NONEMPTY[h]	$h = 2$	$\mathbf{\Delta_2^p}$-c	in **P**
	$h \in \mathbb{N} \setminus \{1, 2\}$	$\mathbf{\Sigma_2^p}$-c	in **P**
	$h = \infty$	$\mathbf{\Sigma_2^p}$-c	**coNP**-c

Table 1: Summary of complexity results. Hardness results hold over additively separable weight functions.

A summary of our results is reported in Table 1, by distinguishing *normal* and *stratified environments* $\Omega = \langle N, P, \omega \rangle$, that is, when P_N (hence, each program P_S with $S \subseteq N$) is normal and stratified, respectively. Note that, for each problem, say P, we report the complexity when the input is restricted over environments such that $|N| = h$, where $h \in \mathbb{N}$ is a given fixed natural number, by denoting the problem as P[h]. For notational uniformity, the original problem P, without any bound on N, is also denoted as P[∞].

4.1 Overview of the Proofs

We now discuss some representative proofs for the results reported in Table 1. In particular, we focus on intractability results only, and we start with stratified environments.

Theorem 4. CAS-CHECK[∞] *is* $\mathbf{coNP}$*-complete on stratified environments.*

Proof sketch. Membership is routine. For the hardness, let $\Phi = c_1 \wedge \ldots \wedge c_m$ be a Boolean formula in conjunctive normal form over the variables in $\{X_1, \ldots, X_n\}$. For a literal X_i (resp., $\neg X_i$) in a clause c_j, let $\pi(X_i) = X_i$ (resp., $\pi(\neg X_i) = not\ X_i$). Deciding the satisfiability of such formulas is a well-known $\mathbf{NP}$-hard problem.

Based on Φ, we build in polynomial time an environment $\Omega = \langle N, P, \omega \rangle$ such that $N = \{1, \ldots, n\}$, $P_i = \{X_i \leftarrow i.\} \cup Q$, for each $i \in N$, where

$$Q = \begin{cases} sat_j \leftarrow \pi(\ell_i). & \forall j \in \{1, \ldots, m\} \text{ and } \ell_i \in c_j \\ sat \leftarrow sat_1, \ldots, sat_m. & \end{cases}$$

and where, for each $i \in N$, $\omega_i(I) = 1$ (resp., $\omega_i(I) = 0$) if $sat \in I$ (resp., $sat \notin I$). W.l.o.g., assume that the truth assignment $\bar{\sigma}$ where all variables evaluate true is not satisfying, and consider the answer set $M = \{sat_j \mid c_j \text{ evaluates true in } \bar{\sigma}\} \cup \bigcup_{i \in N} \{i.\}$ of P_N. Note that M is the unique answer set of P_N, as the program is stratified. The result then follows by checking that $M \notin \mathsf{CAS}(P, \mathcal{G}_\Omega)$ if, and only if, Φ is satisfiable. $\square$

Considering normal environments, completeness results for the polynomial-time closure of $\mathbf{NP}$ emerge.

Theorem 5. *$\forall h \in \mathbb{N} \setminus \{1\} \cup \{\infty\}$, IAS-NONEMPTY$[h]$ is $\mathbf{\Delta_2^p}$-complete on normal environments.*

Proof Sketch. For the membership part, by Theorem 3, we have to check whether there exists an answer set satisfying the individual rationality property. To check this property, we can first compute, for each $i \in N$, the value $m_i = \max\{\omega_i(M') \mid M' \in \mathsf{AS}(P_{\{i\}})\}$. This can be done by means of a binary search within the range for the possible values for m_i. At each step, we just ask whether there is an answer set $M' \in \mathsf{AS}(P_{\{i\}})$ with $\omega_i(M')$ exceeding the current given threshold. Each step is hence feasible in $\mathbf{NP}$, so m_i can be computed in $\mathbf{\Delta_2^p}$. Eventually, we can guess an interpretation M by then checking that $M \in \mathsf{AS}(P_N)$ and that, for each $i \in N$, $\omega_i(M) \geq m_i$. The whole computation is in $\mathbf{\Delta_2^p}$.

For the hardness, we exhibit a reduction from the $\mathbf{\Delta_2^p}$-complete OddLexMaxSAT problem [Wagner, 1987]. We are given a Boolean formula $\Phi = c_1 \wedge ... \wedge c_m$ in conjunctive normal form over the variables in $\{X_1, ..., X_n\}$. Such variables are lexicographically ordered ($X_1 < \cdots < X_n$), and we ask whether X_1 evaluates true in the lexicographically maximum satisfying assignment—w.l.o.g., Φ is satisfiable by a satisfying assignment where X_1 evaluates true. In the following, for a literal X_i (resp., $\neg X_i$), let $\pi(X_i) = X_i$ (resp., $\pi(\neg X_i) = not\ X_i$). Based on Φ, we build in polynomial time an environment $\Omega = \langle N, P, \omega\rangle$ over two agents, i.e., $N = \{1, 2\}$, and where $P_1 = P_2$ consists of the following rules:

$$\begin{cases} X_i \leftarrow not\ \bar{X}_i. & \forall i \in \{1, ..., n\} \\ \bar{X}_i \leftarrow not\ X_i. & \forall i \in \{1, ..., n\} \\ sat_j \leftarrow \pi(\ell_i). & \forall j \in \{1, ..., m\} \text{ and } \ell_i \in c_j \\ sat \leftarrow sat_1, ..., sat_m. & \\ \leftarrow not\ sat. & \end{cases}$$

Finally, we consider two additively separable weight functions, ω_1 and ω_2, given by

$$\omega_1(\{a\}) = \begin{cases} 2^i & \text{if } a = X_i, \\ 0 & \text{otherwise.} \end{cases} \quad \omega_2(\{a\}) = \begin{cases} 1 & \text{if } a = X_1, \\ 0 & \text{otherwise.} \end{cases}$$

We claim that Φ is a "yes" instance of OddLexMaxSAT if, and only if, $\mathsf{IAS}(P, \mathcal{G}_\Omega) \neq \emptyset$—the reduction can be extended to the case where $|N| > 2$, by setting $\omega_i(I) = 0$, for each agent $i \in N \setminus \{1, 2\}$ and interpretation I. In the following, for each assignment σ, let M_σ be the (univocally determined) answer set of P_N with $M_\sigma \supseteq \{X_i \mid X_i \text{ evaluates true in } \sigma\}$.

Assume that X_1 evaluates false in the lexicographically maximum satisfying assignment σ. Recall that Φ is satisfiable by a satisfying assignment, say $\hat{\sigma}$, where X_1 evaluates true. Consider the interpretation $M_{\hat{\sigma}}$. Assume, for the sake of contradiction, that M is an answer set in $\mathsf{IAS}(P, \mathcal{G}_\Omega)$. Note that $\omega_2(M) \geq \omega_2(M_{\hat{\sigma}}) = 1$ must hold, since M is individually rational. That is, X_1 must occur in M. Consider then the assignment $\bar{\sigma}$ such that X_i evaluates true if, and only if, $X_i \in M$. Note that $M = M_{\bar{\sigma}}$. However, $\omega_1(M_{\bar{\sigma}}) < \omega_1(M_\sigma)$, because σ is the lexicographically maximum satisfying assignment and $\bar{\sigma} \neq \sigma$, as X_1 evaluates true (resp., false) in $\bar{\sigma}$ (resp., σ). So, M is not individually rational. This leads to a contradiction.

Eventually, assume that X_1 is true in the lexicographically maximum satisfying assignment σ^*. Note that $M_{\sigma^*} \supseteq \{X_1\}$, $\omega_1(M_{\sigma^*}) = \sum_{X_i \mid X_i \text{ evaluates true in } \sigma^*} 2^i$ and $\omega_2(M_{\sigma^*}) = 1$. Therefore, it can be checked that M_{σ^*} is efficient and individually rational. □

We leave the section by exhibiting a completeness result for the second level of the polynomial hierarchy.

Theorem 6. *$\forall h \in \mathbb{N} \setminus \{1, 2\} \cup \{\infty\}$, CAS-NONEMPTY$[h]$ is $\mathbf{\Sigma_2^p}$-complete on normal environments.*

Proof Sketch. In order to solve CAS-NONEMPTY, we can start by guessing an interpretation M. This task is clearly feasible in $\mathbf{NP}$. Subsequently, we have to check whether M is a core answer set, which is feasible in $\mathbf{coNP}$. Therefore, CAS-NONEMPTY is in $\mathbf{\Sigma_2^p}$. For the hardness, we exhibit a reduction from the prototypical $\mathbf{\Sigma_2^p}$-complete problem of deciding the validity of a quantified Boolean formula Ψ having the form $\exists X_1, ..., X_n \forall Y_1, ..., Y_m \Phi$. Without loss of generality, we assume that Φ is in disjunctive normal form, that is, $\Phi = D_1 \vee ... \vee D_r$ where, for each $k \in \{1, ..., r\}$, $D_k = l_{k1} \wedge l_{k2} \wedge l_{k3}$ is a conjunction of literals.

Consider the following normal program:

$$P_\Phi = \begin{cases} X_i \leftarrow not\ \bar{X}_i. & \forall i \in \{1, ..., n\} \\ \bar{X}_i \leftarrow\ not\ X_i. & \forall i \in \{1, ..., n\} \\ Y_j \leftarrow not\ \bar{Y}_j. & \forall j \in \{1, ..., m\} \\ \bar{Y}_j \leftarrow not\ Y_j. & \forall j \in \{1, ..., m\} \\ sat \leftarrow \pi(l_{k1}), \pi(l_{k2}), \pi(l_{k3}). & \forall k \in \{1, ..., r\} \\ \bar{sat} \leftarrow not\ sat. & \end{cases}$$

where, for each $s \in \{1, 2, 3\}$, $\pi(l_{ks}) = \begin{cases} \bar{X}_i & \text{if } l_{ks} = \neg X_i; \\ \bar{Y}_j & \text{if } l_{ks} = \neg Y_j; \\ l_{ks} & \text{otherwise.} \end{cases}$

Let $C = \{c_{12}, c_{13}, c_{23}, c_{123}\}$. Moreover, consider the following normal logic program

$$Q = \begin{cases} c_{12} \leftarrow 1, 2, not\ 3. \\ c_{23} \leftarrow 2, 3, not\ 1. \\ c_{13} \leftarrow 1, 3, not\ 2. \\ c_{123} \leftarrow 1, 2, 3. \end{cases}$$

Now, based on Ψ and given P and C, we build the answer set game $\mathcal{G}_\Omega$, where $\Omega = \langle N, P, \omega\rangle$ is defined as follows.

First, we define $N = \{1, 2, 3\}$. Then, we let $P_1 = P_2 = P_3 = P_\Phi \cup Q$. And, finally, the weight functions are the additively separable functions given by

$$\omega_1(\{a\}) = \begin{cases} 1 & \text{if } a = \bar{sat}, \\ 2^i & \text{if } a = X_i, \\ 2^{n+1} & \text{if } a \in C, \\ 0 & \text{otherwise.} \end{cases}$$

$$\omega_2(\{a\}) = \begin{cases} 1 & \text{if } a = \bar{sat}, \\ 2^i & \text{if } a = \bar{X}_i, \\ 2^{n+1} & \text{if } a \in C, \\ 0 & \text{otherwise.} \end{cases} \quad \omega_3(\{a\}) = \begin{cases} 1 & \text{if } a = sat, \\ 0 & \text{otherwise.} \end{cases}$$

Let M be an answer set in $\mathsf{AS}(P_N)$ and observe that $M \supseteq \{1, 2, 3, c_{123}\}$. Therefore, we have:

- $\omega_1(M) = 2^{n+1} + \sum_{X_i \in M} 2^i + 1 - \omega_3(M)$;
- $\omega_2(M) = 2^{n+1} + \sum_{\bar{X}_i \in M} 2^i + 1 - \omega_3(M)$.

Eventually, it can be checked that Ψ is valid if, and only if, there is a core answer set.

(only-if part) Assume that σ_X is a truth assignment over the variables in $\{X_1, ..., X_n\}$ witnessing the validity of Ψ. Let σ be a truth assignment for Φ whose restriction on $\{X_1, ..., X_n\}$ coincides with σ_X, and note that σ is satisfying. Consider the answer set $M_\sigma \in \mathsf{AS}(P_N)$ such that $M_\sigma \supseteq \{X_i \mid X_i$ evaluates true in $\sigma\}$. Note that M_σ is univocally defined and $sat \in M_\sigma$. Now, we show that M is individually rational. Indeed, consider an answer set $M_1 \in \mathsf{AS}(P_{\{1\}})$, hence, such that $C \cap M_1 = \emptyset$. Then, $\omega_1(M_1) \leq 2^{n+1} \leq \omega_1(M_\sigma)$. Similarly, for any answer set $M_2 \in \mathsf{AS}(P_{\{2\}})$, we have $\omega_2(M_2) \leq 2^{n+1} \leq \omega_2(M_\sigma)$. Eventually, for any answer set $M_3 \in \mathsf{AS}(P_{\{3\}})$, we have $\omega_3(M_3) \leq 1 = \omega_3(M_\sigma)$. By similar arguments, we can show that M_σ is efficient and satisfies the objection property.

(if part) Assume that Ψ is not valid, and let $M \in \mathsf{AS}(P_N)$. In the case where $sat \in M$, we can consider any answer set M' of $P_{\{1,2\}}$, such that $M' \cap \{X_1, ..., X_n\} = M \cap \{X_1, ..., X_n\}$, and $\bar{sat} \in M'$. As Ψ is not valid, one answer set of this kind clearly exists. Moreover, we can note that $\omega_1(M') = \omega_1(M) + 1$ and $\omega_2(M') = \omega_2(M) + 1$. Therefore, M' is $\{1, 2\}$-feasible and such that $\omega_k(M') > \omega_k(M)$, for each $k \in \{1, 2\}$. Hence, M is not a core answer set. Consider, then, the case where $\bar{sat} \in M$. W.l.o.g., assume there is an answer set M' of $P_{\{2,3\}}$ such that $M' \cap \{X_1, ..., X_n\} = M \cap \{X_1, ..., X_n\}$ and $sat \in M'$. Then, $\omega_3(M') = 1 > \omega_3(M) = 0$ and $\omega_2(M') = \omega_2(M) + 1$. Again, this shows that M is not a core answer set. □

5 Implementation and Experiments

To support the reasoning tasks discussed in the paper, a system prototype, named `ntu2DLV`, has been implemented on top of the well-known answer set programming DLV reasoner [Leone *et al.*, 2006]. In the following, we discuss the architecture of the system and results of some experimental activities we have conducted on it.

5.1 System Architecture

Let $\Omega = \langle N, P, \omega \rangle$ be a given environment. In order to use `ntu2DLV`, each program $P_i \in P$ has to be stored in a file named "agent[*i*].dl". Analogously, each weight function $\omega_i \in \omega$ has to be stored in a file named "weight[*i*].txt". We support additively separable functions, so that each line of the file consists of a pattern "$[a] \rightarrow [v]$", where v is the value associated to the atom a. All files have to be stored in a folder and the system can be then invoked as: `./ntu2DLV` `folderName -reasoning=[imputations|core]`, where the option selects the desired reasoning task, that is, computing imputations or core answer sets, respectively.[3]

[3]The system prototype and further notes on its usage are also available at `http://ntu2dlv.altervista.org/`.

IAS	3	6	9	12	15
normal	0,085	0,164	0,731	11,711	625,770
uniform	0,083	0,148	0,677	4,349	173,067
power-law	0,082	0,121	0,245	0,602	1,653

Table 2: Execution time (sec) for matching games, on random graphs for various degree distributions.

The system has been implemented in Java. Upon startup, it analyzes the input folder, by parsing the files and creating the associated data structures encoding $\mathcal{G}_\Omega$. These structures are taken as input by a reasoning module, which interacts with the DLV system via the *DLVWrapper* library [Ricca, 2003]:

"*-reasoning=imputations*": When the focus is on imputation answer sets, DLV is invoked to compute $\mathsf{AS}(P_{\{i\}})$, for each $i \in N$, and to compute $\mathsf{AS}(P_N)$. The system then calculates the values $\max\{\omega_i(M') \mid M' \in \mathsf{AS}(P_{\{i\}})\}$ to discard the elements of $\mathsf{AS}(P_N)$ that are not individually rational. For each remaining answer set M, the system checks whether there is an agent i such that $\omega_i(M) = \max\{\omega_i(M) \mid M \in \mathsf{AS}(P_N)\}$. In the affirmative case, M is an imputation answer set and is returned as output.

"*-reasoning=core*": Whenever the system is asked to return core answer sets, DLV is additionally invoked to compute $\mathsf{AS}(P_S)$, for each coalition $S \subset N$ with $|S| > 1$. For each $M \in \mathsf{AS}(P_S)$, the elements $I \in \mathsf{IAS}(P, \mathcal{G}_\Omega)$ such that $\omega_i(M) > \omega_i(I)$, for each $i \in S$, are discarded. The imputation answer sets that survive to the filtering process are promoted to core answer sets and returned as output by the system.

5.2 Experiments and Results

Experimental activity has been conducted to assess the efficiency and the effectiveness of `ntu2DLV`. Tests have been carried out on an Intel Core i7-4710HQ, 2.50 GHz, with 16 Gb Ram, running Linux Operating System; for each test we allowed a maximum running time of 1800 seconds.

In a first series of experiments, we considered the setting discussed in Example 2, where a graph G is given and where Ω encodes a matching game. The system has been tested on different data sets of randomly-generated graphs, for *normal*, *uniform* and *power-law* distributions of node degrees.[4] For each given distribution and desired number of nodes, 3 graphs have been generated and average times are discussed. In particular, Table 2 reports the time taken by `ntu2DLV` to compute an imputation answer set at the growing of the number of the nodes. Note that with the power law distribution, the scaling of the system is quite effective. Indeed, for imputation answer sets, the number of invocations to DLV is always linear w.r.t. the number of agents. However, the number of rules of each program P_i is determined by the number of neighbors of agent i (see Example 2) so that, with the power-law distribution where most agents have only a few neighbors, reasoning

[4]The data generator along with a user guide can be downloaded from the system web-site.

	3	6	9	12	15
CAS	0,091	0,298	6,795	353,850	≥1800
2-CAS	0,092	0,154	0,717	11,670	635,077
3-CAS	-	0,175	0,942	12,212	644,897
4-CAS	-	0,204	3,553	13,550	675,085

Table 3: Execution time (sec) for matching games, on random graphs with normal degree distribution.

on their associated programs is easier—DLV reasoning time tend to be exponential w.r.t. the size of the program.

Table 3 reports the (average) time taken by `ntu2DLV` to compute a core answer set, by focusing on normal edge distributions. In this case, DLV has to be invoked $2^{|N|}$ times in order to check whether there is some coalition that can object. Therefore, the exponential dependency w.r.t. the number of nodes is unavoidable, and the system does not scale to large scenarios. Pragmatically, as often done in these contexts, one might want to focus on certain coalitions only, as to restrict the search space. For instance, the table reports execution times when coalitions of size 2, 3, and 4 are considered.

	3	6	9	12	15
IAS	0,098	0,158	0,300	0,536	1,213
CAS	0,110	0,411	6,110	118,295	≥1800
2-CAS	0,107	0,186	0,358	0,711	1,455
3-CAS	-	0,267	0,590	1,386	3,121
4-CAS	-	0,296	1,134	4,009	12,124

Table 4: Execution time (sec) for independent-set games, on random graphs with normal degree distributions.

For a further series of experiments, we focused on the flexibility of answer set systems to define and solve combinatorial problems (in their turn, inducing NTU games). In particular, we considered the same scenario as in Example 2, but defining programs P_S whose answer sets correspond to *independent sets* over the subgraph induced on S—the encoding is standard and is omitted. In this case, on the specific family of graphs we have considered, it emerged that fewer answer sets are generated compared to matching games. Accordingly, it comes with no surprise that DLV reasoning times are better than those discussed so far. Details are reported in Table 4.

6 Discussion and Conclusion

A compact representation for non-transferable utility games has been proposed which founds on the use of answer set programming. The computational complexity of the framework has been studied, and a system supporting the basic reasoning tasks arising therein has been made available.

Our contribution fills a gap in the literature, where no general mechanism to compactly specify NTU games was proposed, so far. An avenue of research that is related to our work defines NTU games by equipping underlying TU games with some application-oriented *constraints* on the possible worth distributions—see, e.g., [Aumann and Dreze, 1974; Byford, 2007; Jiang and Baras, 2007] and the general framework of Greco *et al.* [2010]. Actually, in that framework, constraints are used to define the set $V(N)$, only. For instance, to induce an NTU game where $V(\{1,2,3\}) = \{(1,1,1)\}$, we have to specify that x_1, x_2, and x_3 are positive integers, and that $x_1 + x_2 + x_3 = 3$. Then, for each coalition S, consequences in $V(S)$ are implicitly induced in [Greco *et al.*, 2010] by projecting $V(N)$ over S. For instance, we have $V(\{1,2\}) = \{(1,1)\}$ and we have no flexibility to define, e.g., $V(\{1,2\}) = \{(1,1),(2,0)\}$. By moving from numerical constraints to logical theories, answer set games are more flexible (in fact, fully expressive). Indeed, the input programs P_i, for each $i \in N$, can be defined in a way that the program P_S associated with coalition S encodes any desired set $V(S)$.

Natural avenues of further research include the study of other solution concepts over answer set games, for instance taking into account scenarios with restricted agent interactions (see, e.g., [Chalkiadakis *et al.*, 2016]), and the definition of optimization strategies for the system prototype.

Acknowledgments

The authors are supported by the Italian Ministry of University and Research under PON project "Ba2Know (Business Analytics to Know) Service Innovation - LAB", No. PON03PE_00001_1. Gianluigi Greco's work was also supported by a Kurt Gödel Research Fellowship, awarded by the Kurt Gödel Society.

References

[Aumann and Dreze, 1974] Robert J. Aumann and Jacques H. Dreze. Cooperative games with coalition structures. *International Journal of Game Theory*, 3(4):217–237, 1974.

[Aumann and Peleg, 1960] Robert J. Aumann and Bezalel Peleg. Von neumann-morgenstern solutions to cooperative games without side payments. *Bulletin of the American Mathematical Society*, 66(3):173–179, 1960.

[Aumann, 1961] Robert J. Aumann. The core of a cooperative game without side payments. *Transactions of the American Mathematical Society*, 98:599–612, 1961.

[Bilbao, 2012] Jesús Mario Bilbao. *Cooperative games on combinatorial structures*, volume 26. Springer Science & Business Media, 2012.

[Byford, 2007] Martin Charles Byford. A constrained coalitional approach to price formation. In *North American Summer Meetings of the Econometric Society*, Durham, NC, USA, June 21–24 2007.

[Chalkiadakis *et al.*, 2011] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168, 2011.

[Chalkiadakis *et al.*, 2016] Georgios Chalkiadakis, Gianluigi Greco, and Evangelos Markakis. Characteristic function games with restricted agent interactions: Core-stability and coalition structures. *Artificial Intelligence*, 232:76–113, 2016.

[Dantsin *et al.*, 2001] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.

[Deng and Papadimitriou, 1994] Xiaotie Deng and Christos H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.

[Edgeworth, 1881] Francis Ysidro Edgeworth. *Mathematical Psychics: An essay on the mathematics to the moral sciences*. C. Kegan Paul & Co., London, 1881.

[Gelfond and Lifschitz, 1991] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.

[Grasso *et al.*, 2013] Giovanni Grasso, Nicola Leone, and Francesco Ricca. Answer set programming: Language, applications and development tools. In *Proc. of RR'13*, pages 19–34, 2013.

[Greco *et al.*, 2010] Gianluigi Greco, Enrico Malizia, Luigi Palopoli, and Francesco Scarcello. Non-transferable utility coalitional games via mixed-integer linear constraints. *Journal of Artificial Intelligence Research*, 38:633–685, 2010.

[Greco *et al.*, 2011] Gianluigi Greco, Enrico Malizia, Luigi Palopoli, and Francesco Scarcello. On the complexity of core, kernel, and bargaining set. *Artificial Intelligence*, 175(12-13):1877–1910, 2011.

[Jiang and Baras, 2007] Tao Jiang and John S. Baras. Fundamental tradeoffs and constrained coalitional games in autonomic wireless networks. In *Proc. of WiOpt'07*, pages 1–8, Limassol, Cyprus, 2007.

[Leone *et al.*, 2006] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV System for Knowledge Representation and Reasoning. *ACM Transactions on Computational Logic*, 7:499–562, 2006.

[Osborne and Rubinstein, 1994] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, Cambridge, MA, USA, July 1994.

[Peleg, 1963] Bezalel Peleg. Bargaining sets of cooperative games without side payments. *Israel Journal of Mathematics*, 1(4):197–200, 1963.

[Ricca, 2003] Francesco Ricca. The dlv java wrapper. In *Proc. of CILC'03*, pages 263–274, 2003.

[Roth and Oliveira Sotomayor, 1990] Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-sided matching : a study in game-theoretic modeling and analysis*. Econometric society monographs. Cambridge University Press, 1990.

[Shapley and Shubik, 1971] Lloyd S. Shapley and Martin Shubik. The assignment game i: The core. *International Journal of Game Theory*, 1(1):111–130, 1971.

[von Neumann and Morgenstern, 1944] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior (1st edition)*. Princeton University Press, Princeton, NJ, 1944.

[Wagner, 1987] Klaus W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theor. Comput. Sci.*, 51:53–80, 1987.