Efficient Boolean Games Equilibria: A Scalable Approach (Extended Abstract)

Zohar Komarovsky
Ben-Gurion University
Be'er-Sheva, Israel
zoharkom@cs.bgu.ac.il

Vadim Levit
Ben-Gurion University
Be'er-Sheva, Israel
levitv@cs.bgu.ac.il

Tal Grinshpoun
Ariel University
Ariel, Israel
talgr@ariel.ac.il

Amnon Meisels
Ben-Gurion University
Be'er-Sheva, Israel
am@cs.bgu.ac.il

ABSTRACT

The present study focuses on a family of Boolean games whose agents' interactions are defined by a social network. The task of finding social-welfare-maximizing outcomes for such games is NP-hard. Moreover, such optimal outcomes are not necessarily stable. Therefore, our aim is to devise a procedure that finds stable outcomes with an as high as possible social welfare. To this end, we construct a quadratic-time procedure, by which any initial outcome of a game in this family can be transformed into a stable solution by the use of side payments. The resulting stable outcome is ensured to be at least as efficient as the initial outcome. Considering the fact that this procedure applies for any initial state, one may use good search heuristics to find an outcome of high social welfare, and then apply the procedure to it. This naturally leads to a scalable process that finds desirable efficient and stable solutions.

Keywords

Boolean games; side payments; efficient equilibria

1. INTRODUCTION

One of the major research challenges in Boolean games [1], as well as in game theory in general, is stabilization, i.e., securing the existence of a *pure-strategy Nash equilibrium* (PNE). However, a *stable* outcome from which no agent wants to unilaterally deviate, is not necessarily an *efficient* outcome. Herein, we relate to *social welfare*, which is a common notion of efficiency representing the sum of agents' utilities. In that sense, an outcome that maximizes social welfare is considered efficient.

In this study we address the problematic tradeoff between stability and efficiency in the context of a family of Boolean games, in which the interactions of agents are defined by some underlying social network. This family of Boolean games is inspired by the well-studied class of network games known as "best-shot" public goods games [2, 3]. There, the action that each agent takes (or avoids) is associated with an investment in some local public good (e.g., buying a book or some other product that is easily lent from one agent to another). Each agent wants that the action will be taken by at least one agent in its neighborhood, including itself. However, there is a cost associated with taking the action, so if any of its neighbors take the action then the agent prefers to avoid taking it.

Contemplating on the tradeoff between stability and efficiency, it is important to note that finding a social-welfare-maximizing outcome is equivalent in our settings to finding a minimal dominating set, which is known to be NP-hard. Moreover, such optimal solutions are not necessarily PNEs. Therefore, our objective is to reach efficient (yet not necessarily optimal) and stable solutions, while still remaining scalable. For this purpose we construct a quadratic-time procedure that works on any initial arbitrary outcome of the games at focus. We use the *side payments* mechanism [5] in order to ensure stability. However, not necessarily every outcome can be transformed to a stable state by the use of side payments [4]. Thus, we move, if needed, to a different outcome that *can* be stabilized using side payments. The new outcome is ensured to be at least as efficient as the original outcome. Finally, the new outcome is transformed, if needed, to a PNE by the use of side payments.

Considering the fact that the above procedure applies for any arbitrary state, one may use good search heuristics to find a state with some desirable properties. Using a run-time-efficient heuristic in the initial stage, combined with the quadratic-time procedure that follows, leads to a scalable process that finds desirable efficient and stable solutions.

2. PUBLIC GOODS BOOLEAN GAMES

A *public goods Boolean Game* (PGBG) is defined according to some underlying graph that describes the agents' interactions network. Each vertex represents an agent $i \in A$ in the corresponding Boolean game, and edges represent the interaction structure of the game. The set of i's neighbors is denoted by N_i; these are the agents whose actions may impact i's payoff (by enabling/disabling i's ability to achieve its personal goal). The *neighborhood* of i is the set $\{i\} \cup N_i$.

This study assumes that each agent i possesses a single Boolean variable $a_i \in \Phi$, which denotes i's chosen strategy: $a_i = \top$ (for performing an action), or $a_i = \bot$ (for avoiding it). The personal goal γ_i of agent i is a disjunction of all

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

Boolean variables possessed by agents in i's neighborhood. This definition of personal goal implies that each participant's primary aim is that at least one agent in its neighborhood will perform an action. Consequently, each agent can ensure the achievement of its personal goal by performing the action. The strategic nature of the game comes from the fact that performing the action involves costs. More formally, the cost function $c : \Phi \times \mathbb{B} \to \mathbb{R}_{\geq}$ defines a zero cost for choosing $\bot$ ($\forall i \in A : c(a_i, \bot) = 0$), and some positive cost $\mathcal{C}_i > 0$ for choosing $\top$ ($\forall i \in A : c(a_i, \top) = \mathcal{C}_i$).

Given an outcome of the game $v \in V$, the possible *states* of an agent $i \in A$, denoted by $State(i, v)$, can be divided into the next four distinct types:

1. $\mathbb{TT}$: $v(a_i) = \top$ and $\exists j \in N_i : v(a_j) = \top$
2. $\mathbb{TF}$: $v(a_i) = \top$ and $\forall j \in N_i : v(a_j) = \bot$
3. $\mathbb{FT}$: $v(a_i) = \bot$ and $\exists j \in N_i : v(a_j) = \top$
4. $\mathbb{FF}$: $v(a_i) = \bot$ and $\forall j \in N_i : v(a_j) = \bot$

We say that agent i *depends* on agent j under outcome v, denoted by $i \lessdot^v j$, if the achievement of i's personal goal depends only on j's choice at outcome v.

3. SIDE PAYMENTS

Side payments enable Boolean games to be transformed from the inside, by endowing agents with the possibility of sacrificing part of their payoff in order to convince other agents to play a certain strategy. This incentive mechanism inherently fits the PGBG scenario, since it is in the best interest of an agent to sacrifice part of its payoff in order to convince one of its neighbors in the network to take the action (invest in the public good). We adopt the Boolean transfer functions $\beta_i : V \times A \to \mathbb{R}_{\geq}$ of Turrini [5].

Since the use of side payments is mainly motivated by the attempt to secure a stable state (PNE) with certain properties, it is required to differentiate between outcomes that can be transformed to a PNE and those that can not.

Definition 1. An outcome v in a Boolean game G is *side payments enforceable* (SPE) if there exists a transfer function β, such that: $\forall i \in A, \forall v'_i \in V_i : u_i^\beta(v_i, v_{-i}) \geq u_i^\beta(v'_i, v_{-i})$, where $u_i^\beta(v)$ denotes the utility of agent i from outcome v under transfer function β.

4. STABILITY ENFORCING PROCEDURE

Starting from an initial outcome v, the first stage of the procedure finds an improved outcome v^* that is guaranteed to be SPE. The second stage applies the side payments mechanism to transform the improved outcome to a stable one. We restrict our attention to cases where all costs ($\mathcal{C}_i$) are identical, namely all costs equal $\mathcal{C}$.

First Stage. The first stage consists of two loops, each of which deals with one unstable state ($\mathbb{FF}$ or $\mathbb{TT}$). Starting at some outcome v, the first loop ensures that no agent will stay in state $\mathbb{FF}$, by changing the choice of each such agent from $\bot$ to $\top$. After the first loop, each agent is in one of three states: $\mathbb{TF}$, $\mathbb{FT}$ or $\mathbb{TT}$. The second loop ensures that each agent in state $\mathbb{TT}$ will have at least one other agent that depends on it, or otherwise the agent changes its choice from $\top$ to $\bot$. Such dependence enables compensating the agent by the use of side payments, and thus stability can be enforced. Applying these loops results in outcome v^*.

PROPOSITION 1. *For every outcome $v \in V$ it holds that v^* is side payments enforceable.*

PROPOSITION 2. *For every outcome $v \in V$, the utility of any agent in v^* is not lower than its utility in v.*

All proofs are omitted due to space limitation. Since Proposition 2 ensures that the utility of every agent is non-decreasing, then so is the sum of all agents' utilities.

COROLLARY 3. *For every outcome $v \in V$, the social welfare of v^* is greater than or equal to that of v.*

Second Stage. Here we apply a transfer function β that incentivizes only agents in state $\mathbb{TT}$ and only in outcome v^*:

$$\beta_j(v^*, i) := \begin{cases} \frac{\mathcal{C}}{d_i(v^*)}, & \text{if } State(i, v^*) = \mathbb{TT} \wedge j \lessdot^{v^*} i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $d_i(v^*)$ denotes the number of agents in A that depend on i in outcome v^*. Note that for all other outcomes $v' \neq v^*$ no payments are transferred (i.e., $\forall j, i \in A : \beta_j(v', i) = 0$).

PROPOSITION 4. *The stability enforcing procedure runs in quadratic time.*

5. CONCLUSIONS

We have introduced a stability enforcing procedure. Applying it on an initially efficient outcome, e.g., of maximal social welfare, results in an efficient equilibrium, as desired. However, finding an outcome of maximal social welfare is NP-hard, thus it is applicable only for relatively small networks. A more widely applicable approach would be to use a heuristic in order to find an initial outcome v with relatively high (yet not necessarily optimal) social welfare, and then apply the stability enforcing procedure to it. Using a run-time-efficient heuristic, combined with the quadratic run-time complexity of the procedure (Proposition 4), results in a scalable process for securing an efficient equilibrium. To conclude, the process consists of the following stages:

1. Applying a heuristic that returns an initial outcome v.
2. Transforming v to v^* according to the first stage.
3. Securing stability using the β transfers of Equation 1.

Acknowledgments

This study was supported by the Lynn and William Frankel center for Computer Sciences.

REFERENCES

[1] P. Harrenstein, W. van der Hoek, J.-J. Meyer, and C. Witteveen. Boolean games. In *TARK*, pages 287–298, 2001.

[2] J. Hirshleifer. From weakest-link to best-shot: The voluntary provision of public goods. *Public choice*, 41(3):371–386, 1983.

[3] M. O. Jackson and Y. Zenou. Games on networks. *Handbook of game theory*, 4, 2014.

[4] V. Levit, Z. Komarovsky, T. Grinshpoun, and A. Meisels. Tradeoffs between incentive mechanisms in Boolean games. In *IJCAI*, pages 68–74, 2015.

[5] P. Turrini. Endogenous Boolean games. In *IJCAI*, pages 390–396, 2013.