Role Assignment for Game-Theoretic Cooperation *

Catherine Moon and Vincent Conitzer

Duke University, Durham, NC 27708, USA

Abstract. In multiagent systems, often agents need to be assigned to different *roles*. Multiple aspects should be taken into account for this, such as agents' skills and constraints posed by existing assignments. In this paper, we focus on another aspect: when the agents are self-interested, careful role assignment is necessary to make cooperative behavior an equilibrium of the repeated game. We formalize this problem and provide an easy-to-check necessary and sufficient condition for a given role assignment to induce cooperation. However, we show that finding whether such a role assignment exists is in general NP-hard. Nevertheless, we give two algorithms for solving the problem. The first is based on a mixed-integer linear program formulation. The second is based on a dynamic program, and runs in pseudopolynomial time if the number of agents is constant. However, the first algorithm is much, much faster in our experiments.

1 Introduction

Role assignment is an important problem in the design of multiagent systems. When multiple agents come together to execute a plan, there is generally a natural set of roles in the plan to which the agents need to be assigned. There are, of course, many aspects to take into account in such role assignment. It may be impossible to assign certain combinations of roles to the same agent, for example due to resource constraints. Some agents may be more skilled at a given role than others.

In this paper, we assume agents are interchangeable and instead consider another aspect: if the agents are self-interested, then the assignment of roles has certain *game-theoretic* ramifications. A careful assignment of roles might induce cooperation whereas a careless assignment may result in incentives for an agent to defect. Specifically, we consider a setting where there are multiple *minigames* in which agents need to be assigned roles. These games are then infinitely repeated, and roles cannot be reassigned later on.[1] It is well known, via the folk

* We thank ARO and NSF for support under grants W911NF-12-1-0550, W911NF-11-1-0332, IIS-0953756, CCF-1101659, and CCF-1337215.

[1] We disallow reassigning agents in our model because in many contexts, such reassignment is infeasible or prohibitively costly due to agents having built up personalized infrastructure or specialized expertise for their roles, as is easily seen in some of the examples in the next paragraph.

theorem, that sometimes cooperation can be sustained in infinitely repeated games due to the threat of future punishment. Nevertheless, some infinitely repeated games, in and of themselves, do not offer sufficient opportunity to punish certain players for misbehaving. If so, cooperation may still be attained by the threat of punishing the defecting agent in *another* (mini)game. But for this to be effective, the defecting agent needs to be assigned the right role in the other minigame. This is the game-theoretic role-assignment problem that we study in this paper.

Our work contrasts with much work in game theory in which the model zooms in on a single setting without considering it in its broader strategic context. In such models, firms make production and investment decisions based on competition in a single market; teammates decide on how much effort to put in on a single project; and countries decide whether to abide by an agreement on, for instance, reducing pollution. In reality, however, it is rare to have an isolated problem at hand, as the same agents generally interact with each other in other settings as well. Firms often compete in several markets (e.g., on computers, phones, cameras, and displays); members of a team usually work on several projects simultaneously; and countries interact with each other in other contexts, say trade agreements, as well.

Looking at a problem in such an isolated manner can be limiting. There are games where a player has insufficient incentive to play the "cooperative" action, as the payoff from that action and the threat of punishment for defection are not high enough. In such scenarios, putting two or more games with compensating asymmetries can leave hope for cooperation. A firm may allow another firm to dominate one market in return for dominance in another; a team member may agree to take on an undesirable task on one project in return for a desirable one on another; and a country may agree to a severe emissions-reducing role in one agreement in return for being given a desirable role in a trade agreement.

In this paper, we first formalize this setup. Subsequently, we give useful necessary and sufficient conditions for a role assignment to sustain cooperation. We then consider the computational problem of finding a role assignment satisfying these conditions. We show that this problem is NP-hard. We then give two algorithms for solving the problem nonetheless. One relies on an integer program formulation; the other relies on a dynamic programming formulation. We show the latter solves the problem in pseudopolynomial time when the number of agents is constant. However, in our experiments, the former algorithm is much, much faster, as shown at the end of our paper.

2 Motivating Example

Consider two individuals (e.g., faculty members or board members) that together need to constitute two distinct committees. Each of the committees needs a chair and another member; these are the roles we need to assign to the two individuals. Each committee's chair can choose to behave selfishly or cooperatively. Each committee's other member can choose to sabotage the committee or be cooperative. The precise payoffs differ slightly across the two committees because of their different charges. (For example, acting selfishly as the chair of a graduate admissions committee is likely to lead to different payoffs than acting selfishly as the chair of a faculty search committee.) The payoffs are as follows.

		Member	
		sabotage	cooperate
Chair	selfish	2, 1	3, 0
	cooperate	1, 1	2, 2

		Member	
		sabotage	cooperate
Chair	selfish	2, 1	4, 0
	cooperate	0, 1	2, 3

Let us first consider each of these two minigames separately. If the minigame is only played once, the chair has a strictly dominant strategy of playing selfishly (and hence, by iterated dominance, the other member will sabotage the committee). Even if the game is repeated (with a discount factor $\delta < 1$), we cannot sustain the (cooperate, cooperate) outcome forever. This is because the chair would receive a payoff of 2 in each round from this outcome—but defecting to playing selfishly would give her an immediate utility of 3 or 4 in that round after which she can still guarantee herself a utility of at least 2 in each remaining round by playing selfishly.

Now let us consider the minigames together. If the same agent is assigned as chair in each minigame, again we could not sustain the (cooperate, cooperate) outcome in both minigames, because the chair would gain $3 + 4$ immediately from defecting and still be able to obtain $2 + 2$ in each round forever after. On the other hand, if each agent is chair in one game, then with a reasonably high discount factor, (cooperate, cooperate) can be sustained. For suppose the chair of the second committee deviates by acting selfishly in that committee. This will give her an immediate gain of $4 - 2 = 2$. However, the other agent can respond by playing selfishly on committee 1 and sabotaging committee 2 forever after. Hence in each later round the original defector can get only $1 + 2 = 3$ instead of the $2+2 = 4$ from both agents cooperating, resulting in a loss of 1 in each round relative to cooperation. Hence, if δ is such that $2 \leq \delta/(1-\delta)$, the defection does not benefit her in the long run. This shows that linking the minigames allows us to attain cooperative behavior where this would not have been possible in each individual minigame separately. It also illustrates the importance of assigning the roles carefully in order to attain cooperation.

One may wonder what would happen if we link the minigames in a single-shot (i.e., not repeated) context. This would correspond to the case $\delta = 0$, so that the above formula indicates that cooperation is not attained in this case. In fact, linking minigames cannot help in single-shot games in general: in a single-shot model, any equilibrium of the (linked) game must consist simply of playing an equilibrium of each individual minigame. (Otherwise, a player could improve her overall payoff by deviating in a minigame where she is not best-responding.) Linking becomes useful only when the game is repeated, because then one's actions in one minigame can affect one's future payoffs in other minigames, by affecting other players' future actions. This is why the repeated game aspect is essential to our model.

3 Definitions

When there is a risk of confusion, we distinguish between *minigames* (of which there are two in the example above, corresponding to the two committees) and the larger *metagame* that the minigames together constitute after roles have been assigned to agents. Note that technically the players in minigames are roles, not agents, and the metagame among the agents is not defined until roles have been assigned to them. Let $N = \{1, \dots, n\}$ be the set of agents and G the set of minigames. We assume each minigame has n *roles* (such as committee chair or committee member); if this is not the case, we can simply add dummy roles. For each minigame $g \in G$ and each role r in g, there is a set of actions A_r^g for the agent in that role to play, and a utility function u_r^g that takes as input a profile of actions $\boldsymbol{a}^g$ in g, consisting of one action a_r^g for each role r in g, and as output returns the utility of the agent in role r from this minigame. An agent i's total payoff in round t is $\sum_{g \in G} u^g_{r(i,g)}(\boldsymbol{a}^g(t))$, where $r(i,g)$ denotes the role assigned to agent i in minigame g and $\boldsymbol{a}^g(t)$ is the profile played in minigame g in round t. For the repeated game, we consider both discounted payoff $(\sum_{t=0}^{\infty} \delta^t \sum_{g \in G} u^g_{r(i,g)}(\boldsymbol{a}^g(t)))$ and limit average payoff $(\liminf_{T \to \infty} (1/T) \sum_{t=0}^{T-1} \sum_{g \in G} u^g_{r(i,g)}(\boldsymbol{a}^g(t)))$. We assume perfect monitoring, i.e., agents observe all actions taken by all agents in prior rounds. All this is common knowledge, and so is the role assignment $r(i,g)$ once the agents play the game.

Our main interest is in assessing whether a particular outcome can be sustained in repeated play. We assume that for each game g, for each role r in g, there is a distinguished action $a_r^{*g} \in A_r^g$ that we call the *cooperative* action. (There is no requirement that this action is truly "cooperative" in the sense of increasing other agents' utilities; it can be any action, for example one that a principal assigning the roles would like to see happen for exogenous reasons.) Our main question is whether there exists a role assignment function $r(i,g)$ such that there is an equilibrium of the repeated game where every agent always plays the cooperative action in every role assigned to her. For this question, the key issue is which roles (from different minigames) are bundled together, rather than which particular agent is assigned this bundle of roles. We postpone the definition of this question as a computational problem until we have done some further simplifying analysis.

4 Related Literature

The assignment of roles in multiagent systems has of course received previous attention, especially in domains such as RoboCup soccer. However, we are not aware of any multiagent systems literature on assigning roles across multiple games in a way to achieve game-theoretically stable cooperation.

In the economics literature, some of the first work to recognize the effect of playing multiple games in parallel took place in research on industrial behavior. In 1955, Corwin Edwards proposed the possibility that multimarket contact between firms could allow them to reach strategically stable arrangements that could not be reached in a single market and thereby foster anticompetitive outcomes. The first theoretical work to explore the effect on economic performance investigates the effects of cost- and demand-based linkages across markets in the context of static oligopolistic markets [4]. Other work concerns the effect of multimarket contact on the degree of cooperation that firms can sustain in settings of repeated competition, by exploring the set of credible nonbinding agreements available to firms [2].

Papers by Folmer and von Mouche [5] and Just and Netanyahu [8] concern how the structure of the linked component games affect the potential for cooperation. Both papers identify the expansion in the bargaining set through linkage as the key to potential increase in cooperation. Just and Netanyahu [8] further examine linking common game classes such as the prisoner's dilemma, assurance, iterated dominance, and chicken games; they do not observe the chances of coming to a fully cooperative equilibrium increasing significantly except for the case of linking prisoner's dilemma games.

The intuition that linking games relaxes incentive constraints and the results on what game structure allows linkage to lead to cooperation are both in accordance with the results by Jackson and Sonnenschein [7]. These authors formally address the relaxation of incentive constraints through linking decision problems and the resulting efficiency gains in the general context of social choice with preference announcements. The efficiency of linking decisions is demonstrated as the number of decisions tend to infinity, relying on the law of large numbers. The authors also suggest, by means of example, that the correlation on intensity between the issues affects the gains expected from linking decisions.

5 Theoretical Analysis

In this section, we show that to know whether or not our problem has a solution comes down to a single number per minigame role. The intuition that allows us to show this is as follows. To determine whether a given agent i will defect (i.e., play something other than the cooperative action in some role assigned to her), by the folk theorem, we may assume that all other agents will play their cooperative actions until some defection has taken place, after which they maximally punish agent i (in *all* games, not just the ones in which she defected). Thus, in the round in which agent i defects, she may as well play the single-round best-response to the cooperative actions in every role assigned to her; afterwards, she will forever receive the best she can do in response to maximal punishment. (Since we only consider Nash equilibrium, we do not have to worry about multiple agents deviating.) The net effect of the defection on i's utility may be positive or negative for any given role; whether i will defect depends solely on the sum of these effects. We now formalize this. Recall that a^{*g} is the profile of cooperative actions for minigame g.

Definition 1. *Given a minigame g and a role r in g, let c_r^g denote the (long-run)* ***cooperation value*** *for that role. With limit average payoffs, $c_r^g = u_r^g(\boldsymbol{a}^{*g})$. With discounted payoffs, $c_r^g = \sum_{t=0}^{\infty} \delta^t u_r^g(\boldsymbol{a}^{*g}) = u_r^g(\boldsymbol{a}^{*g})/(1-\delta)$.*

Next, we want to specify the defection value. This requires us to know what utility a player will get in rounds after defection, which depends on how effective the other players are in punishing. In a two-player game, the punishing player should play a minimax strategy—i.e., play as if she were playing a zero-sum game where her utility is the negative of that of the defecting player. With three or more players, an important question is whether the players other than the defector can coordinate (i.e., correlate) their strategies. If not, this leads to NP-hardness [3]. Therefore, we assume that they can correlate, which alllows polynomial-time computability [10]. Formally, when the player in role r has defected, the remaining players $-r$ will play

$$\arg\min_{\sigma_{-r}^g} \max_{a_r^g} u_r^g(a_r^g, \sigma_{-r}^g)$$

where σ_{-r}^g is a mixed strategy for the players $-r$ (allowed to be correlated if there are two or more players in $-r$). In the case of discounted payoffs, we also need to know the utility a player will get in the first round she defects; this is $\max_{a_r^g} u_r^g(a_r^g, a_{-r}^{*g})$.

Definition 2. *Given a minigame g and a role r in g, let d_r^g denote the (long-run)* ***defection value*** *for that role. With limit average payoffs, $d_r^g = \min_{\sigma_{-r}^g} \max_{a_r^g} u_r^g(a_r^g, \sigma_{-r}^g)$. With discounted payoffs, $d_r^g = \max_{a_r^g} u_r^g(a_r^g, a_{-r}^{*g}) + \sum_{t=1}^{\infty} \delta^t \min_{\sigma_{-r}^g} \max_{a_r^g} u_r^g(a_r^g, \sigma_{-r}^g) = \max_{a_r^g} u_r^g(a_r^g, a_{-r}^{*g}) + \frac{\delta}{1-\delta} \min_{\sigma_{-r}^g} \max_{a_r^g} u_r^g(a_r^g, \sigma_{-r}^g)$.*

In the end, what matters is the net effect of defection.

Definition 3. *Given a minigame g and a role r in g, let $m_r^g = c_r^g - d_r^g$ denote the* ***robustness measure*** *for that role.*

The theorem now says that we can obtain the desired outcome if and only if there is an assignment that gives each agent a nonnegative sum of robustness measures.

Theorem 1. *The repeated metagame has an equilibrium (allowing correlated punishment) in which on the path of play, in every round t, in every minigame g, every role r plays a_r^{*g} if and only if the assignment $r(i,g)$ is such that for all i, $\sum_g m_{r(i,g)}^g \geq 0$.*

Proof. We first prove the "if" direction, supposing that $\sum_g m_{r(i,g)}^g \geq 0$ for all i. Consider a grim trigger strategy profile where all players cooperate (play a_r^{*g}) in every role r to which they have been assigned as long as everyone else does so; if some player (say i) has deviated, the other players $-i$ switch to maximally punishing i via correlated punishment—that is, in every game g, they play a strategy in $\arg\min_{\sigma_{-r(i,g)}^g} \max_{a_{r(i,g)}^g} u_{r(i,g)}^g(a_{r(i,g)}^g, \sigma_{-r(i,g)}^g)$. We must show this strategy profile is an equilibrium. Consider an arbitrary agent i. Not deviating will give i a long-term utility of $\sum_g c_{r(i,g)}^g$. What about deviating? Without loss of generality, suppose i deviates in the first round. The highest expected payoff i can obtain in that first round is $\sum_g \max_{a_{r(i,g)}^g} u_{r(i,g)}^g(a_{r(i,g)}^g, a_{-r(i,g)}^{*g})$; in every remaining round, she can obtain at most $\sum_g \min_{\sigma_{-r(i,g)}^g} \max_{a_{r(i,g)}^g} u_{r(i,g)}^g(a_{r(i,g)}^g, \sigma_{-r(i,g)}^g)$. Hence, her long-term utility is at most $\sum_g d_{r(i,g)}^g$. But by assumption, $\sum_g m_{r(i,g)}^g \geq 0$, which is equivalent to $\sum_g c_{r(i,g)}^g \geq \sum_g d_{r(i,g)}^g$. So agent i has no incentive to deviate.

We now prove the "only if" direction, supposing that $\sum_g m_{r(i,g)}^g < 0$ for some i. Consider a strategy profile where on the path of play all players cooperate (play a_r^{*g}). Again, not deviating will give player i a long-term utility of $\sum_g c_{r(i,g)}^g$. By the minimax theorem, $\min_{\sigma_{-r(i,g)}^g} \max_{a_{r(i,g)}^g} u_{r(i,g)}^g(a_{r(i,g)}^g, \sigma_{-r(i,g)}^g) = \max_{\sigma_{r(i,g)}^g} \min_{a_{-r(i,g)}^g} u_{r(i,g)}^g(\sigma_{r(i,g)}^g, a_{-r(i,g)}^g)$. Consider the deviating strategy where player i plays, in every minigame g, a strategy from $\arg\max_{a_{r(i,g)}^g} u_{r(i,g)}^g(a_{r(i,g)}^g, a_{-r(i,g)}^{*g})$ in the first round, and a mixed strategy from $\arg\max_{\sigma_{r(i,g)}^g} \min_{a_{-r(i,g)}^g} u_{r(i,g)}^g(\sigma_{r(i,g)}^g, a_{-r(i,g)}^g)$. Then this guarantees her a long-term utility of at least $d_{r(i,g)}^g$ in each minigame g. By assumption, $\sum_g m_{r(i,g)}^g < 0$, which is equivalent to $\sum_g c_{r(i,g)}^g < \sum_g d_{r(i,g)}^g$. So player i has an incentive to deviate and the original strategy profile is not an equilibrium.

Because it is straightforward to compute the robustness measures from the minigames using the formulas above, Theorem 1 allows us to efficiently check whether a given role assignment has the desired behavior as an equilibrium. It also reduces the problem of *finding* such a role assignment to the following computational problem (which, by the above, is in NP).

Definition 4 (ROLE-ASSIGNMENT). *We are given the m_r^g (a vector of $n|G|$ numbers). We are asked whether there exists a function $r(i,g)$ that maps players one-to-one to the roles of the game g, such that for all i, $\sum_g m_{r(i,g)}^g \geq 0$.*

6 Complexity of ROLE-ASSIGNMENT

In this section, we show that the ROLE-ASSIGNMENT problem is NP-complete. We first show that it is weakly NP-complete even in an extremely restricted special case, namely, the case where we have only two players and each minigame has the following structure.

		Active	
		cooperate	defect
Passive	(no choice)	$-x_g$ x_g	0 0

Each minigame has two roles, Active and Passive. Passive has no choice. Active can choose to defect, in which case both players get 0, or to cooperate, in which case Active gets $-x_g \leq 0$ and Passive gets $x_g \geq 0$. Hence, the robustness values are $m^g_{\text{Active}} = -x_g$ and $m^g_{\text{Passive}} = x_g$, for all $g \in G$. We call these games *Two-Player Active-Passive (2PAP)* games. Intuitively, cooperating in a game means giving the other player a specified gift, and enabling cooperation in all games requires that we balance the gifts exactly between the players.[2] This suggests a reduction from the PARTITION problem.

Theorem 2. *ROLE-ASSIGNMENT is (weakly) NP-complete for 2PAP games.*

Proof. We have already noted that ROLE-ASSIGNMENT is in NP. To prove NP-hardness, we reduce from the PARTITION problem, in which we are given a set of integers $\{w_1, \ldots, w_q\}$, and are asked whether there exists a subset $S \subseteq \{1, \ldots, q\}$ such that $\sum_{j \in S} w_j = \sum_{j=1}^{q} w_i/2$. For an arbitrary instance of the PARTITION problem, we construct the following ROLE-ASSIGNMENT instance with 2PAP games. For each $j \in \{1, \ldots, q\}$, create a 2PAP game $g(j)$ with $x_{g(j)} = w_j$.

If a solution S to the PARTITION instance exists, then assign agent 1 to the Active role in all $g(j)$ with $j \in S$ and to the Passive role in all other games. Then, we have $\sum_g m^g_{r(1,g)} = -\sum_{j \in S} w_j + \sum_{j \notin S} w_j = 0$ and $\sum_g m^g_{r(2,g)} = -\sum_{j \notin S} w_j + \sum_{j \in S} w_j = 0$. Hence a solution to the ROlE-ASSIGNMENT instance exists.

Conversely, if a solution to the ROLE-ASSIGNMENT instance exists, let S be the set of all j such that agent 1 is assigned to the Active role in $g(j)$. We know $0 \leq \sum_g m^g_{r(1,g)} = -\sum_{j \in S} w_j + \sum_{j \notin S} w_j$ and $0 \leq \sum_g m^g_{r(2,g)} = -\sum_{j \notin S} w_j + \sum_{j \in S} w_j$. Hence $\sum_{j \in S} w_j = \sum_{j \notin S} w_j$ and S is a solution to the PARTITION instance.

In Subsection 7.2, we will show that the ROLE-ASSIGNMENT problem can in fact be solved in pseudopolynomial time when there are at most a constant number of agents. We now proceed to prove strong NP-completeness for n-Player Active-Passive (nPAP) games, in which each minigame g has one Active and $n-1$ Passive players, and the Active player can choose to make a specified gift x_g that will be equally divided among the other players (so they each receive $x_g/(n-1)$). The reduction is based on the strongly NP-hard 3-PARTITION problem.

[2] One may wonder why cooperation is desirable at all in these games, but note that the reduction will work just as well if Passive receives a sufficiently small bonus ϵ when receiving a gift and Active does not have to pay this bonus. Alternatively, the principal may have an exogenous reason for preferring cooperation.

Theorem 3. *ROLE-ASSIGNMENT is (strongly) NP-complete for nPAP games.*

Proof. Again, we have already noted that ROLE-ASSIGNMENT is in NP. To prove NP-hardness, we reduce from the 3-PARTITION problem, in which we are given a set of integers $\{w_1, \ldots, w_q\}$ with q divisible by 3, and asked whether $\{1, \ldots, q\}$ can be partitioned into $S_1, \ldots, S_{q/3}$ such that for each i, $\sum_{j \in S_i} w_j = B$ where $B = (3/q) \sum_{j=1}^{q} w_j$. (3-PARTITION remains NP-complete even when for each j we have $B/4 < w_j < B/2$, in which case we must have $|S_i| = 3$ for each i.) For an arbitrary instance of the 3-PARTITION problem, we construct the following ROLE-ASSIGNMENT instance with $n = q/3$ players. For each $j \in \{1, \ldots, q\}$, create an nPAP game $g(j)$ with $x_{g(j)} = w_j$.

If a solution S to the 3-PARTITION instance exists, then assign agent i to the Active role in all $g(j)$ with $j \in S_i$ and to the Passive role in all other games. Then, for each i, we have $\sum_g m^g_{r(i,g)} = -\sum_{j \in S_i} w_j + \sum_{j \notin S_i} w_j/(n-1) = -B+(n-1)B/(n-1) = 0$. Hence a solution to the ROlE-ASSIGNMENT instance exists.

Conversely, if a solution to the ROLE-ASSIGNMENT instance exists, let S_i be the set of all j such that agent i is assigned to the Active role in $g(j)$. We know that for every agent i, $0 \leq \sum_g m^g_{r(1,g)} = -\sum_{j \in S_i} w_j + \sum_{j \notin S_i} w_j/(n-1)$. Because $\sum_{i=1}^{n}[-\sum_{j \in S_i} w_j + \sum_{j \notin S_i} w_j/(n-1)] = \sum_{j=1}^{n}[-w_j + (n-1)w_j/(n-1))] = 0$, it follows that for all i $-\sum_{j \in S_i} w_j + \sum_{j \notin S_i} w_j/(n-1) = 0$. Because $\sum_{j \notin S_i} w_j = nB - \sum_{j \in S_i} w_j$, we have $\sum_{j \in S_i} w_j = nB/(n-1) - \sum_{j \in S_i} w_j/(n-1)$, or equivalently $\sum_{j \in S_i} w_j = B$. Hence the S_i constitute a solution to the 3-PARTITION instance.

7 Algorithms for Role Assignment

In this section, we give two algorithms for ROLE-ASSIGNMENT.

7.1 Integer Program

First, we reduce ROLE-ASSIGNMENT to the integer program (IP) in Figure 1. Combining this with any IP solver results in an algorithm for ROLE-ASSIGNMENT. (We will use CPLEX in our experiments.) The robustness measure m^g_r for each minigame g and each role r in g is a parameter of the integer program. We have an indicator variable $b(i, g, r) \in \{0, 1\}$ for each agent i, each minigame g, and each role r in g. ($b(i, g, r) = 1$ if and only if i is assigned role r in g.) There is another variable v which the solver will end up setting to the minimum aggregate robustness value, $\min_i \sum_g m^g_{r(i,g)}$; maximizing this is the objective of the IP. Hence, the IP not only determines whether the ROLE-ASSIGNMENT instance has a solution (which is the case if and only if the optimal objective value is nonnegative), but also the "most robust" solution.

$$
\begin{array}{lr}
\textbf{maximize } v & \\
\textbf{subject to} & \\
(\forall i) \;\; v - \sum_g \sum_{r \text{ in } g} m_r^g b_{i,g,r} \leq 0 & \textit{(min. robustness)} \\
(\forall g, r \text{ in } g) \;\; \sum_i b_{i,g,r} = 1 & \textit{(one player per role)} \\
(\forall i, g) \;\; \sum_{r \text{ in } g} b_{i,g,r} = 1 & \textit{(one role per player per game)}
\end{array}
$$

Fig. 1. Integer program for ROLE ASSIGNMENT.

7.2 Dynamic Program

Even though the general n-player ROLE-ASSIGNMENT problem is strongly NP-complete (Theorem 3), below we give a dynamic programming algorithm that solves it in pseudopolynomial time when the number of agents is constant. For the purpose of presenting this algorithm, we assume that the payoffs are integers. (Of course, any rational numbers could be scaled up to integers).

The algorithm takes as input the the robustness measure m_r^g for each minigame $g \in G$ and each role r in g. Let $L = \sum_g \min\{0, \min_r(m_r^g)\}$ (the lowest possible aggregate robustness measure for an agent from any subset of the games), $U = \sum_g \max\{0, \max_r(m_r^g)\}$ (the highest), and $X = U - L$. Also, let P_g be a (size $r!$) set of vectors of length $|N|$, where each element is a permutation of $\{m_1^g, m_2^g, \ldots, m_{|N|}^g\}$. That is, P_g is the set of all the possible robustness measure combinations from game g.

The algorithm fills up a table of size $|G| \times X^n$ containing Boolean values, with the first axis of the table ranging from 1 to $|G|$, and the other n axes (one for each agent) ranging from L to U. The table entry $T(g, k_1, k_2, \ldots, k_n)$ represents whether it is possible for each player i to obtain an aggregate robustness measure of k_i from role assignments to the first g minigames only (arbitrarily labeling the games as $1, \ldots, |G|$). The rule in Figure 2 is used to fill the table.

1: $T(1, k_1, k_2, ..., k_n) = 1$ if and only if the vector $(k_1, k_2, ..., k_n) \in P_1$.
2: For $1 < g \leq |G|$, $T(g, k_1, k_2, ..., k_n) = 1$ if and only if for some $(h_1, h_2, ..., h_n) \in P_g$, $t(g-1, k_1 - h_1, k_2 - h_2, ..., k_n - h_n) = 1$.

Fig. 2. Dynamic program for ROLE-ASSIGNMENT.

The ROLE-ASSIGNMENT instance then has a solution if and only if the last row of the table (for $g = |G|$) has a 1 for an entry with nonnegative values for the other axes—i.e., $(\exists k_1, \ldots, k_n \geq 0)\ T(|G|, k_1, \ldots, k_n) = 1$. In this row we can also find the maxmin aggregate robustness level that the integer programming algorithm finds, i.e., $\max\{v : (\exists k_1, \ldots, k_n \geq v)\ T(|G|, k_1, \ldots, k_n) = 1\}$. Note that the amount of time required for these steps is dominated by the steps in Figure 2.

Theorem 4. *ROLE-ASSIGNMENT can be solved in pseudopolynomial time for a constant number of agents n.*

Proof. The table has $|G| \times X^n$ entries, and filling in an entry requires up to $n!$ lookups. Moreover, $X \leq |G| \cdot d$, where d is the maximum difference between two robustness values in a minigame, which itself is $O(\upsilon - \lambda)$ where υ (λ) is the largest (smallest) single payoff in a minigame. Hence, with constant n, the algorithm is polynomial in $|G|$ and $\upsilon - \lambda$. (Of course, the input size is polynomial in $|G|$ and $\log(\upsilon - \lambda)$, which is why the algorithm is only pseudopolynomial.)

8 Simulation Analysis

In this section, we evaluate the two algorithms on random instances. We use GAMUT [11] to generate instances. For a given number n of players, a given number $|G|$ of minigames, and a given game generator in GAMUT, we generate an instance by drawing $|G|$ n-player games with payoffs in the interval $[-5, 5]$ from the generator. Because the DP algorithm requires payoffs to be integers, we round all the payoffs in each game to integers in $\{-5, -4, \ldots, 5\}$. We evaluate the IP algorithm on nondiscretized payoffs. (When we do run it on the rounded payoffs, the IP algorithm is in fact even faster, and always returns the same solution as the DP.)

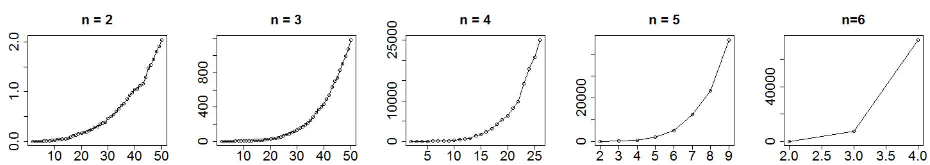

Fig. 3. The plot represents the average runtime of solving an instance of ROLE-ASSIGNMENT through dynamic programming. The x axis represents how many minigames are to be assigned ($|G|$), and the y-axis represents how long it took to solve the instances, in seconds. Each data point in the graph is an average of multiple instances (ranging from 2 to 200, due to cases such as $n = 6$, where we decide to timeout the program). While we studied many different game generators, as a representative case, we present the results for the case where G consists of uniformly random games.

Figure 3 shows the results for the DP algorithm. Predictably, the runtime of the DP algorithm closely tracks the number of table entries that need to be filled in ($|G| \times X^n$). The number of table entries blows up quickly when n increases.

Figure 4 shows the results for the IP algorithm, which scales much, much better. Occasionally, though, some instance is particularly unfavorable to the IP's performance. (See, for example, the case of having 10 minigames when

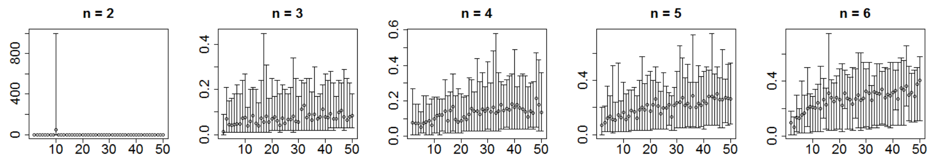

Fig. 4. The plot represents the average runtime of solving an instance of ROLE-ASSIGNMENT through integer programming. The x axis represents how many minigames are to be assigned ($|G|$), and the y-axis represents how long it took to solve the instances, in seconds. Each data point in the graph is an average of 200 instances. The top of the range bar indicates the maximum time an individual instance required to be solved, and the bottom of the range bar indicates the shortest. The graphs presented here are from uniformly random games with no integral payoff restrictions.

$n = 2$. There is at least one instance that takes almost 1000 seconds to solve. On average, even with this extreme outlier, the instances with that number of minigames are solved within 20 seconds).

9 Conclusion

In this paper, we have identified the problem of assigning roles to agents across multiple games in such a way that cooperative behavior becomes an equilibrium. We provided an easy-to-check necessary and sufficient condition for a given role assignment to induce cooperation and used this to obtain hardness results as well as algorithms for the problem of finding such a role assignment. Our integer programming algorithm significantly outperformed our dynamic programming algorithm in experiments, even though the latter is pseudopolynomial for constant numbers of agents.

We believe that there are many other important directions that can be studied in the context of game-theoretic role assignment. Our model can be extended to allow (perhaps costly) reassignment of roles as time progresses; different agent types that value roles differently, and preferences not only over roles but also over which type of agent one is matched with (providing connections to matching [9] and hedonic games [1]); side payments between agents (providing connections to matching with contracts [6]); not every minigame being played in each round; generalizing from repeated games to stochastic or arbitrary extensive-form games; and so on. We believe that our paper provides a good foundation for such follow-up work.

The availability of a pseudopolynomial-time algorithm when the number of agents is constant also suggests that there may be potential for approximation algorithms. However, note that the problems as we have defined them are decision problems, and it is not immediately obvious what the right optimization variant would be. One possibility may be to consider approximate equilibria.

References

1. Aziz, H., Savani, R.: Hedonic games. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A.D. (eds.) Handbook of Computational Social Choice, chap. 15. Cambridge University Press (2015)
2. Bernheim, B.D., Whinston, M.D.: Multimarket contact and collusive behavior. The Rand Journal of Economics 21(1), 1–26 (1990)
3. Borgs, C., Chayes, J., Immorlica, N., Kalai, A.T., Mirrokni, V., Papadimitriou, C.: The myth of the Folk Theorem. Games and Economic Behavior 70(1), 34–43 (2010)
4. Bulow, J.I., Geanakoplos, J.D., Klemperer, P.D.: Multimarket oligopoly: Strategic substitutes and complements. Journal of Political Economy 93(3), 488–511 (1985)
5. Folmer, H., von Mouche, P.: Linking of repeated games. when does it lead to more cooperation and pareto improvements? Working paper 60.2007, FEEM Fondazione Eni Enrico Mattei, C.so Magenta, 63, 20123 Milano, Italy (May 2007)
6. Hatfield, J.W., Milgrom, P.R.: Matching with contracts. American Economic Review 95(4), 913–935 (September 2005)
7. Jackson, M.O., Sonnenschein, H.F.: Overcoming incentive constraints by linking decisions. Econometrica 75(1), 241–257 (2007)
8. Just, R.E., Netanyahu, S.: The importance of structure in linking games. Agricultural Economics 24(1), 87–100 (2000)
9. Klaus, B., Manlove, D., Rossi, F.: Matching under preferences. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A.D. (eds.) Handbook of Computational Social Choice, chap. 14. Cambridge University Press (2015)
10. Kontogiannis, S.C., Spirakis, P.G.: Equilibrium points in fear of correlated threats. In: Proceedings of the Fourth Workshop on Internet and Network Economics (WINE). pp. 210–221. Shanghai, China (2008)
11. Nudelman, E., Wortman, J., Leyton-Brown, K., Shoham, Y.: Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In: Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). pp. 880–887. New York, NY, USA (2004)