Transferable Utility Planning Games*

Ronen I. Brafman
Computer Science Dept.
Ben-Gurion Univ., Israel
brafman@cs.bgu.ac.il

Carmel Domshlak and **Yagil Engel**
Industrial Eng. and Management
Technion–Israel Inst. of Technology
dcarmel,yagile@ie.technion.ac.il

Moshe Tennenholtz
Microsoft Israel R&D Center
& Technion–IIT
moshet@microsoft.com

Abstract

Connecting between standard AI planning constructs and a classical cooperative model of transferable-utility coalition games, we introduce the notion of *transferable-utility (TU) planning games*. The key representational property of these games is that coalitions are valued *implicitly* based on their ability to carry out efficient joint plans. On the side of the expressiveness, we show that existing succinct representations of monotonic TU games can be efficiently compiled into TU planning games. On the side of computation, TU planning games allow us to provide some of the strongest to date tractability results for core-existence and core-membership queries in succinct TU coalition games.

Introduction

In many settings, self-interested agents in a multi-agent system require both the ability to *plan* a non-trivial course of action and the ability to *collaborate strategically* with other agents in order to carry out such plans or to improve their efficiency. While to date these two core capabilities—action planning and strategic collaboration—have been studied very successfully, unfortunately they were studied mostly in isolation, by the AI planning and computational game theory communities respectively.

Looking for a common ground between the successful techniques for AI planning and collaboration, one may notice that *both* (i) target *succinctly representable problems* and (ii) exploit the relationship between the *structure of the problems* induced by these succinct representations, and the complexity of solving the problems. Here, however, comes a pitfall. It appears that all existing succinct representations of games assume this or another form of locality and/or compactness of the dependence of agent's payoffs on the choices of other agents.This, however, almost "by definition" requires the planning part of the system to be simplistic as the complexity of planning stems from the, possibly indirect, *global dependence* between the agents' choices.

While due to the reasons above most computational results for AI planning and multi-agency remain effectively tangential, some recent work aims at changing this picture. In particular, Brafman *et al.* (2009) introduce the concept of *planning games* that combines and properly extends *both* certain standard game-theoretic concepts and some recent developments in decomposition-based AI planning (Amir and Engelhardt 2003; Brafman and Domshlak 2008). The key point is that each agent in planning games can in principle influence the utility of each other agent, resulting in global inter-agent dependency within the system. Despite that, considering planning games with *non-transferable utilities (NTU)*, Brafman *et al.* (2009) provide some positive computational results by exploiting the structure of the so called *agent-interaction graph*. This graph captures the global topology of the direct dependencies between the agents' capabilities, and the positive results of Brafman *et al.* hold for tree-shaped agent-interaction graphs.

In this work we consider the model of *planning games with transferable utilities (TU)*, connecting between the idea of planning games and the classical model of TU coalition games. As a solution concept for the latter, we focus on the classical concept of the *core*. We show that both core-existence and core-membership queries in TU planning games can be answered efficiently when the underlying agent-interaction graphs have a $O(1)$-bounded treewidth. Hence, for the TU model we achieve much beyond the current positive results for NTU planning games that hold only for tree-shaped (that is, 1-bounded treewidth) agent-interaction graphs of Brafman *et al.* (2009). Our contribution can be summarized from two viewpoints:

- On the side of planning, we show that our notion of *stable* cost-optimal planning is tractable when cooperative cost-optimal planning is tractable.
- On the side of cooperative game theory, the model of TU planning games is *succinctness-wise* as general as all previous models for general monotone TU coalition games, and for core computations it reveals tractability classes *strictly larger* than obtained for previous succinct models.

Preliminaries and Related Work

Coalition Games and the Core

Let $\Phi = \{\varphi_1, \ldots, \varphi_n\}$ denote a set of agents. A **(transferable utility) coalition game** is a pair (v, Φ) where $v : 2^\Phi \rightarrow \Re^+$ captures the value of all possible coalitions $\Gamma \subseteq \Phi$.

*The work of Brafman, Domshlak, and Engel was supported in part by ISF grant 1101/07; Engel was also supported by an Ali Kaufman fellowship at the Technion.

Semantically, $v(\Gamma)$ corresponds to an amount of welfare, or utility, that Γ can achieve without cooperating with the rest of the agents. Coalition games are often assumed to be monotonic, meaning that $v(\Gamma) \leq v(\Gamma')$ when $\Gamma \subseteq \Gamma' \subseteq \Phi$, and hence in particular the value of Φ (called *the grand coalition*) is the largest of all coalitions. We assume monotonicity throughout this paper.

The key question in cooperative game theory is how the value $v(\Phi)$ of the grand coalition should be divided among the agents. There are several solution concepts that provide various guarantees; in this work we focus on the notion of stable solutions, as captured by the concept of the *core* (von Neumann and Morgenstern 1944).

Definition 1. *A vector $x \in \Re^n$ is an* **imputation** *for a coalition game (v, Φ) if $\sum_{i=1}^{n} x_i = v(\Phi)$. An imputation x for (v, Φ) is in the* **core** *of the game if*

$$\forall \Gamma \subseteq \Phi, \sum_{\varphi_i \in \Gamma} x_i \geq v(\Gamma) \quad (1)$$

A core solution is hence an imputation under which no subset of agents can get a higher amount of utility by acting on its own. For a given imputation x, a coalition that violates its respective constraint (1) is said to be *blocking* for x.

The setting of coalition games poses several interesting challenges to computational game theory. First, the basic representation of a coalition game, called *the characteristic form*, corresponds to the coalitions' value function v being given in its explicit, tabular form. This of course requires space linear in the number of coalitions, and hence exponential in the number of agents. For these reasons recent works have studied various **succinct representations** for cooperative games. Of the rich literature on this topic, we refer to work that explicitly considers complexity of core related computation. Deng and Papadimitriou (1994) consider coalition games representable via an edge-weighted graph. Conitzer and Sandholm (2003) propose a representation for superadditive games; later, the same authors propose the *multi-issue representation* (Conitzer and Sandholm 2004) that is based on an additive decomposition of the value function to a set of subgames. All these three representations can be efficiently reduced to *MC-nets* that were later introduced by Ieong and Shoham (2005). In the MC-net model, the value function v is given via a set of logical rules such that the body of each rule is a conjunctive formula over the agent identities, and the rule head is a numeric weight. For each coalition Γ, the value $v(\Gamma)$ is the sum of the weights of all the rules satisfied by Γ. For example, a rule with body $\varphi_1 \wedge \varphi_2 \wedge \neg\varphi_3$ is satisfied by Γ if $\varphi_1, \varphi_2 \in \Gamma$ and $\varphi_3 \notin \Gamma$. Another popular but non-generic representation is by *weighted voting games*, for which core-related complexity is studied by Elkind *et al.* (2007).

Both multi-issue and MC-nets models are fully expressive, meaning that they can represent any transferable-utility coalition game. Of course, the compactness of the representation is determined by the effectiveness of the additive decomposition to issues or rules. In general, *all* the aforementioned proposals connect between the succinctness of game representation with this or another compact additive decomposition of the value function. This joint property of the previous proposals is important for putting our work in a proper context because, in contrast, our contribution relies on a very different type of structural assumption that places no explicit constraints on value function structure, and in particular, can compactly represent value functions that are not additively decomposable.

Once a game is represented succinctly, it becomes relevant to consider the **complexity of computational tasks** associated with it. The fundamental tasks related to the core are *core membership* (CORE-MEM) and *core existence* (CORE-EX).

CORE-MEM Given a coalition game (v, Φ) and an imputation x, return true iff x is in the core.

CORE-EX Given a coalition game (v, Φ), return true iff there exists an imputation that is a core solution for (v, Φ).

To make the solution operational, we require an algorithm for CORE-EX to also return an imputation in the core, if one exists. In general, with succinct representations the two problems are computationally hard. However, previous works emphasize the potential of exploiting the *structure* of the value function to reduce the complexity of the core-related (and other) computational queries. In particular, for MC-nets, Ieong and Shoham (2005) show that the complexity of both CORE-MEM and CORE-EX is exponential only in the treewidth of a graph having the agents as its nodes and an edge between a pair of agents iff both agents appear in the body of some rule. Later on, we formally defined the notion of treewidth.

Planning Games

The notion of planning games recently proposed by Brafman *et al.* (2009) builds upon an earlier work of Brafman and Domshlak (2008) on extending the classical, single-agent STRIPS planning to multi-agent planning for fully cooperative agents. In STRIPS, the world states are captured by subsets of some atoms (propositions) P, the transitions between states are described via actions A, and each action a is described via its precondition $\mathsf{pre}(a) \subset P$, and effects $\mathsf{add}(a)$ and $\mathsf{del}(a)$. Action a is feasible in state s if $\mathsf{pre}(a) \subseteq s$, and if applied in s, it deterministically transforms the system to state $(s \setminus \mathsf{del}(a)) \cup \mathsf{add}(a)$. Given an initial state $I \subseteq P$, a sequence of actions is feasible if its actions are feasible in the respective states. A feasible action sequence achieves $g \in P$ if g holds in the final state reached by the sequence.

MA-STRIPS, a simple extension of STRIPS suggested by Brafman and Domshlak (2008) for multi-agent settings, is defined as follows.

Definition 2. *An* **MA-STRIPS problem** *for a system of agents $\Phi = \{\varphi_i\}_{i=1}^{n}$ is given by a 5-tuple $\Pi = \langle P, \{A_i\}_{i=1}^{n}, c, I, g \rangle$ where P is a finite set of atoms, $I \subseteq P$ and $g \in P$ encode the initial state and goal, respectively, and, for $1 \leq i \leq n$, A_i is the set of actions of agent φ_i. Each action $a \in A = \bigcup A_i$ has the* STRIPS *syntax and semantics. $c : A \to \Re^+$ is an action cost function.*

A basic technical construct we adopt here is that of **annotated action** (Brafman and Domshlak 2008). An annotated action of agent φ is a tuple $(a, t, \{(j_1, t_1), \ldots, (j_m, t_m)\})$ where $a \in A_i$, $t, t_1, \ldots, t_m$ are time points, and $j_1, \ldots, j_m$ are identities of some other agents. The semantics of such an annotated action is that at time t, φ will perform action a, and it requires agents φ_{j_l} to provide the (j_l-th) precondition of a at time t_{j_l}, respectively.[1] The multi-agent part of the planning algorithms for MA-STRIPS in (Brafman and Domshlak 2008) reasons at the level of *feasible sequences of annotated actions*, or **strategies**, of individual agents. A sequence of annotated actions θ of agent φ is called a strategy if all the action preconditions along θ that have *not* been requested by φ from the other agents can be provided by φ itself in the order postulated by θ. In other words, each strategy of φ is an abstraction of an individual plan for φ to only the "interface", that is, only to actions that have something to do with the other agents. The **cost** of a φ's strategy θ is $c(\theta) = \text{LocalCost}(\theta) + \sum_{a \in \theta} c(a)$, where LocalCost$(\theta)$ is the cost φ incurs to provide all the preconditions it implicitly requires from itself along θ. The **domain of** φ, denoted $\mathcal{D}(\varphi)$, includes all strategies of φ, including the "null strategy" $\perp$, corresponding to the empty sequence. From these building blocks the following concepts are defined:

matching A pair of strategies $\theta \in \mathcal{D}(\varphi)$ and $\theta' \in \mathcal{D}(\varphi')$ are *matching*, denoted by $M(\theta, \theta')$, if (i) whenever θ' requires a precondition from φ at time t, θ contains an action producing this precondition at t, and (ii) φ destroys no precondition requested by φ' in the respective time interval; and vice versa. Note that $M(\theta, \perp)$ where $\perp \in \mathcal{D}(\varphi')$, means that θ in particular does not require φ' to supply any precondition.

a joint strategy $\tilde{\theta}$ of a set of agents $\Gamma \subseteq \Phi$, consists of a strategy $\theta \in \mathcal{D}(\varphi)$ for each $\varphi \in \Gamma$, such that all strategies *are pairwise matching*. The domain $\mathcal{D}(\Gamma)$ is the set of joint strategies of Γ. We use $\tilde{\theta}|_\varphi$ to denote the strategy of agent φ within $\tilde{\theta}$. The cost of a joint strategy $\tilde{\theta}$ is $c(\tilde{\theta}) = \sum_{\varphi \in \Gamma} c(\tilde{\theta}|_\varphi)$. A **plan** π of a set of agents $\Gamma \subseteq \Phi$ is a *self-sufficient* joint strategy for Γ, that is, for each $(a, t, \{(j_i, t_i)\}_{i=1}^m) \in \pi$ and each $l \in \{1, \ldots, m\}$, we have $\varphi_{j_l} \in \Gamma$. The **effects** $\mathsf{eff}(\pi)$ **of a plan** π is the set of atoms $p \in P$ such that $\exists (a, t, \ldots) \in \pi$ with $p \in \mathsf{add}(a)$ (or p is in I, in which case $t = 0$), and $\neg \exists (a, t', \ldots) \in \pi$ with $p \in \mathsf{del}(a)$ and $t' \geq t$. π is a **goal-achieving plan** for Π if $g \in \mathsf{eff}(\pi)$.

the agent interaction graph, *AIG*$_\Pi$, is a graph in which the nodes are the agents, and there is an edge between two agents if an action of one of them either adds or deletes a precondition of some action of the other agent.

Following the previous works on MA-STRIPS, we focus on what is termed the **simple agents** assumption, corresponding to considering agents that (i) can generate their individual local plans in polynomial time, and (ii) exhibit to the rest of the system personal strategies of up to $O(1)$ annotated actions (Brafman and Domshlak 2008; Brafman et al. 2009). Note that assumption (ii) puts a linear bound only on the "interface" between the agents, not on the actual joint, or even individual, plans of the agents; the plans can still be of arbitrary (polynomial) length.

Previous results for MA-STRIPS in the context of simple agents are as follows. Brafman and Domshlak (2008) provide a polynomial-time planning algorithm for problems in which the treewidth of AIG is bounded by a constant. This algorithm can also be straightforwardly extended to compute a plan (i) for a particular subset of agents $\Gamma \subset \Phi$, and/or (ii) one that is *cost-optimal* for Γ, that is, having the cost

$$c_\Gamma = \min_{\pi \in \mathcal{D}(\Gamma), g \in \mathsf{eff}(\pi)} c(\pi). \qquad (2)$$

Brafman *et al.* (2009) extend MA-STRIPS to accommodate self-interested agents, by game-theoretic means. In particular, the coalition planning games model they propose is in the spirit of coalition games with *non-transferable utilities*, and the algorithm they provide computes a stable (core) plan when the AIG is tree-shaped. In this work we consider planning games with *transferable utilities*, and show that much stronger computational results can be achieved in this setting. Somewhat relevant to ours is also the recent work on the so called cooperative boolean games (Dunne et al. 2008), defined over a set of boolean variables, each controlled by a different agent. The authors define solution concepts in the spirit of the core, and show that computing such solutions is hard. The mutual ground of that work and ours is in the usage of knowledge representation to structure cooperative games. However, our focus is on identifying tractable classes of problems by exploiting the structure of interactions between the agents, and, in that context, on the classical notion of core.

TU Planning Games

In planning games, self-interested agents need to cooperate in order to achieve goals that yield a reward.

Definition 3. *A* **TU planning game (TuPG)** *for a system of agents Φ is a tuple $\Pi = \langle P, A, c, I, g, r \rangle$, where all the components but r are as in* MA-STRIPS*, and $r \in \Re^+$ is a specification of a reward associated with achieving g.*

Such a setting is naturally associated with coalition games, which are defined by the value that groups of agents obtain from cooperation. The definition for the value of a coalition in a planning game is straightforward: it is the reward associated with the goal, minus the cost the coalition must incur to attain the goal. Since any coalition Γ will always prefer to perform its optimal plan, the cost associated with Γ is c_Γ as in Eq. 2, with $c_\Gamma = \infty$ if Γ has no goal-achieving plan whatsoever. This leads to a definition of a coalition game, associated with a particular planning game.

Definition 4. *Let $\Pi = \langle P, A, I, g, c, r \rangle$ be a TU planning game for the system of agent Φ. The* **coalition game induced** *by Π is $(v_\Pi \cdot, \Phi)$, where $v_\Pi(\Gamma) = \max\{r - c_\Gamma, 0\}$.*[2]

In this framework, it is assumed that any coalition Γ can execute its optimal plan without the interference of other

[1] For convenience, we sometimes still refer directly to plain actions in strategies, with $a \in \theta$ being equivalent to $\exists (a, t, \{(j_1, t_1), \ldots, (j_m, t_m)\}) \in \theta$.

[2] We omit the subscript Π of $v_\Pi(\cdot)$ when clear from context.

algorithm TUPG-CM-Tree(Π, Φ)

input: a tree-shaped TuPG problem Π over agents Φ, an imputation x

output: is x in the core of the coalition game corresponding to Π

fix a topological ordering $\varphi_1, \ldots, \varphi_k$ over Φ

for $i = k$ down to 1:

  for each $\theta \in \mathcal{D}(\varphi_i)$:

    if for some child φ_j of φ_i, θ has no matches in $\mathcal{D}(\varphi_j)$ **then**

      remove θ from $\mathcal{D}(\varphi_i)$

    if $w_g(\theta) < r$ **and** $M(\theta, \bot)$ for $\bot \in \mathcal{D}(Pa(\varphi_i))$ **then return** false

return true

Figure 1: CORE-MEM algorithm for acyclic AIG.

agents. Formally, if π is a plan for Γ, then all $\Phi \setminus \Gamma$ do $\bot$ under π. Semantically, the problem setup is as follows. A task is published to a given agent system Φ, along with a reward r. An outcome of this protocol is that a plan is selected and r is divided in some way between the agents, that is, the outcome is a pair (π, x) where π is a plan and x is a division of r among the agents Φ. Due to agents' rationality, we can limit ourselves to solutions in which the payment for each agent is at least as high as his cost under π. Therefore, given a plan π, the net welfare to be divided is $r - c(\pi)$. If the grand coalition Φ cooperates, they can execute the globally optimal plan, hence incur the minimal cost c_Φ. A solution to the induced coalition game is an imputation x, dividing the value of the grand coalition, hence $\sum_{i=1}^{n} x_i = r - c_\Phi$. x_i is the net utility given to φ_i, under the globally optimal plan.

In order to ensure that all the agents cooperate with this optimal plan, we need to divide the net utility obtained by this plan in a way that is *stable*, that is, no subset of agents prefers to offer their own plan. In other words, if any smaller coalition performs its plan and gets all of r, its utility should not exceed the sum of its members' individual utility values under the proposed solution. This exactly corresponds to an imputation for which (1) holds, and thus the set of stable plans corresponds to the core of the induced coalition game.

As an example, consider a complex construction project, such as a bridge or a building. The project requires cooperation between various autonomous agents (builders, plumbers, electricians, etc.), and requires coalitions among these agents to solve a planning problem in order to figure out their costs. Finding a core solution ensures the stability of the economically efficient outcome by dividing the net utility in a way that mitigates the incentives of any smaller coalition to bid on its own.

Core Membership

Our first computational result is that CORE-MEM can be efficiently computed for coalition games induced by a wide class of TU planning games. We first present a simplified algorithm that works for TU planning games with tree-shaped AIGs, and generalize the algorithm afterwards.

The algorithm TUPG-CM-Tree (depicted in Figure 1), first fixes an *arbitrary* rooting of AIG, and schematically numbers the agents consistently with that rooting, starting with the root φ_1. The basic concept used by the algorithm is the *weight* $w(\theta)$ of each agent strategy $\theta \in \mathcal{D}(\varphi)$. This weight captures the minimum, over all joint strategies $\tilde{\theta}$ of only the agents in the subtree rooted at φ and having $\tilde{\theta}|_\varphi = \theta$, of the sum of total cost and imputation values of

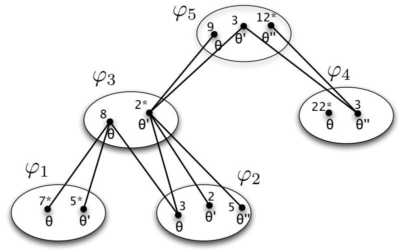

Figure 2: AIG, agent strategies, their costs, and their matching relation for the inline example.

the agents participating in $\tilde{\theta}$. Note that, though the weight is associated with a strategy of a single agent φ, it refers to the "most blocking" *joint* strategy within the subtree of φ that, in particular, includes θ. Crucially, these weights can be computed bottom up from the leafs of AIG as

$$w(\theta) = \begin{cases} 0 & \theta = \bot \\ c(\theta) + x_i + \sum_{\varphi' \in Ch(\varphi_i)} \min_{\substack{\theta' \in \mathcal{D}(\varphi'), \\ M(\theta, \theta')}} w(\theta') & \theta \neq \bot \end{cases}$$

where $\theta \in \mathcal{D}(\varphi_i)$, and $Ch(\varphi)$ are the immediate successors of φ in the rooted AIG. Essentially, a blocking coalition exists if and only if we find a strategy θ in the domain of some agent φ, for which (i) $w(\theta) < r$, and (ii) it does not require preconditions from the parent of φ, meaning that the "subtree joint strategy" responsible for $w(\theta)$ is a plan.[3] The key to proving algorithm correctness is showing that for any plan π, $w(\pi|_\varphi)$ is bounded by the sum of x_{φ_i} and the costs of $\pi|_{\varphi_i}$ over all agents φ_i in the AIG subtree rooted in φ.

Theorem 1. *TUPG-CM-Tree is sound and complete for CORE-MEM in TU planning games with tree-shaped AIGs.*

All proofs are omitted here due to space constraints and are available in a full version technical report. As an example we introduce a system of agents $\Phi = \{\varphi_1, \ldots, \varphi_5\}$, and a reward $r = 20$. The agents' domains and their relationships are depicted in Figure 2 on top of the system's AIG. Each dot in an agents' ellipse denotes a strategy with the cost specified right above it. A line connecting two strategies indicate a match. Goal achieving strategies are denoted by a * next to their cost. The resulting coalitions, their costs, and their values are shown in Table 1a. The optimal plan is by Γ_4, hence $v(\Phi) = 20 - 10 = 10$. Let the input imputation be $x = (0, 2, 4, 2, 2)$. The weights computed by the algorithm are listed in Table 1b. For example, $w(\theta \in \mathcal{D}(\varphi_3)) = 8 + 4 + 5 + 5$, corresponding to local cost + local imputation value + minimal weight of a match in $\mathcal{D}(\varphi_1)$ + weight of the only match in $\mathcal{D}(\varphi_2)$. When the algorithms encounters a strategy that requires no preconditions from the parent (that is, the algorithm finds a plan), it checks whether the weight of that strategy is smaller than the reward. Such strategies in our example are denoted by *

[3] It also has to be ensured that any such plan is goal-achieving. This can be done either by pruning the domain using constraint propagation (Brafman and Domshlak 2008), or, alternatively, by maintaining another weight value per strategy. We omit the technical details to simplify the presentation.

	agents	c_{Γ_i}	$v(\Gamma_i)$		$w(\theta)$	$w(\theta')$	$w(\theta'')$
Γ_1	$\varphi_1\varphi_2\varphi_3$	16	4	φ_1	7	5	
Γ_2	$\varphi_2\varphi_3\varphi_5$	13	7	φ_2	5	4	7
Γ_3	φ_4	22	0	φ_3	22*	10	
Γ_4	$\varphi_2\varphi_3\varphi_4\varphi_5$	10	10	φ_4	24*	5	
Γ_5	$\varphi_4\varphi_5$	15	5	φ_5	21*	20*	19*
	(a)				(b)		

Table 1: (a) coalitions, their costs, and their values, and (b) strategy weights computed by the algorithm; $*$ denotes strategies associated with completed plans.

next to the weight in Table 1b. The weight is at least the reward for $\Gamma_1, \Gamma_2, \Gamma_3$, but it is not so for Γ_5, meaning that this coalition is blocking and the algorithm returns false. Indeed, $v(\Gamma_5) = 5 > 4 = x_4 + x_5$. The imputation is thus not in the core. In contrast, the imputation $(0, 2, 3, 2, 3)$ is in the core: the weights of $\theta \in \mathcal{D}(\varphi_3)$ and $\theta \in \mathcal{D}(\varphi_4)$ are each reduced by 1, and are still not blocking, the weight of the (previously blocking) plan of Γ_5 is 20, meaning it is no longer blocking, and the rest of the weights are the same as before.

Extension to Graphs with Cycles

In optimization problems over graphical structures, tree-specialized algorithms can often be generalized to cyclic graphs using tree decompositions. Whereas no such generalization has been found so far for the tree-specialized algorithm of brafman et al. (2009) for NTU planning games, we now show that such a generalization does exist for our CORE-MEM algorithm for TU planning games.

Definition 5. *A* **tree decomposition** *for a graph* $G = (N, E)$ *is a pair* $(T = (\mathcal{I}, \mathcal{E}), \{\Psi_i | i \in \mathcal{I}\})$*, where* T *is an acyclic graph,* $\Psi_i \subseteq N$ *for all* $i \in \mathcal{I}$*, and (i)* $\bigcup_{i \in \mathcal{I}} \Psi_i = N$*, (ii) for each edge* $(n_1, n_2) \in E$*, there exists* Ψ_i *such that* $n_1, n_2 \in \Psi_i$*, and (iii) for any* $i, j, k \in \mathcal{I}$*, if* j *is on the path from* i *to* k *in* T *then* $\Psi_i \cap \Psi_k \subseteq \Psi_j$*.*

Any graph can be tree-decomposed, and typically in various ways. The width of a tree decomposition is $\max_{i \in \mathcal{I}} |\Psi_i| - 1$, and the **treewidth** of a graph is the minimum width among all its possible tree decompositions.

Let TD_Π denote a tree decomposition of AIG_Π of a problem Π. Each node in TD_Π, which we refer to as *join-node*, refers to some set of agents $\Psi \subseteq \Phi$, and the domain $\mathcal{D}(\Psi)$ includes all the joint strategies of Ψ. We define the matching relation over joint strategies as follows. Let $\tilde{\theta} \in \Psi$ and $\tilde{\theta}' \in \Psi'$. $M(\tilde{\theta}, \tilde{\theta}')$ if $\tilde{\theta}|_\varphi = \tilde{\theta}'|_\varphi$ for any $\varphi \in \Psi \cap \Psi'$. Any collection π of join-node strategies which are pairwise matching is a joint strategy of the respective collection of agents. To see that, let φ and φ' be neighbors in AIG_Π. By property (ii) of tree decomposition, φ and φ' must appear together in at least one join-node Ψ. If $\tilde{\theta} \in \mathcal{D}(\Psi)$, $\theta = \tilde{\theta}|_\varphi$, and $\theta' = \tilde{\theta}|_{\varphi'}$, then $M(\theta, \theta')$ must hold. The matching relation over join-nodes ensures that θ and θ' are the strategies assigned to φ and φ' (respectively) all over the tree, so we can write $\theta = \pi|_\varphi$ and $\theta' = \pi|_{\varphi'}$. Therefore, $M(\pi|_\varphi, \pi|_{\varphi'})$.

To generalize TUPG-CM-Tree to TD_Π, we redefine the weight of a (now joint for agents Ψ) strategy $\tilde{\theta}$. First, let

$$\Psi_{\tilde{\theta}} = \{\varphi \mid \varphi \in \Psi, \tilde{\theta}|_\varphi \neq \perp, \forall \Psi' \in Ch(\Psi) : \varphi \notin \Psi'\}, \quad (3)$$

that is, $\Psi_{\tilde{\theta}}$ consists of the agents of Ψ that participate in $\tilde{\theta}$ and are "new" to Ψ with respect to its children. Then

$$w(\tilde{\theta}) = \sum_{\varphi_j \in \Psi_{\tilde{\theta}}} (c(\tilde{\theta}|_{\varphi_j}) + x_j) + \sum_{\Psi' \in Ch(\Psi)} \min_{\substack{\tilde{\theta}' \in \mathcal{D}(\Psi'),\\ M(\tilde{\theta}, \tilde{\theta}')}} w(\tilde{\theta}'). \quad (4)$$

The condition $\tilde{\theta}|_\varphi \neq \perp$ in (3) ensures that we count the imputation value only for the agents in Ψ that actively participate in the joint strategy, and the condition $\forall \Psi' \in Ch(\Psi), \varphi \notin \Psi'$ in (3) prevents double counting of agents in the intersection of the join-nodes. In the resulting algorithm, named TUPG-CM, we traverse TD_Π bottom-up, and compute the weight of each joint strategy of each join-node incrementally as before. Again, a blocking coalition exists iff there is a joint strategy with weight smaller than r.

Theorem 2. *TUPG-CM is sound and complete for CORE-MEM in arbitrary TU planning games, and its time complexity is exponential only in the treewidth of AIG.*

Core Existence

Considering now the CORE-EX problem, we first note that the existence of a core solution in TU planning games depends in an interesting way on the amount of the reward. First, it is easy to see that for any planning problem, there exists a reward amount that *ensures core existence*. For instance, if the reward is exactly the cost of the optimal plan c_Φ, then the coalition game is trivial (all values are zero) and hence the zero imputation is in the core. In fact, in some cases this is the only reward for which a core solution exists; if two disjoint coalitions have goal-achieving plans of cost c_Φ, and $r - c_\Phi > 0$, then any imputation will be blocked by one of these coalitions.

This is a special case of the following characterization. Let Γ be a minimal coalition with a goal-achieving plan of the globally optimal cost c_Φ, and let $\overline{\Gamma} = \Phi \setminus \Gamma$. The *second-best cost* with respect to Γ is defined as $c_{\overline{\Gamma}}$, that is the optimal value of a coalition disjoint from Γ.

Proposition 1. *If there exists a coalition* Γ *with* $c_\Gamma = c_\Phi$ *for which* $r > c_{\overline{\Gamma}}$*, then the core is empty.*

The converse does not hold, however; it is possible that the reward is smaller than any second-best cost, and yet the core is empty. For example, let $c_{\{\varphi_1\}} = c_{\{\varphi_2\}} = 3$, let $\Gamma = \{\varphi_1, \varphi_2\}$, and $c_\Gamma = 2$. Assume also that $\overline{\Gamma}$ do not have a goal-achieving plan at all, meaning that $c_{\overline{\Gamma}} = \infty$ is the unique second-best cost. Yet, for $r > 4$ the core is empty because there is no way to divide $v(\Phi) = r - 2$ between φ_1 and φ_2 such that each gets at least its own value $r - 3$.

The fact that lower reward is more stable may seem counter-intuitive; however, it has a simple economic explanation. With a large reward, agents have more leverage to act on their own and ensure themselves a high amount, and this leads to inefficiencies. With a small return, agents can gain only if they cooperate in order to reduce costs.

When Proposition 1 is not applicable, CORE-EX has to be solved explicitly. The problem corresponds to a linear program (LP) with objective $\min_x \sum_{\varphi_i \in \Phi} x_i$ and a set of constraints corresponding to (1). If the objective value of the solution to this LP is no higher than $v(\Phi)$, then it is a core solution, otherwise the core is empty. The number of constraints in this LP is exponential in the number of agents. However, the set of constraints need not be created explicitly. It is well-known that the ellipsoid algorithm for linear programming can be executed without the explicit set of constraints. This requires availability of a *separation oracle*, which is a polynomial-time algorithm that gets a prospect solution as input, and either confirms that all the constraints are satisfied, or returns a violated constraint. In our case, the TUPG-CM algorithm for core-membership can serve as a separation oracle—for a given imputation it determines whether it is in the core or not, and if not, it identifies the coalition that blocks it (corresponding to a violated constraint). Identifying this coalition requires a very minor extension to the algorithm.[4]

Theorem 3. *CORE-EX for TU planning games can be solved in time exponential only in the treewidth of AIG.*

Controlling Core Existence

Given that the choice of the reward in planning games has a substantial impact on the core non-emptiness, consider now a slightly different problem. Suppose that an external authority wishes to ensure stability of the most efficient outcome (meaning, Φ committing to a globally cost-optimal plan) by choosing both the reward and an imputation. Of course, as we just discussed, a zero imputation under a zero net reward will ensure stability of any cost-optimal plan of Φ. However, such a solution is not likely to work in practice because agents would normally have other, more profitable things to do. To ensure both stability and participation, the authority hence wishes to find the highest amount, up to the second cost (or a reserve price) under which a core solution still exists. Formally, a **controllable-reward** TU planning games are similar to regular TU planning games with only the reward r being replaced by an *upper bound* $\rho \geq 0$ on the reward. Hence our problem, denoted R-CORE-EX, is

$$\max\{r \mid \text{CORE-EX}(\langle P, A, c, I, g, r\rangle) = \text{true},\ r \leq \rho\}.$$

One way to solve R-CORE-EX is by an iterative application of an algorithm for CORE-EX, starting from a high reward and gradually reducing it by some ϵ until the answer is positive. This procedure converges to the highest (up to ϵ) "stable" reward for the following reason: if there exists a core imputation under reward r, then for any $r > \epsilon > 0$, there also exists a core imputation under reward $r - \epsilon$. To see that, simply take a core imputation x under r, and reduce one of its elements by ϵ. The value of each coalition given the lower reward is lower by exactly ϵ, while its total payment under the new imputation is lower by *at most* ϵ. Having mentioned that, we note that R-CORE-EX can also be solved directly via LP:

$$\max r \quad \text{s.t.} \begin{cases} r \leq \rho, & \sum_{i=1}^{n} x_i = r - c_\Phi \\ \forall \Gamma \subseteq \Phi: & \sum_{\varphi_i \in \Gamma} x_i \geq r - c_\Gamma \end{cases}$$

For a reward and an imputation which satisfy the first two constraints, TUPG-CM again serves as a separation oracle.

[4] This relationship between CORE-MEM and CORE-EX was noted and employed by Ieong and Shoham (2005).

Theorem 4. *R-CORE-EX for controllable-reward TU planning games can be solved in time exponential only in the treewidth of AIG.*

Succinctness and Tractability

It is easy to show that TU planning games is a fully general representation of monotonic TU coalition games. Interestingly, it turns out that they are also at least as general as all existing succinct representations surveyed earlier in the paper, with respect to both succinctness and tractability results.

Theorem 5. *Any coalition game (v, Φ) represented by either a MC-net, weighted graph, multi-issue game, or superadditive game can be efficiently compiled into a TU planning game Π with an efficiently computable bijective mapping between the cores of (v, Φ) and Π.*

Importantly, as formalized by Theorem 6 below, the construction in the proof of Theorem 5 reveals that our tractability results for TU-planning games *strictly generalize* the tractability results of Ieong and Shoham (2005) in the context of monotone coalition games.

Theorem 6. *Any MC-net M with treewidth $tw(M)$ can be efficiently compiled into a TU planning game Π_M having AIG with treewidth $tw(\Pi_M) \leq tw(M) + 2$. Moreover, there exist MC-nets M such that $tw(M) = |\Phi|$ and $tw(\Pi_M) = 1$.*

A simple example shows the second part of the claim: the graph of any MC-net containing a rule whose body involves k agents necessarily includes a clique of size k. In contrast, a planning game can express this synergy with just a *tree* of size k. The key insight of this result is as follows. Whereas edges in previous dependency models such as MC nets must reflect *all the synergies* among agents, TU planning games reveal and exploit a more refined knowledge representation structure, in which two agents are connected with an edge only if one agent has a strategy that *directly enables or disables* a strategy of the other agent.

References

Amir, E., and Engelhardt, B. 2003. Factored planning. In *IJCAI*.

Brafman, R., and Domshlak, C. 2008. From one to many: Planning for loosely coupled multi-agent systems. In *ICAPS*.

Brafman, R.; Domshlak, C.; Engel, Y.; and Tennenholtz, M. 2009. Planning Games. In *IJCAI*.

Conitzer, V., and Sandholm, T. 2003. Complexity of determining nonemptiness of the core. In *IJCAI*.

Conitzer, V., and Sandholm, T. 2004. Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *AAAI*.

Deng, X., and Papadimitriou, C. H. 1994. On the complexity of cooperative solution concepts. *Math. of OR* 19:257–266.

Dunne, P.; van der Hoek, W.; Kraus, S.; and Wooldridge, M. 2008. Cooperative boolean games. In *AAMAS*.

Elkind, E.; Goldberg, L.; Goldberg, P.; and Wooldridge, M. 2007. Computational complexity of weighted threshold games. In *AAAI*.

Ieong, S., and Shoham, Y. 2005. Marginal contribution nets: A compact representation scheme for coalitional games. In *EC*.

von Neumann, J., and Morgenstern, O. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.