

Boolean Games: Inferring Agents’ Goals Using Taxation Queries

Abhijin Adiga¹, Sarit Kraus², Oleg Maksimov² and S. S. Ravi¹

¹Biocomplexity Institute and Initiative, University of Virginia, Charlottesville, VA, USA

²Department of Computer Science Bar Ilan University, Ramat Gan, Israel

{abhijin, ssravi}@virginia.edu, sarit@cs.biu.ac.il, oleg@maksimov.co.il

Abstract

In Boolean games, each agent controls a set of Boolean variables and has a goal represented by a propositional formula. We study inference problems in Boolean games assuming the presence of a PRINCIPAL who has the ability to control the agents and impose taxation schemes. Previous work used taxation schemes to guide a game towards certain equilibria. We present algorithms that show how taxation schemes can also be used to infer agents’ goals. We present experimental results to demonstrate the efficacy our algorithms. We also consider goal inference when only limited information is available in response to a query.

1 Introduction

Boolean games [Harrenstein *et al.*, 2001] are a class of games where agents’ goals are represented by a propositional logic formula. Each agent i controls a distinct set of Boolean variables Φ_i , and there is a cost associated with each assignment. Its formula or goal γ_i is composed of variables (set Γ_i) that are not necessarily in its control. Each agent’s first priority is to achieve its goal and its second priority is to minimize its total cost. Much of the work on Boolean games is of theoretical nature [Harrenstein *et al.*, 2001; Sauro and Villata, 2013; Wooldridge *et al.*, 2013; Grant *et al.*, 2014; Ågotnes *et al.*, 2013; Bonzon *et al.*, 2007; Levit *et al.*, 2013b]. Boolean games have been used to model some real-world problems such as charging electric vehicles and traffic signalling [Levit *et al.*, 2013b].

Recently, a number of works have emerged where a user either actively queries the system or uses passive observations to infer the functions in multi-agent systems [Kleinberg *et al.*, 2017; Adiga *et al.*, 2018; Narasimhan *et al.*, 2015; He *et al.*, 2016]. Wooldridge *et al.* [2013] introduced the notion of a PRINCIPAL – an external agent– who can influence the agents’ decisions through *taxation schemes* (additional costs for assigning values to variables) to achieve a desirable equilibrium. Under this framework, we study the inference problem where the PRINCIPAL’s objective is to infer the agents’ goals by repeatedly “querying” the system and observing the outcomes.

It is known that pure Nash equilibria (PNE) need not exist for every taxation scheme or there may be multiple PNE [Levit *et al.*, 2013b]. Such scenarios might make it impossible to infer some or all the goals. To overcome this problem, we allow the PRINCIPAL to *inhibit* some agents from achieving their goals. A *taxation query* (or simply a query) specifies (i) a subset of inhibited agents and (ii) an unambiguous taxation scheme (where costs for setting a variable to 0 and 1 are not equal). While this framework seems to provide an unrealistic amount of control to the PRINCIPAL, we show scenarios where without such control inference might not be possible. Our approach is to strategically inhibit some agents and infer the goals of uninhibited agents by simultaneously querying them using taxation schemes. To this end, we construct a graph representing dependencies between agents’ goals and apply vertex coloring to that graph. Then, for each query, the PRINCIPAL observes a PNE. The questions of primary interest here are: Does there exist a set of queries so that the PRINCIPAL can infer all goals? If yes, what is its size?

1.1 Summary of Results

Necessary and sufficient conditions for the existence of an NE for a taxation query. Using an undirected graph (called the goal overlap graph) that captures the overlaps between the sets of variables used in agents’ goals, we establish necessary and sufficient conditions for the existence of an NE for a Boolean game and any $\{0,1\}$ -taxation query, i.e., a query with only 0 and 1 costs (Section 3).

Evaluating goals of selected agents at an NE. When there is an NE for a $\{0,1\}$ -taxation query, we show that the value of the goals of selected agents for the zero cost assignment (i.e., the assignment whose cost is zero for a selected agent) can be determined regardless of which NE is reached by the agents (Section 3). We note that the results in this and the previous paragraph hold for any unambiguous taxation query. We use $\{0,1\}$ -taxation queries for convenience.

Goal inference algorithms. Using the results in the above two paragraphs, we show that taxation queries can be used to infer agents’ goals. Our taxation queries play a role similar to membership queries used to learn Boolean functions (e.g., [Abasi *et al.*, 2014; Angluin and Slonim, 1994]). Further, using a valid node coloring of the goal overlap graph, we show that the goals of many agents can be inferred simultaneously. We point out that the coloring-based scheme can

be significantly better than inferring the goals one at a time. We also obtain more efficient (in terms of the number of taxation queries) goal inference algorithms for two special classes of goal functions, namely threshold functions and symmetric functions (Section 4).

Experimental results. We show experimentally that our coloring-based inference algorithm uses significantly fewer queries compared to inferring goals one at a time. Further, the coloring-based approach also uses significantly less time even for games with 36,000 agents (Section 5).

Inference with limited information. We show that goal inference is possible in a context where the response to a query only provides the values assigned to the variables and not whether agents achieved their goals. In particular, we discuss this result when each goal is a threshold function (Section 6).

For space reasons, most proofs are sketched or omitted; detailed proofs can be found in [Adiga *et al.*, 2020].

1.2 Related Work

Harrenstein *et al.* [2001] introduced Boolean games as a class of two-player games and Bonzon *et al.* [2007] generalized the framework to n players. Structural and computational properties of PNE in Boolean games are well-studied. Bonzon *et al.* [2007] define a “dependency graph” between players to characterize PNE in Boolean games, much like our goal overlap graph. Levit *et al.* [2019; 2013b] study methods for finding a taxation scheme that incentivizes the agents to reach a stable state. Also, Levit *et al.* [2013a] discuss an application of Boolean games to the charging of electric vehicles where some vehicles are not allowed to charge at certain time intervals to avoid overloading. This is similar to our notion of the PRINCIPAL inhibiting agents. Boolean games where players have incomplete information about each other’s goals have also been considered in the literature (e.g., [Clercq *et al.*, 2015; Ågotnes *et al.*, 2013]).

2 Definitions and Preliminaries

Components of a Boolean Game. We follow [Wooldridge *et al.*, 2013] in defining a Boolean game. Let $V = \{1, 2, \dots, n\}$ be a collection of n agents. Let Φ be a finite set of Boolean variables. γ_i is the Boolean function (a propositional formula over the variables in Φ) that represents the goal of agent i ($1 \leq i \leq n$). $\Gamma_i \subseteq \Phi$ denotes the set of variables in γ_i . Φ is assumed to be partitioned into n sets $\Phi_1, \Phi_2, \dots, \Phi_n$, where Φ_i denotes the subset of variables controlled by agent i ($1 \leq i \leq n$). Some of the subsets in the collection may be empty; the corresponding agents do not control any of the variables in Φ . Since the collection of subsets of Φ is pairwise disjoint, each variable in Φ is controlled by exactly one agent. We use \mathbb{B} to denote the set $\{0, 1\}$.

In addition to the set of agents V , it is assumed that there is a special agent (external to the game) called the PRINCIPAL. There is a special Boolean variable, called the **inhibitor variable**, ψ_i for each agent i , $1 \leq i \leq n$. (The inhibitor variables are *not* in Φ .) For any agent i , the PRINCIPAL can set ψ_i to 0 to *inhibit* agent i . The behaviors of inhibited and uninhibited agents are discussed later in this section.

Taxation queries and Nash equilibrium. The PRINCIPAL can influence the goal of any agent i in two ways. Firstly, it can set the inhibitor variable ψ_i . Secondly, as discussed by Wooldridge *et al.* [2013], it can choose a **taxation scheme** τ that specifies the tax values, denoted by $\tau(x = 0)$ and $\tau(x = 1)$, for setting each variable x in Φ to 0 and 1 respectively. While our results hold for any unambiguous taxation scheme (where assignments 0 and 1 have different costs), for convenience, we will use the $\{0, 1\}$ -taxation scheme defined by Wooldridge *et al.* [2013]. A taxation scheme τ is a **$\{0, 1\}$ -taxation scheme** if for all $x \in \Phi$ and $b \in \mathbb{B}$, (i) $\tau(x = b) \in \{0, 1\}$ and (ii) $\tau(x = b) = 1 - \tau(x = -b)$.

Our algorithms for goal inference rely on **taxation queries**. Formally, a **taxation query** q is a pair (\mathbb{I}, τ) , where \mathbb{I} is the **inhibition vector** that specifies the value of the inhibitor variable ψ_i for each agent i and τ is the taxation scheme. For each taxation query $q = (\mathbb{I}, \tau)$, the response of an agent i is to set each variable in Φ_i to 0 or 1. The **cost** for agent i is the sum of the tax values over all the variables in Φ_i . Each agent i sets the variables in Φ_i in the following manner.

(a) If agent i is inhibited by the PRINCIPAL, then the agent sets each variable in Φ_i to the value with the lower tax.

(b) If agent i is *not* inhibited by the PRINCIPAL, then it proceeds as follows.

(i) Agent i always prefers assignments to the variables in Φ_i that satisfy the goal γ_i over those which don’t satisfy the goal (given other agents’ assignments).

(ii) When there are two or more assignments that satisfy the goal, the agent prefers one with the minimum cost. (If there are two or more minimum cost assignments that satisfy the goal, the agent chooses one of them arbitrarily.)

(iii) If none of the assignments to the variables in Φ_i satisfies the goal γ_i , then agent i chooses a (non-satisfying) assignment with the minimum cost.

Definition (Nash Equilibrium). An assignment to the variables in Φ is a pure **Nash Equilibrium** (NE) if no agent i has an incentive to *unilaterally* change the assignment it has chosen for the variables in Φ_i .

Response to a taxation query. For each taxation query q , if the agents reach an NE, the PRINCIPAL receives a response consisting of two items: (i) for each uninhibited agent i , whether i achieved its goal and (ii) the value assigned to each variable $x \in \Phi$. If there are two or more equilibria for a taxation query, the agents may choose any one of them. If there is no NE, the system produces the response “No equilibrium” and does not provide any other information.

We now present examples to illustrate the above concepts.

Example 1. Consider a Boolean game with two agents ($\Phi = \{x, y\}$), and $\Phi_1 = \{x\}$ and $\Phi_2 = \{y\}$. Let $\gamma_1 = x \vee y$ and $\gamma_2 = x \oplus y$, where ‘ \oplus ’ is the exclusive-or operator. Let the PRINCIPAL choose the following $\{0, 1\}$ -taxation scheme τ : $\tau(x = 1) = \tau(y = 1) = 0$. (Thus, $\tau(x = 0) = \tau(y = 0) = 1$.) We consider two inhibition vectors with the taxation scheme τ to show that they may lead to different NE.

(i) Suppose the PRINCIPAL does not inhibit either of the agents. The assignment $x = y = 0$ is not an NE for this setting since agent 1 has the incentive to change x to 1 to achieve its goal. Likewise, neither of the two assignments

$(x = 0, y = 1)$ and $(x = 1, y = 1)$ is an NE; in the former, agent 1 has an incentive to change x to 1 to reduce its cost while in the latter, agent 2 has an incentive to change y to 0 to achieve its goal. However, the assignment $(x = 1, y = 0)$ is an NE for this setting.

(ii) Suppose the PRINCIPAL inhibits only agent 2. Since agent 2 is inhibited, it sets y to 1 as that choice makes the cost of agent 2 to be zero. Given agent 2's choice for y , agent 1 achieves its goal by setting $x = 1$ since that choice incurs the minimum cost (namely, zero). Thus, the only NE for this situation is $(x = 1, y = 1)$, which is different from the one where neither agent is inhibited. \square

Example 2: The purpose of this example is to show that there are Boolean games which have no NE for some taxation schemes. The three agent game is defined in Table 1. Suppose

Agent	Goal	Control set
1	$x_1 \vee x_3$	$\Phi_1 = \{x_1\}$
2	$x_1 \vee x_2$	$\Phi_2 = \{x_2\}$
3	$x_2 \vee x_3$	$\Phi_3 = \{x_3\}$

Table 1. A game without a NE for some taxation schemes.

the PRINCIPAL chooses the taxation query q which does *not* inhibit any agent and uses the following taxation scheme: for each variable x_i , $\tau(x_i = 0) = 0$ and $\tau(x_i = 1) = 1$, $1 \leq i \leq 3$. The reader can verify that none of the 8 possible choices of values for the three variables is an NE. Thus, there is no NE for this example. \square

Statement of the Goal Inference Problem. The PRINCIPAL knows the number of agents n and for each agent i , the sets Φ_i and Γ_i (i.e., the set of variables used in the goal γ_i). The PRINCIPAL's objective is to infer the goal function γ_i of each agent i . (a) To that end, the PRINCIPAL issues taxation queries to the system a number of times. (b) For each query, the principal receives a response as described earlier. (c) The PRINCIPAL should use as few queries as possible to infer all the goals.

Challenges in developing goal inference algorithms. Algorithms to infer agents' goals must deal with two primary challenges. First, such an algorithm must avoid using queries for which there is no NE. Second, for a chosen taxation query, there may be multiple equilibria and the agents may arbitrarily choose one of them. Thus, the algorithm must be able to extract useful information regardless of which NE is chosen. We now present an example to illustrate this situation.

Example 3. This example shows that a Boolean game may have multiple equilibria. The game has two agents and $\Phi = \{x_1, x_2\}$. Each agent i controls x_i . Their goals are identical: $\gamma_1 = \gamma_2 = x_1 \leftrightarrow x_2$. Suppose both agents are uninhibited. It is easy to see that for any taxation scheme, $x_1 = x_2 = 0$ and $x_1 = x_2 = 1$ are both Nash equilibria. In both cases, each agent's goal is satisfied. Further, if agent i changes the value of x_i , then it will not achieve its goal. For every taxation scheme, if the agents choose $x_1 = x_2 = 0$ as the response, then the PRINCIPAL will never know the response for other assignments. Thus, an inference algorithm must carefully choose inhibition vectors. \square

2.1 Additional Definitions

Definition 1 (Zero cost assignment). *Given a $\{0,1\}$ -taxation scheme τ and agent i , the **zero cost assignment** with respect to a set $X_i \subseteq \Phi_i$ of variables is obtained by assigning for each $x \in X_i$ the value $b \in \mathbb{B}$ such that $\tau(x = b) = 0$. (Thus, the total cost of this assignment over all the variables in X_i is zero.)*

The following definition of a *goal overlap graph* plays an important role in our inference methods.

Definition 2 (Goal overlap graph). *For any Boolean game, the **goal overlap graph** $G_O(V, E)$ is an undirected graph constructed as follows. Each node in V represents an agent. For any two distinct agents i and j , if $\Gamma_i \cap \Gamma_j \neq \emptyset$, then E contains the edge $\{i, j\}$.*

We assume that the reader is familiar with graph theoretic concepts such as independent set and (node) coloring (see e.g., [West, 2003]). The smallest integer k for which a graph G is k -colorable is called its **chromatic number** $\chi(G)$. In any valid k -coloring, the set of nodes which are assigned the same color constitute a **color class**. Each color class induces an independent set in G . Thus, a valid k -coloring of G partitions the node set V into k independent sets.

A Preliminary Lemma. A useful property of $\{0,1\}$ -taxation queries for which there is an NE is shown below. (The proof is omitted due to space reasons.)

Lemma 1. (Control of variables using taxation queries) *Suppose there is at least one NE for a $\{0,1\}$ -taxation query. Then, the following statements hold for every agent i in every NE. (1) If agent i is inhibited, then i chooses the zero cost assignment with respect to the set of variables Φ_i . (2) If agent i is uninhibited, then i chooses the zero cost assignment with respect to the set of variables $\Phi_i - \Gamma_i$.*

3 Existence of Equilibria Under Taxation Schemes with Inhibition

In this section, we establish necessary and sufficient conditions for the existence of Nash equilibria under taxation schemes that inhibit certain agents.

A sufficient condition for NE. Our sufficient condition can be stated formally as follows.

Theorem 1. *Consider a Boolean game with goal overlap graph $G_O(V, E)$. Let $W \subseteq V$ be any nonempty independent set of G_O . Let an agent be uninhibited if and only if it belongs to W . For every $\{0,1\}$ -taxation scheme τ under this setting, the following results hold.*

- (Existence)** *There exists an NE.*
- (Consistency)** *In all the NE, the outcomes are identical; that is, for any agent i and any pair of NE \mathcal{E}_1 and \mathcal{E}_2 , agent i achieves its goal in both \mathcal{E}_1 and \mathcal{E}_2 or does not achieve its goal in both \mathcal{E}_1 and \mathcal{E}_2 .*
- (Ability to evaluate)** *For every agent $i \in W$, it is possible to evaluate whether its goal is satisfied by the zero cost assignment with respect to the set of variables Γ_i .*

Proof (sketch). For space reasons, we will only sketch the proof for Part 1 of the theorem. We construct an assignment α of 0/1 values to the variables in Φ and prove that it corresponds to an NE. First, for any inhibited agent i (i.e., $i \in V - W$), we choose the zero cost assignment with respect to Φ_i . Now, consider each uninhibited agent i (i.e., $i \in W$), and let i choose the zero cost assignment with respect to the variables in $\Phi_i - \Gamma_i$. It can be shown that for each agent i , such an assignment sets each variable $x \in \Gamma_i - \Phi_i$ (which are controlled by agents other than i) to the value of zero cost. Now, agent i considers each combination of values to the variables in $\Phi_i \cap \Gamma_i$. If none of these combinations satisfies the goal γ_i , it chooses the zero cost assignment with respect to $\Phi_i \cap \Gamma_i$. Otherwise, it chooses an assignment that satisfies its goal and has the lowest cost among all such assignments. It can be shown that this assignment is an NE. ■

A necessary condition for NE. In proving Part 1 of Theorem 1, we assumed that the uninhibited agents form an independent set in the goal overlap graph and that some agents are inhibited. Here, we point out that these two assumptions are necessary to ensure the existence of an NE.

Proposition 1. *There are Boolean games that satisfy the following properties: (i) If the set of uninhibited agents do not form an independent set in the goal overlap graph, then there is no NE for any taxation scheme. (ii) If no agent is inhibited, then there is no NE for any taxation scheme.*

Proof sketch. We use the same Boolean game to prove both parts of this result. Consider a game with two agents denoted by 1 and 2. Let $\Phi = \{p, q\}$ denote the set of variables. Further, let $\Phi_1 = \{p\}$ and $\Phi_2 = \{q\}$. Let γ_1 be given by $p \leftrightarrow q$ and γ_2 be given by $\neg(p \leftrightarrow q)$. Thus, $\Gamma_1 = \Gamma_2 = \{p, q\}$. Assume that neither of the agents is inhibited. It can be seen that the set of agents $\{1, 2\}$ does not form an independent set. A simple case analysis can be used to argue that regardless of the taxation scheme, there is no NE for this system. The result of Part (ii) also follows since in the above example neither of the agents is inhibited. ■

Proposition 1 points out that there are games in which if the uninhibited agents do not form an independent set in the goal overlap graph, no taxation query has an NE. Therefore, taxation queries in which uninhibited agents do not form an independent set may be wasteful. Moreover, Theorem 1 points out that for any game and taxation scheme, if the uninhibited agents form an independent set, then every taxation query has an NE. Accordingly, our goal inference algorithms choose such taxation queries.

Ability to query many agents simultaneously. Our algorithms reduce the number of taxation queries by querying many agents simultaneously. The following lemma shows how this can be done when the uninhibited agents form an independent set in the goal overlap graph. (The proof is omitted for space reasons.)

Lemma 2. (Simultaneous querying) *Consider a Boolean game with goal overlap graph $G_O(V, E)$. Further, let $W \subseteq V$ be a nonempty independent set in G_O . Let an agent be uninhibited if and only if it belongs to W . There exists a $\{0,1\}$ -taxation scheme τ such that for every agent $i \in W$ and any*

given assignment g_i for the variables in Γ_i , one can decide whether g_i satisfies agent i 's goal γ_i using τ .

4 Goal Inference Algorithms

Suppose \mathcal{A} is an algorithm that learns a Boolean function f of r variables using an oracle in the following manner. For each input α to f , suppose the oracle produces the value $f(\alpha)$. Algorithm \mathcal{A} infers the function f from the values provided by the oracle. We now explain how a Boolean game can simulate an oracle that provides the value $f(\alpha)$ for each assignment α to the variables of f .

Consider the Boolean game where there is only one agent whose goal f is unknown. The variables of f are the variables in the game and the single agent controls all the variables. For this game, any input α to the function f can be converted into a $\{0,1\}$ -taxation query $q = (\mathbb{I}, \tau)$ as follows. The inhibition vector \mathbb{I} of q does not inhibit the agent. If α sets a variable x to the Boolean value b , then the $\{0,1\}$ -taxation scheme τ sets $\tau(x = b) = 0$. This ensures that α is the zero cost assignment under this taxation scheme. Since the uninhibited set consisting of a single agent is trivially an independent set, by Theorem 1, there is at least one NE for the taxation query q . Moreover, by Part (3) of Theorem 1, no matter which NE is reached, we can obtain the value of the goal function f for the zero cost assignment α . Thus, the Boolean game simulates the oracle that provides the value $f(\alpha)$ for the input α . In this manner, the Boolean game in conjunction with Algorithm \mathcal{A} can be used to infer the function f . Thus:

Proposition 2. *If there is an algorithm to learn a Boolean function f using β queries to an oracle, then there is an algorithm that infers the goal function f using β $\{0,1\}$ -taxation queries.* ■

Simultaneous inference of multiple goals. Suppose we have a set of m agents in a Boolean game such that the goal overlap graph for these agents is an independent set. Suppose for each goal function γ_i , we have an algorithm that can learn γ_i using q_i queries to an oracle. As discussed above, each of these queries can be translated into a $\{0,1\}$ -taxation query. Further, using Lemma 2, the $\{0,1\}$ -taxation queries for the m agents can be combined into a single $\{0,1\}$ -taxation query for the Boolean game. Since the agents form an independent set in the goal overlap graph, by Part (3) of Theorem 1, we can obtain the values of all the goal functions from the single taxation query. Thus, if $q^* = \max\{q_1, q_2, \dots, q_m\}$, then the number of $\{0,1\}$ -taxation queries used to infer all the m goal functions is q^* . This is an improvement over the simple method of inferring goals one at a time, for which the number of $\{0,1\}$ -taxation queries used is $\sum_{i=1}^m q_i$; in the worst-case, this sum can be as large as mq^* .

We can extend the above idea of simultaneous inference to an arbitrary Boolean game as follows. Let G_O denote the goal overlap graph for the game. Let χ denote the minimum number of colors needed to obtain a node coloring of G_O . Recall from Section 2 that each of the χ color classes forms an independent set. As discussed above, we can infer the goals of all the agents in a color class simultaneously. If q^* denotes the maximum number of $\{0,1\}$ -taxation queries used to infer the goal of any agent, then the number of $\{0,1\}$ -taxation queries

used by this coloring-based method is at most χq^* . We state this result formally below.

Proposition 3. *Let G_O denote the goal overlap graph for a Boolean game and let χ denote the chromatic number of G_O . Let q^* denotes the maximum number of $\{0,1\}$ -taxation queries used to infer the goal of any agent. Then, the number of $\{0,1\}$ -taxation queries used to infer the goals of all the agents is at most χq^* . ■*

Computing the chromatic number of an undirected graph is an NP-hard problem [Garey and Johnson, 1979]. In practice, one can use heuristics to obtain node colorings with a small number of colors. For example, any graph whose maximum node degree is Δ can be efficiently colored using $\Delta + 1$ colors using Brooks' theorem [West, 2003]. Such a method is useful when the maximum node degree is small.

Special classes of Boolean functions. Here, we present inference algorithms when the goals of all the agents are from special classes of Boolean functions. For these classes, the goal inference algorithms use only a small number of $\{0,1\}$ -taxation queries. In particular, we consider two special classes, namely **threshold functions** and **symmetric functions**. We begin with the definitions of these classes of functions (see, e.g., [Kohavi, 1970; Crama and Hammer, 2011]). For any Boolean function with Boolean inputs, the number of 1's in the input is called the **Hamming weight** of that input.

Definition 3. *Let $f(x_1, x_2, \dots, x_r)$ be a Boolean function of r Boolean variables. Function f is **symmetric** if the value of f depends only on the Hamming weight of the input.*

Example 4. Consider the Boolean function $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$, where \oplus is the exclusive-or operator. This function is symmetric since it has the value 1 iff the Hamming weight of the input is *odd*. □

Definition 4. *Let $f(x_1, x_2, \dots, x_r)$ be a Boolean function of r Boolean variables. For each integer $k \geq 0$, f is a **k -threshold function** iff the value of f is 1 when the Hamming weight of the input is at least k .*

Thus, any k -threshold function is also a symmetric function. We note that the 0-threshold function is the constant function which has the value 1 for all inputs. If a function f with r inputs is the constant function which has the value 0 for all inputs, we will regard f as the $(r + 1)$ -threshold function. For space reasons, we will discuss only the inference algorithm for threshold functions; a similar algorithm can be designed for symmetric functions.

Inferring threshold functions. Suppose the goal of each agent i in a Boolean game is the k_i -threshold function for some unknown integer k_i , $0 \leq k_i \leq |\Gamma_i| + 1$. The objective of the inference problem in this case is to determine the value of k_i for each agent a_i , $1 \leq i \leq n$. The algorithm that learns the threshold value of a single agent through queries uses a simple binary search over the range $[0 .. r + 1]$, where r is the number of inputs to the function; thus, the number of queries used is $O(\log r)$. Each query specifies an input with a certain Hamming weight, determined by the binary search procedure. Combining this binary search algorithm with the coloring of G_O and simultaneous inference of thresholds of

Algorithm 1: Inference algorithm for threshold functions.

- 1 Construct the goal overlap graph $G_O(V, E)$.
 - 2 Obtain an optimal node coloring for G_O . Let χ denote the number of colors used and let V_1, V_2, \dots, V_χ denote the color classes themselves.
 - 3 **for** $j = 1$ to χ **do**
 - 4 Inhibit every agent in $V - V_j$.
 - 5 Using a simultaneous binary search over the agents in V_j , get the threshold of each agent in V_j .
 - 6 **end**
 - 7 Output the threshold value of each agent.
-

all the agents in a color class (as discussed above), we obtain the inference procedure which is shown in pseudocode form as Algorithm 1.

Let Γ^* denote a set of maximum cardinality among $\Gamma_1, \Gamma_2, \dots, \Gamma_n$. Thus, the maximum number of taxation queries used for any color class is $O(\log(|\Gamma^*|))$. As a consequence, the number of taxation queries used to infer all the threshold values is $O(\chi \log(|\Gamma^*|))$. We state this result formally below.

Proposition 4. *Suppose the goal of each of the n agents in a Boolean game is a threshold function. Let χ denote the chromatic number of the goal overlap graph and let Γ^* denote a set of maximum cardinality among $\Gamma_1, \Gamma_2, \dots, \Gamma_n$. The threshold values of all the agents can be inferred using $O(\chi \log(|\Gamma^*|))$ taxation queries. ■*

5 Experimental Results

The aim of the experiments is to study two aspects: (a) how the coloring idea reduces the number of queries (compared to inferring the goals one node at a time) and (b) scalability of the approach. Our experimental procedures and results discussed below are for goals which are threshold functions. We use SEQ to denote the (sequential) querying algorithm that infers goals one node at a time and CBQ to denote the coloring-based querying algorithm.

Part (a): Effectiveness of querying using node coloring.

We ran an extensive set of experiments creating a large number of networks and agents' goals by varying the number of nodes, the minimum degree, the number of variables in an agent's goal controlled by the agent and by others and the threshold. In each network, nodes represent agents and the network itself represents the goal overlap graph. We ensured this by creating for each edge $\{u, v\}$, a variable $x_{u,v}$ that appears *only* in the goals of u and v . To make the game general, we also created other variables and generated control sets for nodes so that for each node (agent) i , the sets Φ_i and Γ_i have a nonempty intersection. For each agent i , we generated a random threshold value in the range $[1 .. |\Gamma_i|]$.

Let us first consider the SEQ approach which infers thresholds one node at a time. For each node i , our program finds its threshold by inhibiting all other nodes and constructing $\{0,1\}$ -taxation queries to do binary search over the possible threshold values of i . Thus, we can compute the total number of queries used by SEQ. Next, we generated a Brooks coloring of the graph using the simple greedy algorithm [West,

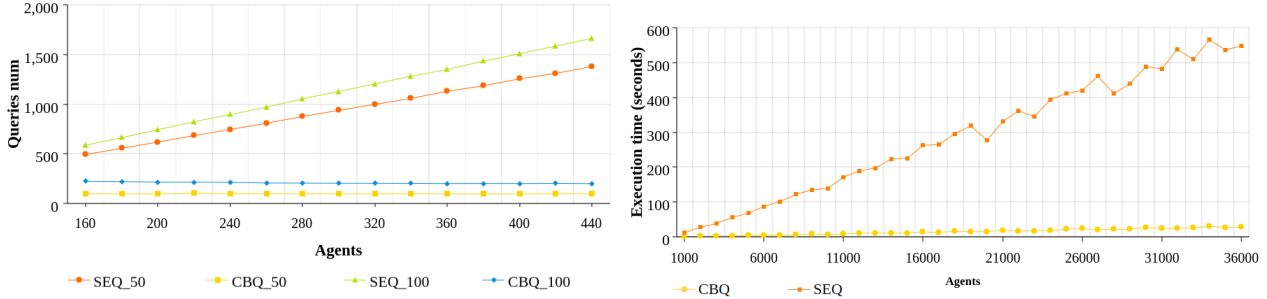


Figure 1. (a) The number of queries as a function of the number of agents for the SEQ algorithm (red and green lines) vs CBQ algorithm (yellow and blue lines) for graphs with minimum node degree of 50 (red and yellow) and minimum degree of 100 (green and blue). (b) Time in seconds as a function of the number of agents for graphs with minimum node degree of 50.

2003]. This method uses $\Delta + 1$ colors, where Δ is the maximum node degree. Now, for each color class, we used queries that simultaneously find the threshold values of all the nodes in the color class. This allows us to compute the total number of queries used by CBQ. For space reasons, we only present a few results from our experiments. In Figure 1(a), we varied the number of agents between 100 and 440 and considered two values for the minimum node degree, namely 50 and 100. We report the number of queries for the two algorithms and the two minimum degree values. It is easy to see from Figure 1(a) that for the sequential algorithm (SEQ), the number of queries increases with the number of agents (red and green lines). However, in the coloring-based algorithm (CBQ), the number of queries depends on the number of color classes, which is closely related to the maximum degree and not to the number of agents (yellow and blue lines). It is important to note that in these experiments, the generation of the Brooks coloring of a graph took on average only 3298 microseconds for graphs with minimum degree of 50 and 5935 microseconds for graphs with minimum degree of 100.

Part (b): Scalability of the coloring-based approach. In Figure 1(b) we demonstrate the scalability of CBQ. We varied the number of nodes between 1,000 and 36,000 and report the computation time for graphs with minimum degree of 50 and under the assumption that there are 20 processors that can be used for running the searches of the same color class, in parallel. It can be seen that the computation time of CBQ increases only at a low rate with the number of agents, while the time for the SEQ algorithm increases at a much higher rate.

6 Inference with Limited Information

Here, we observe that inference is possible for some special classes of goals even when the agents don't indicate whether they achieved their goals. For space reasons, we discuss only an algorithm for inferring thresholds. A similar algorithm can be designed when goals are conjunctions of literals.

Suppose the goal γ_i of each agent i is a threshold function and the threshold value θ_i of agent i is in the range $[1 \dots |\Gamma_i|]$. We also assume that of the $|\Gamma_i|$ variables used in γ_i , at least $q \geq 1$ variables are also in Φ_i ; that is, $q = |\Gamma_i \cap \Phi_i| \geq 1$. For simplicity, we will discuss the algorithm for inferring the threshold θ_i , assuming that all other agents are inhibited.

(The algorithm can be extended to simultaneous inference of goals using the idea of coloring.) The steps are as follows.

1. Issue a $\{0, 1\}$ -taxation query where $\tau(x = 1) = 1$ for all the variables. From the resulting assignment, find the number ℓ of variables in $\Gamma_i \cap \Phi_i$ which are set to 1. If $\ell \geq 1$, then we conclude that the threshold of agent i must be ℓ (since this is the minimum cost for achieving the goal γ_i) and stop.
2. (Here, the value of ℓ from Step 1 is zero; that is, $q < \theta_i \leq |\Gamma_i|$.) Consider the variables in $A_i = \Gamma_i - \Phi_i$. For each j , $1 \leq j \leq |A_i|$, let the $\{0, 1\}$ -taxation query \mathbb{I}_j be constructed as follows: (i) choose an arbitrary subset A' of j variables from A_i and (ii) for each variable x in A' let $\tau(x = 1) = 0$ and for all other variables, let $\tau(x = 1) = 1$. Using these queries, find the *smallest* j (using binary search) such that for the query \mathbb{I}_j , the response sets *all* the variables in $\Gamma_i \cap \Phi_i$ to 1. Then, it can be seen that the threshold value $\theta_i = q + j$.

When the response doesn't indicate whether agents achieved their goals, one cannot distinguish between the threshold value 0 (i.e., the goal function has the value 1 for all inputs) and threshold value ∞ (i.e., the goal function has the value 0 for all inputs). In both cases, regardless of the taxation query, the uninhibited agent i would set all the variables in Φ_i to the value of minimum cost and no useful conclusion can be drawn from the outcome.

7 Future Work

It will be useful to develop inference algorithms for other goal functions. While we established upper bounds on the number of queries used for inference, it will be of interest to develop appropriate lower bounds. Finally, it is also of interest to identify other classes of goal functions that can be inferred with limited information.

Acknowledgments

We thank the IJCAI 2020 reviewers for their comments. This work was partially supported by NSF Grants ACI-1443054 (DIBBS), IIS-1633028 (BIG DATA), CMMI-1745207 (EAGER), OAC-1916805, IIS-1908530 and by the Ministry of Science & Technology, Israel and the Ministry of Education, Science, Research and Sport of the Slovak Republic.

References

- [Abasi *et al.*, 2014] Hasan Abasi, Nader H. Bshouty, and Hanna Mazzawi. On exact learning monotone DNF from membership queries. *CoRR*, abs/1405.0792:1–16, 2014.
- [Adiga *et al.*, 2018] Abhijin Adiga, Chris J. Kuhlman, Madhav V. Marathe, S. S. Ravi, Daniel J. Rosenkrantz, and Richard E. Stearns. Learning the behavior of a dynamical system via a “20 questions” approach. In *Thirty second AAAI Conference on Artificial Intelligence*, pages 4630–4637, Palo Alto, CA, 2018. AAAI Press.
- [Adiga *et al.*, 2020] Abhijin Adiga, Sarit Kraus, Oleg Maksimov, and S. S. Ravi. Boolean Games: Inferring Agents’s Goals Using Taxation Queries. Tech Report, Biocomplexity Institute and Initiative, University of Virginia, Charlottesville, VA, USA, 2020.
- [Ågotnes *et al.*, 2013] Thomas Ågotnes, Paul Harrenstein, Wiebe Van Der Hoek, and Michael Wooldridge. Verifiable equilibria in Boolean games. In *Proc. of AAAI*, pages 689–695, 2013.
- [Angluin and Slonim, 1994] Dana Angluin and Donna K. Slonim. Randomly fallible teachers: Learning monotone DNF with an incomplete membership oracle. *Machine Learning*, 14(1):7–26, 1994.
- [Bonzon *et al.*, 2007] Elise Bonzon, Marie-Christine Lagasquie-Schiex, and Jérôme Lang. Dependencies between players in Boolean games. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 743–754, Heidelberg, Germany, 2007. Springer.
- [Clercq *et al.*, 2015] Sophie De Clercq, Steven Schockaert, Ann Nowé, and Martine De Cock. Multilateral negotiation in Boolean games with incomplete information using generalized possibilistic logic. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 2890–2896, Palo Alto, CA, 2015. AAAI Press.
- [Crama and Hammer, 2011] Yves Crama and Peter L. Hammer. *Boolean Functions: Theory, Algorithms, and Applications*. Cambridge University Press, New York, NY, 2011.
- [Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Co., San Francisco, CA, 1979.
- [Grant *et al.*, 2014] John Grant, Sarit Kraus, Michael Wooldridge, and Inon Zuckerman. Manipulating games by sharing information. *Studia Logica*, 102(2):267–295, 2014.
- [Harrenstein *et al.*, 2001] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer, and Cees Witteveen. Boolean games. In *Proceedings of the 8th conference on Theoretical aspects of rationality and knowledge*, pages 287–298, Burlington, MA, 2001. Morgan Kaufmann Publishers Inc.
- [He *et al.*, 2016] Xinran He, Ke Xu, David Kempe, and Yan Liu. Learning influence functions from incomplete observations. In *Advances in Neural Information Processing Systems*, pages 2073–2081, San Diego, CA, 2016. Neural Information Systems Processing Foundation.
- [Kleinberg *et al.*, 2017] Jon Kleinberg, Sendhil Mullainathan, and Johan Ugander. Comparison-based choices. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 127–144, New York, NY, 2017. ACM.
- [Kohavi, 1970] Zvi Kohavi. *Switching and Finite Automata Theory*. McGraw-Hill, New York, NY, 1970.
- [Levit *et al.*, 2013a] Vadim Levit, Tal Grinshpoun, and Amnon Meisels. Boolean games for charging electric vehicles. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02*, pages 86–93. IEEE Computer Society, 2013.
- [Levit *et al.*, 2013b] Vadim Levit, Tal Grinshpoun, Amnon Meisels, and Ana LC Bazzan. Taxation search in Boolean games. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 183–190, Richland, SC, USA, 2013. International Foundation for Autonomous Agents and Multiagent Systems.
- [Levit *et al.*, 2019] Vadim Levit, Zohar Komarovsky, Tal Grinshpoun, Ana LC Bazzan, and Amnon Meisels. Incentive-based search for equilibria in Boolean games. *Constraints*, 24:288–319, 2019.
- [Narasimhan *et al.*, 2015] Hari Krishna Narasimhan, David C. Parkes, and Yoran Singer. Learnability of influence in networks. In *Advances in Neural Information Processing Systems*, pages 3186–3194, San Diego, CA, 2015. Neural Information Systems Processing Foundation.
- [Sauro and Villata, 2013] Luigi Sauro and Serena Villata. Dependency in cooperative Boolean games. *Journal of Logic and Computation*, 23(2):425–444, 2013.
- [West, 2003] Douglas West. *Introduction to Graph Theory*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 2003.
- [Wooldridge *et al.*, 2013] Michael Wooldridge, Ulle Endriss, Sarit Kraus, and Jérôme Lang. Incentive engineering for Boolean games. *Artificial Intelligence*, 195:418–439, 2013.