

# Behavioural Strategies in Weighted Boolean Games

Dongge Han, Paul Harrenstein, Steven Nugent, Jonathan Philpott, Michael Wooldridge

*Department of Computer Science, University of Oxford, United Kingdom*

---

## Abstract

The adoption of game-theoretic models throughout artificial intelligence and computer science has prompted extensive research into algorithms and heuristics for computing game theoretic solution concepts, of which *mixed strategy Nash equilibrium* is one of the most prominent examples. This paper considers the issues surrounding the computation of mixed strategy Nash equilibria in *weighted Boolean games*: a natural, compact, and expressive class of games that has been widely studied in the artificial intelligence and multi-agent systems communities. In a weighted Boolean game, each player pursues the satisfaction of a weighted set of propositional logic goals by selecting a truth assignment for a set of propositional variables under its unique control; players aim to choose so as to maximise the total weight of formulas satisfied. Unfortunately, the obvious representation of a mixed strategy in weighted Boolean games is of size exponential in the number of variables a player controls. Weighted Boolean games, however, also allow for randomisation at the level of the propositional variables, which gives rise to a more compact model of randomised strategies, which we call *behavioural strategies*. We provide a detailed theoretical analysis of Nash equilibria in behavioural strategies. Two results are significant from an algorithmic point of view: (a) behavioural equilibria correspond to mixed and correlated equilibria that satisfy a specific independence property; and (b) they allow for exponentially fewer supports than mixed equilibria. These findings suggest two ways in which one can leverage existing algorithms and heuristics for computing mixed equilibria to find behavioural equilibria. The first is a

naive approach, in which we search for mixed equilibria and check whether they satisfy the aforesaid independence property. The second is more sophisticated and is based on support enumeration. In an additional third approach, which is inspired by the linear programming characterisation of correlated equilibria, we use numerical methods to find behavioural equilibria directly. In an extensive experimental study, we compare the performance of these approaches—with and without attendant heuristics—for finding some and all behavioural equilibria of a weighted Boolean game.

*Keywords:* Multi-agent Systems, Game Theory, Boolean games, Nash Equilibrium, Equilibrium Computation

---

## 1. Introduction

*Boolean games* are a compact and expressive class of logic-based games, which have attracted considerable attention within the multi-agent systems community [1, 2, 3, 4, 5, 6, 7]. In a Boolean game, each player  $i$  exercises unique  
5 control over a set of Boolean variables  $\Phi_i$ , and the pure strategies available to player  $i$  correspond to the set of valuations for  $\Phi_i$ , that is, the set of all possible assignments of truth or falsity to the variables in  $\Phi_i$ . Preferences in Boolean games are defined by associating each player  $i$  with a propositional logic formula  $\gamma_i$ , which that player desires to satisfy. When every player has selected a  
10 valuation for her variables, the resulting overall valuation determines whether or not each player’s goal is satisfied. Strategic considerations arise in Boolean games because the satisfaction of a player’s goal may depend on the choices made by other players. Boolean games are a natural model for studying strategic behaviour in multi-agent systems: the use of logical goals/specifications is common in the  
15 Artificial Intelligence (AI) and automated verification communities; and the fact that players act by assigning values to Boolean variables mirrors the behaviour of computer programs.

Our present work studies a generalisation of Boolean games, called *weighted Boolean games*. In such a game, each player is associated with a *goal set*  $\Gamma_i$

20 of weighted Boolean formulas of the form  $(\gamma, m)$ , where  $\gamma$  is a propositional logic formula and  $m \in \mathbb{R}$  represents  $i$ 's value for the formula  $\gamma$ . The utility a player obtains from a valuation is then simply the sum of weights associated with formulas that are satisfied by the valuation: each player aims to strategically choose a valuation so as to maximise this utility.

25 Research on Boolean games has focussed almost exclusively on pure strategy Nash equilibria. A basic result in this area is that checking for the existence of pure strategy Nash equilibria is  $\Sigma_2^P$ -complete [2]. A comprehensive collection of high intractability results for pure Nash equilibrium and related concepts in weighted Boolean games can be found in the work by Mavronicolas et al. [7].  
30 As in conventional non-cooperative games, pure strategy Nash equilibria are not guaranteed to exist in Boolean games. It is therefore natural to consider mixed strategy Nash equilibria for Boolean games, but what little research has been carried out on mixed strategies in Boolean games has pointed to very high computational complexity. For example, Ianovsky and Ong proved that  
35 the problem of checking whether there exists a mixed Nash equilibrium of a given Boolean game in which each player gets a given payoff is NEXPTIME-complete [8]. Note that a mixed strategy in a Boolean game is naturally defined as a probability distribution over the  $2^{|\Phi_i|}$  truth-value assignments (or valuations) that player  $i$  may make to its variables. The obvious representation of such mixed  
40 strategies involves explicitly listing a probability for each of these valuations. Unfortunately, this representation is at least of size exponential in the number of variables a player controls, which means it is clearly unrealistic as a practical representation scheme.

In this paper, we present and investigate an alternative representation of  
45 randomised strategies, which is specific for (weighted) Boolean games and which we refer to as *behavioural strategies*. The idea is that, rather than randomising at the level of valuations, with behavioural strategies we randomise at the level of propositional variables. Thus, instead of having to list a probability for all  $2^{|\Phi_i|}$  valuations for player  $i$ , we only need to list  $|\Phi_i|$  probabilities: one probability for  
50 each variable. There are several reasons for studying such behavioural strategies.

First, as we have already hinted, the behavioural strategy representation of mixed strategies is *exponentially more succinct than the naive representation*. If we are to use mixed strategies in Boolean games, then a succinct representation is *essential*: it would not be possible to use the naive representation for anything  
55 other than very small games. Of course, there are surely many other succinct representations possible, but given their naturalness, behavioural strategies are worth investigating.

Second, there are some situations in which behavioural strategies are the only feasible solution. Imagine a situation in which a player controls two variables. A  
60 regular mixed strategy would involve choosing a valuation for both these variables from a certain probability distribution. But this implies that the selection of values can be coordinated. If this is not possible—for example, if the variables correspond to situations where communication between them is impossible—then a behavioural strategy may be the only feasible way of randomising.

Our main contributions in the present paper are threefold. Our first contribution is to introduce this representation of mixed strategies (Section 2), and our second is to provide a detailed theoretical analysis of mixed strategies by behavioural strategies and the accompanying behavioural equilibrium concept (Section 3). We find that equilibria in behavioural strategies are not guaranteed  
70 to exist (Proposition 8) and that, even in the two-player case, the unique equilibrium may involve randomisation with irrational probabilities (Proposition 9). Our third contribution is the identification of three approaches to computing behavioural equilibria in two-player weighted Boolean games (Section 4) and an extensive experimental study in which we compare the performance of algo-  
75 rithms based on these three approaches, where we also come to consider various heuristics suitable for our algorithms (Section 5).

Two of our theoretical results in Section 3 lie at the basis of the algorithms presented in Section 4. We give a characterisation of Nash equilibria in behavioural strategies as equilibrium profiles of mixed strategies that satisfy a  
80 particular independence property we refer to as the *behavioural independence property* (*BIP*) (Proposition 4). As a corollary we find that a similar observation

holds for correlated equilibria (Corollary 7). Furthermore, we show that the number of supports of a behavioural strategy is polynomial in the set of outcomes, rather than exponential as in the case of mixed strategies (Proposition 3).

85     These findings suggest two ways in which one can leverage existing algorithms and heuristics for computing mixed equilibria to find behavioural equilibria. The first is a naive approach, in which we search for mixed equilibria and check whether they satisfy the aforesaid independence property (Section 4.1). The second approach is more sophisticated and is based on support enumeration,  
 90     which checks each for the existence of a behavioural equilibrium with that support (Sections 4.2 and 4.3). We explore both the use of linear programming and numerical methods to perform these checks. A third approach, which is inspired by the linear programming characterisation of correlated equilibria, allows us to use numerical methods to find behavioural equilibria immediately (Section 4.4).

95     In our experimental study in Section 5, we exploit the optimisations provided by the standard game-solving toolkit GAMBIT [9] for the naive approach. The support enumeration method can be enhanced by using a *support elimination* heuristic (Section 4.3), which enables us to filter out sets of pure strategies that we know cannot be the support of a behavioural equilibrium.

## 100     2. Boolean Games and Weighted Boolean Games

Boolean games provide a concise and elegant representation of strategic games with binary payoffs in  $\{0, 1\}$ . Boolean games were first presented by Harrenstein et al. [1], although more general definition (now the “standard”) was given by Bonzon et al. [10, 2, 3]. In the latter model, each player chooses the  
 105     values for a set of propositional variables under its unique control in pursuit of the satisfaction of a goal, which is expressed as a formula of propositional logic.

In this paper, we will extend the traditional framework of Boolean games to weighted Boolean games, in which the preferences of the players are given by a set of weighted formulas, rather than by a single, unweighted one. This approach  
 110     is inspired by Jeong and Shoham’s *marginal contribution nets* for cooperative

games [11], Elkind and Wooldridge’s *hedonic coalition nets* for hedonic games [12], and the *weighted Boolean formula games* proposed by Mavronicolas et al. [7].

In what follows, we make extensive use of the standard language of propositional logic, in which formulas are constructed from a finite set  $\Phi$  of Boolean variables, using the classical Boolean connectives  $\neg$  (negation),  $\wedge$  (conjunction),  $\vee$  (disjunction),  $\rightarrow$  (material implication), and  $\leftrightarrow$  (bi-implication).

Formally, a **(weighted) Boolean game** is given by a structure

$$(N, \Phi_1, \dots, \Phi_n, \Gamma_1, \dots, \Gamma_n),$$

where  $N$  is a set of  $n$  **players**,  $\Phi_i$  is the set of **(propositional) variables** controlled by player  $i$ , and  $\Gamma_i$  is a finite **goal set** (or **Boolean net**) for player  $i$  consisting of **rules** of the form  $(\gamma, m)$ . We assume that the sets  $\Phi_1, \dots, \Phi_n$  are pairwise disjoint, that is,  $i \neq j$  implies  $\Phi_i \cap \Phi_j = \emptyset$ . We further assume that every variable is controlled by some player, so  $\Phi = \Phi_1, \dots, \Phi_n$ . Each rule  $(\gamma, m) \in \Gamma_i$  consists of a propositional formula  $\gamma$  and a real number  $m \in \mathbb{R}$ .<sup>1</sup> The goal set  $\Gamma_i$  represents the preferences of player  $i$ , in a manner that we describe below. The special class of weighted Boolean games in which each player’s goal set  $\Gamma_i$  consists of a single pair  $(\gamma_i, 1)$  we refer to as **01-Boolean games**. We then also say that  $\gamma_i$  is player  $i$ ’s goal and simply write  $\gamma_i$  for  $(\gamma_i, 1)$ . The games in this class correspond to Boolean games as they were originally conceived.

A **(pure) strategy** for player  $i$  is a truth valuation for the variables under the control of  $i$ , that is, an assignment of either truth or falsity to each element of  $\Phi_i$ . Pure strategies for a player  $i$  are conveniently represented as subsets  $v_i$  of  $\Phi_i$ , intuitively consisting of those variables that are set to true. A **profile (of pure strategies)** or **outcome** is a tuple  $v = (v_1, \dots, v_n)$  consisting of one pure strategy for each player. Each such strategy profile  $v = (v_1, \dots, v_n)$  naturally corresponds to the valuation given by  $v_1 \cup \dots \cup v_n$  and propositional formulas

---

<sup>1</sup>Restricted classes of games can be studied by putting further restrictions on the values that  $m$  can take. This paper concerns the general model without such restrictions.

	$r$	$\bar{r}$		$rs$	$r\bar{s}$	$\bar{r}s$	$\bar{r}\bar{s}$
$pq$	0, 1	1, 0	$pq$	1, 0	0, 0	0, 1	0, 1
$p\bar{q}$	0, 1	0, 1	$p\bar{q}$	1, 0	0, 1	0, 0	0, 0
$\bar{p}q$	0, 1	0, 1	$\bar{p}q$	0, 1	1, 0	0, 0	1, 0
$\bar{p}\bar{q}$	1, 0	0, 1	$\bar{p}\bar{q}$	0, 1	0, 1	1, 0	1, 0
$G_1$			$G_2$				

Figure 1: Two two-player Boolean games,  $G_1$  and  $G_2$ .

over  $\Phi$  can thus straightforwardly be interpreted on strategy profiles using the standard semantics of propositional logic. By  $v \models \varphi$  we will denote that the valuation corresponding to strategy profile  $v$  satisfies formula  $\varphi$ .

Player  $i$  with goal set  $\Gamma_i$  seeks to maximise a utility function  $u_i$  which is defined such that, for all profiles  $v$ ,

$$u_i(v) = \sum_{(\gamma, m) \in \Gamma_i} (m \cdot \text{sgn}(v, \gamma)),$$

where

$$\text{sgn}(v, \gamma) = \begin{cases} 1 & \text{if } v \models \gamma \\ 0 & \text{otherwise.} \end{cases}$$

For a player  $i$  with goal  $\gamma_i$  in a 01-Boolean game, this reduces to  $u_i(v) = \text{sgn}(v, \gamma_i)$ , that is,  $u_i(v) = 1$  if  $v \models \gamma_i$ , and  $u_i(v) = 0$  otherwise. We let  $(v_{-i}, v'_i)$  denote the profile  $(v_i, \dots, v_{i-1}, v'_i, v_{i+1}, \dots, v_n)$ . Moreover, we have  $p\bar{q}\bar{r}$  denote the profile in which variable  $p$  is set to true and variables  $q$  and  $r$  to false, and similarly for other profiles and strategies.

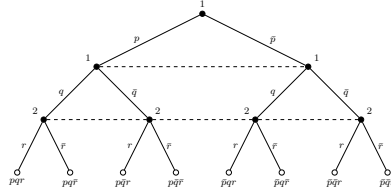
We present two examples of two-player 01-Boolean games in Figure 1. In  $G_1$ , player 1 controls variables  $p$  and  $q$ , and chooses rows, whereas player 2 controls  $r$  and chooses columns. The goals of the two players are given by  $\gamma_1 = (p \leftrightarrow q) \wedge (p \leftrightarrow \neg r)$  and  $\gamma_2 = \neg \gamma_1$ , respectively. Observe that thus  $G_1$  is a zero-sum game, which is not a general characteristic of Boolean games, as witnessed by game  $G_2$ , where the players have goals  $\gamma'_1 = (p \wedge r \wedge s) \vee (\neg p \wedge q \wedge \neg s) \vee (\neg p \wedge \neg q \wedge \neg r)$  and  $\gamma'_2 = (p \wedge q \wedge \neg r) \vee (\neg p \wedge r \wedge s) \vee (\neg q \wedge r \wedge \neg s)$ , respectively.

150 2.1. Mixed and Behavioural Strategies

Although Boolean games have mostly been studied in the context of pure strategies, they admit two intuitive ways in which a player can randomise over her choices. The first is the natural interpretation of mixed strategies for Boolean games, in which each pure strategy is played with a certain probability. Thus, a mixed strategy is defined by a probability distribution over pure strategies. However, a second natural interpretation is also possible, in which we randomise at the level of individual variables. In this latter setting, for each variable, the controlling player assigns the probability that this variable will be set to true. In this second setting, which is the main focus of this paper, we say that players have *behavioural strategies*.

Thus, formally, a **mixed strategy** for player  $i$  controlling variables  $\Phi_i$  is a probability distribution over  $2^{\Phi_i}$ , that is, a function  $\sigma_i : 2^{\Phi_i} \rightarrow [0, 1]$  such that  $\sum_{v \in 2^{\Phi_i}} \sigma(v) = 1$ . A **mixed strategy profile** is then a tuple  $\sigma = (\sigma_1, \dots, \sigma_n)$ , where each  $\sigma_i$  is a mixed strategy for player  $i$ . By contrast, a **behavioural strategy** for player  $i$  controlling variables  $\Phi_i$  is a function  $\beta_i : \Phi_i \rightarrow [0, 1]$ , with the intended interpretation that  $\beta_i(p)$  is the probability that variable  $p$  will be assigned the value true.<sup>2</sup> We write  $\beta_i(\bar{p})$  for  $1 - \beta_i(p)$ . There is an obvious one-to-one correspondence between pure strategies and behavioural strategies

<sup>2</sup>Our terminology is inspired by a close formal relationship between our concept of behavioural strategy and the one that is commonly known from the theory of extensive-form games. A **behavioural strategy** for a player  $i$  in an extensive-form game is a function that assigns to each information set of  $i$  a probability distribution over the actions available to  $i$  (see, e.g., Chapter 6 in [13] for an extensive presentation). For every Boolean game  $G$  on a set of variables  $\Phi$  we can define an extensive-form game of imperfect information of height  $|\Phi|$  whose behavioural strategies correspond to the behavioural strategies in the original Boolean game. All vertices of same height compose a separate information set and at each of them one propositional variable can be set to true or false by the player controlling the respective variable. The extensive-form game to the right thus depicts the extensive-form game associated with game  $G_1$  in Figure 1. A straightforward bijection now maps the behavioural strategies of  $G$  onto the behavioural strategies in  $E_G$  in such a way that very behavioural profile in  $G$  yields the same probability distribution over the outcomes as the corresponding profile does in  $E_G$ . The extensive-form games thus obtained from Boolean games, however, do not generally satisfy the property known as *perfect recall*, rendering much of the theory on extensive-form games inapplicable to our setting.





that take values in 0 and 1: define for pure strategy  $v_i$  behavioural strategy  $\beta_i^v$  such that  $\beta_i(p) = 1$  if  $p \in v$ , and  $\beta_i(p) = 0$ , otherwise.

Clearly, every behavioural strategy  $\beta_i$  corresponds to a mixed strategy  $\hat{\beta}_i$  such that, for every  $v_i \in 2^{\Phi_i}$ ,

$$\hat{\beta}_i(v_i) = \prod_{p \in v_i} \beta_i(p) \cdot \prod_{p \in \Phi_i \setminus v_i} \beta_i(\bar{p}).$$

We also say that behavioural strategy  $\beta_i$  **induces** mixed strategy  $\hat{\beta}_i$ , and that  $\hat{\beta}_i$  is **sustained** by  $\beta_i$ .

As the domain  $\Phi_i$  of a behavioural strategy is exponentially smaller than the domain  $2^{\Phi_i}$  of a mixed strategy, the former provides a concise representation of the latter. This representation, however, is not complete. If a player controls only one variable, her mixed and behavioural strategies coincide. However, if she controls more than one variable, not every mixed strategy is induced by a behavioural strategy. For instance, in game  $G_1$  in Figure 1, no behavioural strategy for the row player induces the mixed strategy  $\sigma_{Row}$  defined such that  $\sigma_{Row}(pq) = \sigma_{Row}(\bar{p}\bar{q}) = \frac{1}{2}$  and  $\sigma_{Row}(p\bar{q}) = \sigma_{Row}(\bar{p}q) = 0$ . We say that a mixed strategy  $\sigma_i: 2^{\Phi_i} \rightarrow [0, 1]$  has the **behavioural independence property (BIP)** if for all  $p \in \Phi_i$  and all  $v_i, w_i \in 2^{\Phi_i}$ ,

$$\sigma_i(v_i \cup \{p\}) \cdot \sigma_i(w_i \setminus \{p\}) = \sigma_i(w_i \cup \{p\}) \cdot \sigma_i(v_i \setminus \{p\}). \quad (\text{BIP})$$

The following result then gives a complete characterisation of when a mixed strategy profile is sustained by a behavioural one.

**Proposition 1.** *A mixed strategy  $\sigma_i: 2^{\Phi_i} \rightarrow [0, 1]$  is sustained by some behavioural strategy  $\beta_i: \Phi_i \rightarrow [0, 1]$  if and only if  $\sigma_i$  has the behavioural independence property. Moreover,  $\sigma_i$  is sustained by a behavioural strategy if and only if it is sustained by a behavioural strategy  $\beta_i$  such that for all  $p \in \Phi_i$  and every*

strategy  $v_i$  in  $2^{\Phi_i}$  with  $\sigma_i(v_i) > 0$ ,

$$\beta_i(p) = \frac{\sigma_i(v_i \cup \{p\})}{\sigma_i(v_i \cup \{p\}) + \sigma_i(v_i \setminus \{p\})}.$$

*Sketch of proof:* The “only if”-direction of the first part is straightforward by expanding and rearranging the terms. For the opposite direction, assume that, for all  $p \in \Phi_i$  and all  $v_i, w_i \in 2^{\Phi_i}$ ,

$$\sigma_i(v_i \cup \{p\}) \cdot \sigma_i(w_i \setminus \{p\}) = \sigma_i(w_i \cup \{p\}) \cdot \sigma_i(v_i \setminus \{p\}). \quad (*)$$

175 Define for each player  $i$  the behavioural strategy  $\beta_i: \Phi_i \rightarrow [0, 1]$  such that  $\beta(p) = \sum_{v_i \in 2^{\Phi_i}: p \in v_i} \sigma(v_i)$  for every  $p \in \Phi_i$ . For every  $\Psi \subseteq \Phi_i$ , moreover, define the mixed strategy  $\sigma_i|_{\Psi}: 2^{\Psi} \rightarrow [0, 1]$  such that  $\sigma_i|_{\Psi}(v_i) = \sum_{u_i \in 2^{\Phi_i \setminus \Psi}} \sigma(v_i \cup u_i)$  for all  $v_i \in 2^{\Psi}$ . Observe that, defined thus,  $\sigma_i|_{\Psi}$  is a well-defined probability distribution over  $2^{\Psi}$ . By induction on the size of  $\Psi$ , it can now be proven that  
 180  $\sigma_i|_{\Psi}(v_i) = \prod_{p \in \Psi \cap v_i} \beta(p) \cdot \prod_{p \in \Psi \setminus v_i} \beta(\bar{p})$  for every  $\Psi \subseteq \Phi_i$  and every  $v_i \in 2^{\Psi}$ . The result then follows by setting  $\Psi = \Phi_i$ .

The “only if”-direction of the second part is immediate. For the opposite direction, let  $v_i \in 2^{\Phi_i}$  such that  $\sigma_i(v_i) > 0$  and define

$$c = \prod_{q \in v_i \cap (\Phi_i \setminus \{p\})} \beta_i(q) \cdot \prod_{q \in v_i \setminus (\Phi_i \cup \{p\})} \beta_i(\bar{q}).$$

Since  $\sigma_i(v_i) > 0$ , also  $c > 0$ . Now assume  $\beta_i$  sustains  $\sigma_i$ . Then, both  $\sigma_i(v_i \cup \{p\}) = \beta(p) \cdot c$  and  $\sigma_i(v_i \setminus \{p\}) = \beta(\bar{p}) \cdot c$ . Therefore,  $\beta_i(p) = \frac{\sigma_i(v_i \cup \{p\})}{c}$ . As  $\beta_i(p) + \beta_i(\bar{p}) = 1$ , moreover,

$$\beta_i(p) = \frac{\sigma_i(v_i \cup \{p\})}{c \cdot (\beta_i(p) + \beta_i(\bar{p}))} = \frac{\sigma_i(v_i \cup \{p\})}{\sigma_i(v_i \cup \{p\}) + \sigma_i(v_i \setminus \{p\})}.$$

This concludes the proof sketch.  $\square$

With this characterisation in place, we can decide whether a mixed strategy is sustained by a behavioural one, but the question remains how computationally

185 complex this question is in general. If the mixed strategy is represented explicitly,  
 say as a list associating a probability with each pure strategy, the problem is  
 “padded” to such an extent that it is easily seen the problem can be solved in  
 polynomial time by leveraging Proposition 1. The more interesting case is, if the  
 mixed strategy is given by a polynomially-sized Turing machine that outputs  
 190 the probability for each pure strategy in polynomial time. This leads us to the  
 following decision problem.

#### BEHAVIOURAL STRATEGY

*Given:* A player  $i$ , a set  $\Phi_i$  of variables with  $|\Phi_i| \geq 2$ , and a mixed  
 strategy  $\sigma_i: 2^{\Phi_i} \rightarrow [0, 1]$  represented by a Turing machine of  
 polynomial size in  $|\Phi_i|$  that returns  $\sigma_i(v_i)$  in polynomial time  
 for each pure strategy  $v_i$ .

*Problem:* Is there a behavioural strategy  $\beta_i: \Phi_i \rightarrow [0, 1]$  for  $i$  that in-  
 duces  $\sigma_i$ , that is,  $\hat{\beta}_i = \sigma_i$ ?

Leveraging the first part of Proposition 1, we find that BEHAVIOURAL STRATEGY  
 is coNP-complete.

195 **Proposition 2.** BEHAVIOURAL STRATEGY is coNP-complete.

*Sketch of proof:* For membership in coNP, it suffices to observe that one can  
 check in polynomial time one can verify for given valuations  $v_i$  and  $w_i$  and  
 propositional variable  $p$  whether  $\sigma_i(v_i \cup \{p\}) \cdot \sigma_i(w_i \setminus \{p\}) \neq \sigma_i(w_i \cup \{p\}) \cdot \sigma_i(v_i \setminus \{p\})$ .  
 The first part of Proposition 1 then yields the result.

For coNP-hardness, we reduce TAUTOLOGY. Let  $\varphi$  be a formula over a set  
 of  $k$  propositional variables in  $\Phi_i$  and  $p$  a variable in  $\Phi_i$ . Now define the mixed  
 strategy  $\sigma_i: \Phi \rightarrow [0, 1]$  such that for every  $v_i \in 2^{\Phi_i}$ ,

$$\sigma_i(v_i) = \begin{cases} \frac{1}{2^k} & \text{if } v_i \cup \{p\} \models \varphi \text{ and } v_i \setminus \{p\} \models \varphi, \\ \frac{2}{2^k} & \text{if } |v_i| \text{ is even, and } v_i \cup \{p\} \not\models \varphi \text{ or } v_i \setminus \{p\} \not\models \varphi, \\ 0 & \text{otherwise.} \end{cases}$$

200 Observe that, defined thus,  $\sigma_i$  is a well-defined mixed strategy that, moreover,  
 can be represented by a Turing machine of size polynomial in  $|\Phi_i|$ . We can then  
 show that  $\sigma_i$  is sustained by a behavioural strategy if and only if  $\varphi$  is a tautology.  
 This concludes the proof sketch.  $\square$

The prospects for an *efficient* algorithm for deciding sustenance of mixed  
 205 strategies by behavioural strategies are therefore rather bleak. The second part  
 of Proposition 1 nevertheless enables us to formulate an algorithm, which we  
 refer to as BEHAVIOURAL (Algorithm 1), which identifies some pure strategy  $v_i$   
 that  $\sigma_i$  assigns positive probability to, sets the values for  $\beta_i$  as in Theorem 1, and  
 checks whether  $\beta_i$  in fact induces  $\sigma_i$ . We can thus check in time  $O(|\Phi_i| \cdot 2^{|\Phi_i|})$   
 210 whether a given mixed strategy  $\sigma_i$  is sustained by a behavioural one.

## 2.2. Mixed and Behavioural Supports

A key concept for both mixed and behavioural strategies is their *support*.  
 In Section 4, we will see the important role they play in the computation of  
 behavioural equilibria. The **(mixed) support**  $S_{\sigma_i}$  of a mixed strategy  $\sigma_i$  is  
 215 the set of pure strategies that are given positive probability under  $\sigma_i$ . That is,  
 $S_{\sigma_i} = \{v_i \in 2^{\Phi_i} : \sigma_i(v_i) > 0\}$ . It is easy to see that the set of possible mixed  
 supports for player  $i$  in a Boolean game is of size doubly exponential in the  
 number  $|\Phi_i|$  of propositional variables.

If we restrict attention to mixed strategies  $\hat{\beta}_i$  that are induced by behavioural  
 strategies  $\beta_i : \Phi_i \rightarrow [0, 1]$ , we find that the number of supports is exponentially  
 smaller. Let  $v_i$  in be a pure strategy in  $2^{\Phi_i}$ . It can then easily be appreciated  
 that  $v_i \in S_{\hat{\beta}_i}$  if and only if, for all  $p \in \Phi_i$ ,  $\beta_i(p) = 1$  implies  $p \in v_i$ , and  
 $\beta_i(p) = 0$  implies  $p \notin v_i$ . For instance, if  $\beta_i$  is such that  $\beta_i(p) = 0$ ,  $\beta_i(q) = 1$ , and  
 $\beta_i(r) = \frac{1}{2}$ , then  $S_{\hat{\beta}_i} = \{\bar{p}qr, \bar{p}q\bar{r}\}$ . Thus, the support of a behavioural strategy  
 is fully determined if we know, for each variable  $p \in \Phi_i$ , whether  $\beta_i(p) = 0$ ,  
 $\beta_i(p) = 1$ , or  $\beta_i(p) \in (0, 1)$ . It follows that the number of possible supports of  
 mixed strategies that are induced by behavioural strategies, which we also refer  
 to as **behavioural supports**, is  $3^{|\Phi_i|}$ . This number is of size polynomial in the

---

**Algorithm 1** BEHAVIOURAL( $\sigma_i, \Phi_i$ )

---

**Input:** a list  $\Phi_i[1 \dots |\Phi_i|]$  of propositional variables in  $\Phi_i$ , a mixed strategy  $\sigma_i$  over  $2^{\Phi_i}$ .

**Output:** The behavioural strategy  $\beta_i : \Phi_i \rightarrow [0, 1]$  that induces  $\sigma_i$ , or “NULL” if none exists.

```
1: Initialise map  $\beta_i : \Phi_i \rightarrow [0, 1]$ 
2: // Find any  $w_i \in 2^{\Phi_i}$  with  $\sigma_i(w_i) > 0$ .
3:  $w_i \leftarrow 0$ 
4: for  $v_i = 0$  to  $2^{|\Phi_i|} - 1$  do
5:   if  $\sigma_i(v_i) > 0$  then
6:      $w_i \leftarrow v_i$ 
7:     Break
8: // Determine value of  $\beta_i(p)$  for each  $p \in \Phi_i$ .
9: for  $j = 1$  to  $|\Phi_i|$  do
10:  if  $\Phi[j] \in w_i$  then
11:     $\beta_i[\Phi_i[j]] \leftarrow \sigma_i(w_i) / (\sigma_i(w_i) + \sigma_i(w_i \setminus \{\Phi_i[j]\}))$ 
12:  else
13:     $\beta_i[\Phi_i[j]] \leftarrow \sigma_i(w_i \cup \{\Phi_i[j]\}) / (\sigma_i(w_i \cup \{\Phi_i[j]\}) + \sigma_i(w_i))$ 
14: // Lastly, check if  $\beta_i$  induces  $\sigma_i$ .
15: for  $v_i = 0$  to  $2^{|\Phi_i|} - 1$  do
16:    $prod \leftarrow 1$ 
17:   // For each bit of  $v_i$ , multiply  $prod$  by  $\beta_i[\Phi_i[j]]$  if the bit is 1, and by  $1 - \beta_i[\Phi_i[j]]$ 
   otherwise.
18:    $v'_i \leftarrow v_i$ 
19:   for  $j = 1$  to  $|\Phi_i|$  do
20:     if  $v'_i \equiv 1 \pmod{2}$  then
21:        $prod \leftarrow prod * \beta_i[\Phi_i[j]]$ 
22:        $v'_i \leftarrow v'_i - 1$ 
23:     else
24:        $prod \leftarrow prod * (1 - \beta_i[\Phi_i[j]])$ 
25:        $v'_i \leftarrow v'_i / 2$ 
26:   if  $\sigma_i(v_i) \neq prod$  then
27:     return NULL
28: return  $\beta_i$ 
```

---

number  $2^{|\Phi_i|}$  of pure strategies and, consequently, exponentially smaller than the number of supports for mixed strategies for  $i$ . To make this precise, we define a **behavioural support function** for player  $i$  as a function  $B_i : \Phi_i \rightarrow \{0, 1, 2\}$ . We can now associate with each behavioural strategy  $\beta_i$  a behavioural support

function  $B_{\beta_i}$  such that, for all  $p \in \Phi_i$ ,

$$B_{\beta_i}(p) = \begin{cases} 0 & \text{if } \beta_i(p) = 0, \\ 1 & \text{if } \beta_i(p) = 1, \text{ and} \\ 2 & \text{if } \beta_i(p) \in (0, 1). \end{cases}$$

Formally, we have the following result.

**Proposition 3.** *Let player  $i$  control  $\Phi_i$  in a Boolean game. Then,*

$$|\{S_{\hat{\beta}_i} : \beta_i \text{ is a behavioural strategy}\}| = 3^{|\Phi_i|}.$$

Moreover, for  $\beta_i$  a behavioural strategy for player  $i$  and  $v \in 2^{\Phi_i}$ , we have  $v_i \in S_{\hat{\beta}_i}$  if and only if for all  $p \in \Phi_i$  both (i)  $B_{\beta_i}(p) = 0$  implies  $p \notin v_i$ , and (ii)  $B_{\beta_i}(p) = 1$  implies  $p \in v_i$ .

*Proof:* For the first part we define a one-one and onto mapping that associates every behavioural support function  $B \in \{0, 1, 2\}^{\Phi_i}$  with a mixed support  $X_B \subseteq 2^{\Phi_i}$  such that  $v_i \in X_B$  if and only if for all  $p \in \Phi_i$  both  $B(p) = 1$  implies  $p \in v_i$ , and  $B(p) = 0$  implies  $p \notin v_i$ .

To see that  $X_B$  is a behavioural support, let  $\beta_i^B$  be defined such that  $\beta_i^B(p) = 0$  if  $B(p) = 0$ ,  $\beta_i^B(p) = 1$  if  $B(p) = 1$ , and  $\beta_i^B(p) = \frac{1}{2}$  otherwise. It then holds that  $S_i(\hat{\beta}_i^B) = X_B$ .

To see that the mapping is one-one, assume  $B \neq B'$ . Without loss of generality, we may assume that one of the following cases obtains. For some  $p \in \Phi_i$ , (i)  $B(p) = 1$  and  $B'(p) = 0$ , (ii)  $B(p) = 2$  and  $B'(p) = 0$ , or (iii)  $B(p) = 2$  and  $B'(p) = 1$ . Let  $v_i \in X_B$ . Then, if (i),  $p \in v_i$ , and it follows that  $v_i \notin X_{B'}$ . If (ii), then  $v_i \cup \{p\} \in X_B$  as well. However,  $v_i \cup \{p\} \notin X_{B'}$ . Similarly, if (iii), then  $v_i \setminus \{p\} \in X_B$ , but  $v_i \setminus \{p\} \notin X_{B'}$ . In each case we find that  $X_B \neq X_{B'}$ , as desired.

To show that the mapping is onto, let  $\beta_i$  be a behavioural strategy and  $v_i$  a pure strategy for player  $i$ . Then, the following equivalences hold:

$$\begin{aligned}
\hat{\beta}_i(v_i) = 0 & \text{ iff } \prod_{p \in v_i} \beta_i(p) \cdot \prod_{p \in \Phi_i \setminus v_i} \beta_i(\bar{p}) = 0 \\
& \text{ iff } \beta_i(p) = 0 \text{ for some } p \in v_i, \text{ or } \beta_i(p) = 1 \text{ for some } p \in \Phi_i \setminus v_i \\
& \text{ iff } B_{\beta_i}(p) = 0 \text{ for some } p \in v_i, \text{ or } B_{\beta_i}(p) = 1 \text{ for some } p \in \Phi_i \setminus v_i \\
& \text{ iff } v_i \notin X_{B_{\beta_i}}.
\end{aligned}$$

Hence,  $X_{B_{\beta_i}} = S_{\hat{\beta}_i}$ , which also proves the second part of the proposition.  $\square$

Clearly,  $S_{\hat{\beta}_i} = S_{\hat{\beta}'_i}$  whenever  $B_{\beta_i} = B_{\beta'_i}$ . We therefore also introduce the following notation for the behavioural support corresponding to a behavioural support function  $B_i$ :

$$\hat{B}_i = \{v_i \in 2^{\Phi_i} : \text{for all } p \in \Phi_i, B(p) = 0 \text{ implies } p \notin v_i, \text{ and } B(p) = 1 \text{ implies } p \in v_i\}.$$

Using the *sgn*-function introduced on page 7, the second part of Proposition 3 can also be stated more concisely as follows:  $v_i \in \hat{B}_i$  if and only if for all  $p \in \Phi_i$  we have  $B_i(p) \in \{\text{sgn}(v_i, p), 2\}$ . If  $\Phi_i = \{p_1, \dots, p_k\}$ , this observation furthermore establishes a connection between the *ternary* list representation  $[B_i(p_1), \dots, B_i(p_k)]$  of behavioural support functions  $B_i$  and the *binary* list representation  $[\text{sgn}(v_i, p_1), \dots, \text{sgn}(v_i, p_k)]$  of pure strategies  $v_i$ . Whether pure strategy  $v_i$  is contained in support  $\hat{B}_i$  can now be decided by means of a straightforward symbolic (pairwise) comparison of the entries in the list representations of  $v_i$  and  $B_i$ .

### 3. Behavioural Equilibrium

*Nash equilibrium* is arguably the most prominent solution concept in game theory and applies to strategy profiles from which none of the players has an incentive to deviate. Nash equilibria can be defined for mixed and behavioural strategies alike, using the concept of expected utility.

Let  $\sigma(v) = \prod_{i \in N} \sigma_i(v_i)$  and define the **expected utility** of a mixed strategy profile  $\sigma = (\sigma_1, \dots, \sigma_n)$  for a player  $i$  in the conventional manner as

$$u_i(\sigma) = \sum_{v \in 2^\Phi} \sigma(v) u_i(v).$$

If, for a given a profile  $\sigma = (\sigma_1, \dots, \sigma_n)$  of mixed strategies,  $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$  for all mixed strategies  $\sigma'_i$  for player  $i$ , then we refer to  $\sigma_i$  as a **best response** for player  $i$  to  $\sigma_{-i}$ . Profile  $\sigma = (\sigma_1, \dots, \sigma_n)$  is a **Nash equilibrium in mixed strategies** (henceforth also **mixed equilibrium**), if for all players  $i$ ,  $\sigma_i$  is a best response to  $\sigma_{-i}$ . A Nash equilibrium  $\sigma = (\sigma_1, \dots, \sigma_n)$  is a **pure Nash equilibrium** if  $\sigma_i(v_i) \in \{0, 1\}$  for all players  $i$  and  $v_i \in 2^{\Phi_i}$ . Analogously, we define a **Nash equilibrium in behavioural strategies**, or simply a **behavioural equilibrium**, as a profile  $\beta = (\beta_1, \dots, \beta_n)$  of behavioural strategies such that, for all players  $i$  and all behavioural strategies  $\beta'_i$ ,

$$u_i(\hat{\beta}_i, \hat{\beta}_{-i}) \geq u_i(\hat{\beta}'_i, \hat{\beta}_{-i}).$$

Observe that this definition only takes into account players deviating from a *behavioural* strategy to another *behavioural* strategy. Even so, we find that the mixed profile  $\hat{\beta}$  induced by a behavioural equilibrium  $\beta$  is also a Nash equilibrium in mixed strategies. The key insight is every mixed as well as every behavioural strategy profile may be assumed to have a pure best response for each player. Intuitively—and with a slight abuse of terminology—one could say that behavioural equilibria are exactly those mixed equilibria that satisfy the behavioural independence property. More formally, we have the following proposition, which also implies that every pure equilibrium defines a behavioural equilibrium.

**Proposition 4.** *If  $\beta = (\beta_1, \dots, \beta_n)$  is a behavioural equilibrium, then  $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_n)$  is an equilibrium in mixed strategies. Moreover, a mixed strategy profile  $\sigma = (\sigma_1, \dots, \sigma_n)$  is sustained by a behavioural equilibrium if and only if  $\sigma$  is a mixed equilibrium and satisfies the behavioural independence property.*



*Proof:* First assume that  $\beta = (\beta_1, \dots, \beta_n)$  is a behavioural equilibrium and, for contradiction, that  $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_n)$  is not a mixed equilibrium. Then, there is some player  $i$  and some pure strategy  $v_i$  such that  $u_i(\hat{\beta}_{-i}, v_i) > u_i(\hat{\beta})$ . Define behavioural strategy  $\beta_i^*$  such that, for all  $p \in \Phi_i$ , we have that  $\beta_i^*(p) = 1$  if  $p \in v_i$ , and  $\beta_i^*(p) = 0$  otherwise. Then,  $\hat{\beta}_i^* = v_i$ , and it follows that  $\beta$  is not a behavioural equilibrium.

For the “only if”-direction of the second part, let  $\sigma$  be a profile of mixed strategies that is sustained by a behavioural equilibrium  $\beta^*$ . By the first part,  $\hat{\beta}^*$  is an equilibrium in mixed strategies. Moreover, by Proposition 1, we know that  $\sigma$  satisfies the behavioural independence property.

Finally, for the “if”-direction, assume that  $\sigma$  is a mixed equilibrium and satisfies the behavioural independence property. By Proposition 1, some profile  $\beta = (\beta_1, \dots, \beta_n)$  of behavioural strategies sustains  $\sigma$ , that is,  $(\hat{\beta}_1, \dots, \hat{\beta}_n) = (\sigma_1, \dots, \sigma_n)$ . Then,  $u_i(\hat{\beta}) \geq (\hat{\beta}_{-i}, \sigma'_i)$  for all players  $i$  and all mixed strategies  $\sigma'_i$ . Hence, in particular,  $u_i(\hat{\beta}) \geq (\hat{\beta}_{-i}, \hat{\beta}'_i)$  for all behavioural strategies  $\beta'_i$  for player  $i$ . It follows that  $\beta$  is a behavioural equilibrium that sustains  $\sigma$ .  $\square$

Proposition 4 also yields a behavioural analogue to the well-known *Indifference Principle*, which intuitively says that a profile  $\sigma$  is a mixed equilibrium if and only if, given  $\sigma_{-i}$ , each player  $i$  is indifferent between playing any two pure strategies in the support of  $\sigma_i$ , and weakly prefers any pure strategy in the support of  $\sigma_i$  to any pure strategy that is not. This eventually gives us an algorithm to check for the existence of behavioural Nash equilibria. The following proposition is a corollary of Proposition 4 and the Indifference Principle for mixed strategies.

**Proposition 5 (Behavioural Indifference Principle).** *Let  $\beta$  be a behavioural strategy profile in a Boolean game  $G$ . Then,  $\beta$  is a behavioural equilibrium if and only if  $u_i(v_i, \hat{\beta}_{-i}) = u_i(v'_i, \hat{\beta}_{-i})$  and  $u_i(v_i, \hat{\beta}_{-i}) \geq u_i(v''_i, \hat{\beta}_{-i})$ , for all players  $i$  and strategies  $v_i, v'_i \in S_{\hat{\beta}_i}$  and  $v''_i \notin S_{\hat{\beta}_i}$ .*

As a corollary we obtain the following result, which has a stronger behavioural flavour.

295 **Corollary 6.** *A behavioural strategy profile  $\beta$  is a behavioural Nash equilibrium if and only if for every player  $i$  and every behavioural strategy  $\beta'_i$ , we have  $u_i(\hat{\beta}_i, \hat{\beta}_{-i}) = u_i(\hat{\beta}'_i, \hat{\beta}_{-i})$ , if  $B_{\beta_i} = B_{\beta'_i}$ , and  $u_i(\hat{\beta}_i, \hat{\beta}_{-i}) \geq u_i(\hat{\beta}'_i, \hat{\beta}_{-i})$ , otherwise. In particular, if  $\beta_i(p) \in (0, 1)$ , then  $i$  is indifferent among all values of  $\beta_i(p)$ .*

Another important game-theoretic solution concept is *correlated equilibrium* [14, 13, Chapter 8]. Let  $\pi: 2^\Phi \rightarrow [0, 1]$  denote a probability distribution over the set  $2^\Phi$  of pure strategies profiles of a weighted Boolean game. Then,  $\pi$  is a **correlated equilibrium** if for all players  $i$  and all actions  $v_i, v'_i \in 2^{\Phi_i}$ ,

$$\sum_{w_{-i} \in 2^{\Phi-i}} \pi(w_{-i}, v_i) u_i(w_{-i}, v_i) \geq \sum_{w_{-i} \in 2^{\Phi-i}} \pi(w_{-i}, v_i) u_i(w_{-i}, v'_i). \quad (\text{CE})$$

Correlated equilibrium extends mixed Nash equilibrium in the sense that a probability distribution  $\pi_\sigma: 2^\Phi \rightarrow [0, 1]$  induced by a mixed equilibrium  $\sigma = (\sigma_1, \dots, \sigma_n)$  will be a correlated equilibrium. By virtue of Proposition 4, we find that the analogous statement holds for behavioural equilibria as well. As, moreover,  $\pi_\sigma$  is a correlated equilibrium if  $\sigma$  is a mixed equilibrium, we have the following counterpart of Proposition 4 for correlated equilibria.

**Corollary 7.** *If  $\beta = (\beta_1, \dots, \beta_n)$  is a behavioural equilibrium, then  $\pi_\beta$  is a correlated equilibrium. Moreover, a probability distribution  $\pi_\sigma: 2^\Phi \rightarrow [0, 1]$  is induced by a mixed profile  $\hat{\beta}$  sustained by a behavioural equilibrium  $\beta$  if and only if  $\pi$  is a correlated equilibrium and satisfies the following independence property:*

$$\text{for all } v, w \in 2^\Phi \text{ and } p \in \Phi: \quad \pi(v \cup \{p\}) \cdot \pi(w \setminus \{p\}) = \pi(w \cup \{p\}) \cdot \pi(v \setminus \{p\}).$$

### 305 3.1. Properties of Behavioural Equilibria

While Nash's theorem [15, 16] guarantees that every Boolean game has a Nash equilibrium in mixed strategies, we have seen that behavioural strategies are strictly less expressive than mixed strategies. Therefore, Nash's theorem does not guarantee the existence of a behavioural Nash equilibrium. Instead, we make the following observation.

**Proposition 8.** *Behavioural equilibria are not guaranteed to exist in Boolean games.*

To see this, consider again the Boolean game  $G_1$  depicted in Figure 1 and assume for contradiction that  $\beta = (\beta_1, \beta_2)$  is a behavioural equilibrium. This game has only one Nash equilibrium in mixed strategies, namely,  $\sigma = (\sigma_1, \sigma_2)$  such that  $\sigma_1(pq) = \sigma_1(\bar{p}\bar{q}) = \frac{1}{2}$ ,  $\sigma_1(p\bar{q}) = \sigma_1(\bar{p}q) = 0$ , and  $\sigma_2(r) = \sigma_2(\bar{r}) = \frac{1}{2}$ . By Proposition 4, it thus follows that  $\hat{\beta} = \sigma$ . Now consider the pure strategies  $v_1 = pq$  and  $w_1 = \bar{p}\bar{q}$  for player 1. Then,  $\sigma_1(pq) \cdot \sigma_1(\bar{p}\bar{q}) = \frac{1}{4} \neq 0 = \sigma_1(\bar{p}q) \cdot \sigma_1(p\bar{q})$ . Hence, mixed strategy  $\sigma_1$  does not satisfy the behavioural independence property, and by Proposition 1 it follows that  $\sigma_1$  is not sustained by any behavioural strategy of player 1, in particular not by  $\beta$ , a contradiction.

This example is significant because it shows that even zero-sum 01-Boolean games in which one player controls only one variable are not guaranteed to have behavioural Nash equilibria. Note, however, that if every player controls just one propositional variable, mixed and behavioural strategies coincide. Hence, in that special case, Nash equilibria are guaranteed to exist by virtue of Nash's theorem. In Section 5.5, we present some empirical data suggesting that in behavioural equilibria are considerably more common than pure equilibria.

It is commonly known that two-player strategic games always have a mixed Nash equilibrium in rational probabilities. We find that this is no longer the case for behavioural Nash equilibria. For an example, consider again the Boolean game  $G_2$  in Figure 1. We find it has a (unique) behavioural equilibrium  $\beta^* = (\beta_1^*, \beta_2^*)$  in which every variable is played with probability  $\frac{\sqrt{5}-1}{2}$ . To see this, consider the support  $S_{\beta^*} = (2^{\Phi_1}, 2^{\Phi_2})$ . By the Indifference Principle (Proposition 5), the expected utility of player 1 at  $\bar{p}q$  and  $\bar{p}\bar{q}$  against  $\beta_2^*$  should be identical, that is  $\beta_2^*(r) \cdot \beta_2^*(\bar{s}) + \beta^*(\bar{r}) \cdot \beta^*(\bar{s}) = \beta_2^*(\bar{r}) \cdot \beta_2^*(s) + \beta^*(\bar{r}) \cdot \beta^*(\bar{s})$ . Hence,  $\beta_2^*(r) = \beta_2^*(s)$ . Thus, we have:

$$\begin{aligned} u_1(pq, \hat{\beta}_2^*) &= \beta_2^*(r) \cdot \beta_2^*(s) = \beta_2^*(r)^2, \text{ and} \\ u_1(\bar{p}q, \beta_2^*) &= \beta_2(r) \cdot \beta_2(\bar{s}) + \beta_2^*(\bar{r}) \cdot \beta_2(\bar{s}) = 1 - \beta_2(r). \end{aligned}$$

As player 1 is similarly indifferent between  $pq$  and  $\bar{p}q$ , we obtain  $\beta_2^*(r)^2 + \beta_2^*(r) -$   
330  $1 = 0$ . It follows that  $\beta_2^*(r) = \beta_2^*(s) = \frac{\sqrt{5}-1}{2}$  is the only solution in  $[0, 1]$ . Indeed,  
it can easily be checked that  $u_1(v_1, \hat{\beta}_2^*) = \left(\frac{\sqrt{5}-1}{2}\right)^2$  for every  $v_1 \in 2^{\Phi_1}$ . An  
analogous argument for player 1 yields  $\beta_1^*(p) = \beta_2^*(q) = \frac{\sqrt{5}-1}{2}$ , and, hence,  
that  $\beta^*$  is a behavioural equilibrium. By more tenacious reasoning, it can  
335 moreover be seen that the game has no behavioural equilibria in which one of  
the players plays some propositional variable with probability 1. Hence,  $\beta^*$  is  
unique, even though, interestingly,  $G_2$  has infinitely many *mixed* equilibria.

**Proposition 9.** *A two-player Boolean game with both players controlling at least two variables may have a unique and irrational behavioural Nash equilibrium.*

#### 4. Computing Behavioural Equilibria

340 The theoretical results obtained in the previous sections give rise to three  
natural ways to compute behavioural equilibria in two-player Boolean games.  
After presenting these approaches, we evaluate them by means of a detailed  
experimental study in the next section.

##### 4.1. A Naive Approach: BNE\_NAIVE

345 Our first approach is *naive* in that it first searches for mixed Nash equilibria  
and then checks whether those found satisfy the behavioural independence  
property. The soundness of this method relies on Propositions 1 and 4. The  
advantage of this approach is that it allows leveraging the optimisations for  
finding mixed equilibria in general strategic games, such as those provided by  
350 GAMBIT [9].

There are algorithms to enumerate the mixed Nash equilibria of two-player  
games—for example, the *Lemke-Howson algorithm*, *Linear Complementarity*  
(LCP), or *Support Enumeration* [17, 18, 19, 20]. Consequently, Proposition 4 gives  
us a naive algorithm, to check the existence of behavioural Nash equilibria: given  
355 a Boolean game, we can iterate through the mixed Nash equilibria of the game and

---

**Algorithm 2** BNE\_NAIVE\_ALL( $G$ )

---

**Input:** A nondegenerate Boolean game  $G = (N, \Phi_1, \dots, \Phi_n, \gamma_1, \dots, \gamma_n)$  where for all  $i \leq N$ ,  $\Phi_i[1 \dots |\Phi_i|]$  is a list of propositional variables.

**Output:** A behavioural Nash equilibrium  $\beta$  of  $G$  or outputs “No behavioural NE” if no behavioural Nash equilibrium exists.

```
1:  $\Sigma^* \leftarrow \text{MIXED\_EQUILIBRIA}(G)$ 
2: // Determine the mixed equilibria of  $G$ .
3:  $\text{behavioural\_ne} \leftarrow \text{false}$ 
4: for  $\sigma \in \Sigma^*$  do
5:   Let  $\beta$  be an  $n$ -dimensional vector
6:   for  $i = 1$  to  $n$  do
7:      $\beta_i \leftarrow \text{BEHAVIOURAL}(\sigma_i, \Phi_i)$ 
8:     if  $\beta_i = \text{NULL}$  then
9:        $\beta \leftarrow \text{NULL}$ 
10:    Break
11:   if  $\beta \neq \text{NULL}$  then
12:      $\text{behavioural\_ne} \leftarrow \text{true}$ 
13:   output  $\beta$ 
14:   // ‘Break’ at this point in BNE_NAIVE_SOME
15: if not  $\text{behavioural\_ne}$  then
16:   output “No behavioural NE.”
```

---

check using BEHAVIOURAL whether they have equivalent behavioural strategy profiles. This algorithm—which has a variable helper program MIXED\_EQUILIBRIA that finds all mixed Nash equilibria—we refer to as BNE\_NAIVE. The algorithm comes in two varieties: BNE\_NAIVE\_ALL, to compute *all* behavioural equilibria, and BNE\_NAIVE\_SOME, to find a first behavioural equilibrium. Algorithm 2 presents the pseudo-code for BNE\_NAIVE\_ALL; BNE\_NAIVE\_SOME only differs in that ‘Break’ is inserted after line 13.

In order to analyse the time complexity of BNE\_NAIVE, let  $f(n)$  be the time complexity of the selected algorithm MIXED\_EQUILIBRIA used to compute the mixed Nash equilibria of a game with  $n$  outcomes. Observe that  $f$  will generally be an exponential function. Then, finding the mixed Nash equilibria of  $G$  takes time  $f(2^{|\Phi|})$ . Let  $M$  be the number of mixed equilibria of  $G$ . Recall that BEHAVIOURAL has time complexity  $O(|\Phi_i| \cdot 2^{|\Phi_i|})$ , so checking BEHAVIOURAL for

all players takes time in the order of  $\sum_{i \in N} |\Phi_i| \cdot 2^{|\Phi_i|} \leq \sum_{i \in N} |\Phi_i| \cdot 2^{|\Phi|} = |\Phi| \cdot 2^{|\Phi|}$ .  
 370 Therefore, BNE\_NAIVE has overall time complexity  $O(f(2^{|\Phi|}) + M \cdot |\Phi| \cdot 2^{|\Phi|})$ .

Observe that this naive approach involves first computing *all* mixed equilibria and then checks those found for their satisfying the behavioural independence property, that is, for their being sustained by a behavioural strategy profile. This is only feasible if the number of mixed equilibria is finite, and we will duly restrict  
 375 our experiments to such games (also see Section 5.2). A finite number of mixed equilibria is guaranteed for **non-degenerate** games, that is, for two-player games in which the number of pure best responses to no mixed strategy exceeds the size of its support [17, 21]. It is known that, in a precise formal sense, almost all strategic-form games are non-degenerate and, accordingly, are guaranteed  
 380 to have a finite and odd number of mixed Nash equilibria [22]. It is important to note that in 01-Boolean games specifically, degeneracy no longer necessarily occurs with probability zero. Rather, all non-trivial 01-Boolean games tend to be degenerate [23].

#### 4.2. Two Support Enumeration Methods: BNE\_SE\_LP and BNE\_SE\_NUM

385 Our second approach is inspired by the *Support Enumeration* algorithm for finding mixed Nash equilibria in normal-form games [24, 25, Chapter 4]. By leveraging the result of Proposition 3 that there are only  $3^{|\Phi|}$  behavioural supports, obtain two algorithms for finding behavioural equilibria that run *polynomially* in the number of  $2^{|\Phi|}$  of outcomes. The first is based on linear  
 390 programming and the second uses numerical methods.

The support enumeration algorithm for mixed equilibria iterates through the possible mixed strategy supports and solves a linear feasibility problem to find mixed equilibria with this support. This feasibility program, SE\_LP, is depicted in Figure 2 and captures the conditions of the Indifference Principle  
 395 under which a two-player strategic game has a mixed Nash equilibrium  $\sigma$  with support  $S = (S_1, S_2)$ . Since linear feasibility programs can be solved in time polynomial in the size of its constraints, solving SE\_LP is polynomial in  $|2^{\Phi_1}| \cdot |2^{\Phi_2}| = 2^{|\Phi|}$ . Recall that, although the linear feasibility program is

$$\begin{aligned}
\sum_{v_{-i} \in 2^{\Phi-i}} \sigma_{-i}(v_{-i}) u_i(v_i, v_{-i}) &= x_i & \forall i \in \{1, 2\}, v_i \in S_i & \quad (1) \\
\sum_{v_{-i} \in 2^{\Phi-i}} \sigma_{-i}(v_{-i}) u_i(v_i, v_{-i}) &\leq x_i & \forall i \in \{1, 2\}, v_i \in 2^{\Phi_i} \setminus S_i & \quad (2) \\
\sigma_i(v_i) &\geq 0 & \forall i \in \{1, 2\}, v_i \in S_i & \quad (3) \\
\sigma_i(v_i) &= 0 & \forall i \in \{1, 2\}, v_i \in 2^{\Phi_i} \setminus S_i & \quad (4) \\
\sum_{v_i \in 2^{\Phi_i}} \sigma_i(v_i) &= 1 & \forall i \in \{1, 2\} & \quad (5)
\end{aligned}$$

Figure 2: Linear feasibility program  $\text{SE\_LP}$  for mixed support  $S = (S_1, S_2)$  (adapted from [25]).

polynomial in the number of outcomes, there are exponentially many possible mixed strategy supports.

Our methods, however, exploit the result of Proposition 3 that *the number of behavioural supports is only  $3^{|\Phi_i|}$* . By virtue of Proposition 4, it then suffices to test, e.g., by invoking `BEHAVIOURAL`, whether any mixed equilibrium found with some behavioural support also satisfies the behavioural independence property. This gives us our first support enumeration algorithm for finding behavioural equilibria, `BNE_SE_LP` which runs polynomially in the number of outcomes. Like `BNE_NAIVE`, `BNE_SE_LP` also comes in two varieties: `BNE_SE_LP_ALL` (Algorithm 3), which finds all behavioural equilibria, and `BNE_SE_LP_SOME` which computes some if there is any. Algorithm `BNE_SE_LP` uses helper methods `MIXED_SUPPORT`, which translates a behavioural support function  $B$  into the corresponding mixed strategy support  $\hat{B}$  it induces, and `UTILITY`, which defines the utility function vector for the Boolean game that is needed to define the linear feasibility program.

As in the original support enumeration algorithm for finding mixed Nash equilibria, we require that the input game has finitely many mixed Nash equilibria. Given this assumption, the following result entails that, if there is a solution to  $\text{SE\_LP}$  for some mixed support, it will always be unique. In this way we preclude that solving  $\text{SE\_LP}$  for some support  $\hat{B}$  yields a solution that does not have the behavioural independence property, even if there is another solution to  $\text{SE\_LP}$  for  $\hat{B}$  that does.

---

**Algorithm 3** BNE\_SE\_LP\_ALL( $G$ )

---

**Input:**  $G = (\{1, 2\}, \Phi_1, \Phi_2, \gamma_1, \gamma_2)$  is a Boolean game with finitely many mixed NE.

**Output:** all behavioural equilibria  $\beta^*$  of  $G$ , where  $\beta_i : \Phi_i \rightarrow [0, 1]$  for  $i \in \{1, 2\}$ , or NULL if none exists.

```
1:  $\hat{G} \leftarrow (\{1, 2\}, 2^{\Phi_1}, 2^{\Phi_2}, \text{UTILITY}(G))$ 
2:  $\text{behavioural\_ne} \leftarrow \text{false}$ 
3: for  $B_1 = 0$  to  $3^{|\Phi_1|}$  (sorted in increasing order of size) do
4:   for  $B_2 = 0$  to  $3^{|\Phi_2|}$  (sorted in increasing order of size) do
5:     //  $\text{support}_i$  represents a behavioural support for player  $i$ . The  $j$ th trinary bit
     // of  $B_i$  is 1 if  $b_i(\Phi_i[j]) = 0$ ; 2 if  $b_i(\Phi_i[j]) = 1$ ; and 0 otherwise.
6:     for  $i = 1$  to 2 do
7:        $\hat{B}_i \leftarrow \text{MIXED\_SUPPORT}(B_i)$ 
8:       if there is a solution  $(\sigma_1, \sigma_2, x_1, x_2)$  to  $\text{SE\_LP}(\hat{G}, \hat{B}_1, \hat{B}_2)$  then
9:          $(\beta_1, \beta_2) \leftarrow (\text{BEHAVIOURAL}(\sigma_1), \text{BEHAVIOURAL}(\sigma_2))$ 
10:        if  $\beta_1 \neq \text{and } \beta_2 \neq$  then
11:           $\text{behavioural\_ne} \leftarrow \text{true}$ 
12:          output  $(\beta_1, \beta_2)$ 
13:          // ‘Break’ at this point in BNE_SE_LP_SOME.
14: if not  $\text{behavioural\_ne}$  then
15:   output “No behavioural Nash equilibria.”
```

---

**Proposition 10.** *If a two-player game has two mixed equilibria with the same support, then it has infinitely many mixed equilibria.*

*Sketch of proof:* Assume two-player game  $G$  has two mixed equilibria  $\sigma^* = (\sigma_1^*, \sigma_2^*)$  and  $\tau^* = (\tau_1^*, \tau_2^*)$  such that  $S_{\sigma^*} = S_{\tau^*}$ . Without loss of generality we  
425 may assume that  $\sigma_1^* \neq \tau_1^*$ . Now consider an arbitrary  $\lambda \in [0, 1]$ , and let  $\sigma_1^\lambda$  be  
defined such that  $\sigma_1^\lambda(v_i) = \lambda \cdot \sigma_1(v_i) + (1 - \lambda) \cdot \tau_1(v_i)$ , for each pure strategy  $v_i$ .  
By the Indifference Principle for mixed strategies it then follows that  $(\sigma_1^\lambda, \sigma_2^*)$  is  
a mixed equilibrium as well. Hence,  $G$  has infinitely many mixed equilibria.  $\square$

Thus, if there is more than one solution to SE\_LP for some support vector  
430  $S = (S_1, S_2)$ , then the respective game has infinitely many mixed equilibria.

Our second support enumeration algorithm, BNE\_SE\_NUM, is depicted in Algorithm 4. It differs from BNE\_SE\_LP in that it solves a non-linear program SE\_NUM, represented in Figure 3, instead of SE\_LP. In a similar way as SE\_LP captures the



$$\sum_{v_{-i} \in 2^{\Phi-i}} \left( \prod_{p \in v_{-i}} \beta_i(p) \cdot \prod_{p \in 2^{\Phi-i} \setminus v_{-i}} (1 - \beta_i(p)) \right) \cdot u_i(v_i, v_{-i}) = x_i \quad \forall i \in \{1, 2\}, v_i \in \hat{B}_i \quad (6)$$

$$\sum_{v_{-i} \in 2^{\Phi-i}} \left( \prod_{p \in v_{-i}} \beta_i(p) \cdot \prod_{p \in 2^{\Phi-i} \setminus v_{-i}} (1 - \beta_i(p)) \right) \cdot u_i(v_i, v_{-i}) \leq x_i \quad \forall i \in \{1, 2\}, v_i \in 2^{\Phi_i} \setminus \hat{B}_i \quad (7)$$

$$\beta_i(p) \geq 0 \quad \forall i \in \{1, 2\}, B_i(p) = 2 \quad (8)$$

$$\beta_i(p) \leq 1 \quad \forall i \in \{1, 2\}, B_i(p) = 2 \quad (9)$$

$$\beta_i(p) = 1 \quad \forall i \in \{1, 2\}, B_i(p) = 1 \quad (10)$$

$$\beta_i(p) = 0 \quad \forall i \in \{1, 2\}, B_i(p) = 0 \quad (11)$$

Figure 3: Non-linear feasibility program `SE_NUM` for behavioural supports  $B = (B_1, B_2)$ .

conditions of the mixed Indifference Principle, `SE_NUM` reflects the conditions of the Behavioural Indifference Principle. As `SE_NUM` solves towards values  $\beta_i(p)$  for propositional variables  $p$  directly, `BNE_SE_NUM` can moreover dispense with the check whether the solutions to `SE_NUM` satisfy the behavioural independence property.

Thus, we can use a high-performance numerical solver to find equilibria in behavioural strategies of the form  $\beta_i: \Phi_i \rightarrow [0, 1]$  immediately. Again, we distinguish between `BNE_SE_NUM_ALL` and `BNE_SE_NUM_SOME`, which compute all and some behavioural equilibria in a given weighted Boolean game, respectively. As the constraints of `SE_NUM` are not linear, however, for `BNE_SE_NUM` we do no longer have a runtime polynomial in the number of outcomes.

#### 4.3. Weakly Dominated Strategies and the Support Elimination Heuristic

A well-known computational bottleneck of the original support enumeration algorithms for finding mixed equilibria is the sheer number of supports it iterates through. To this end, Porter et al. [24] proposed a heuristic based on the concept of eliminating, what they called, *conditionally dominated strategies*, a concept that is closely related to that of strict dominance between strategies.

Here, we propose a similar heuristic, which, however, is founded on the concept of weak dominance. The key observation is that, if a given mixed support contains a pure strategy that is *weakly dominated within this support*,

---

**Algorithm 4** BNE\_SE\_NUM\_ALL( $G$ )

---

**Input:**  $G = (\{1, 2\}, \Phi_1, \Phi_2, \gamma_1, \gamma_2)$  is a Boolean game with finitely many mixed equilibria, where  $\Phi_1 = \{p_1^1, \dots, p_1^k\}$  and  $\Phi_2 = \{p_2^1, \dots, p_2^\ell\}$ .

**Output:** all behavioural equilibria  $\beta^*$  of  $G$ , where  $\beta_i^* : \Phi_i \rightarrow [0, 1]$  for  $i \in \{1, 2\}$ , or NULL if none exists.

```

1:  $\hat{G} \leftarrow (\{1, 2\}, 2^{\Phi_1}, 2^{\Phi_2}, \text{UTILITY}(G))$ 
2:  $\text{behavioural\_ne} \leftarrow \text{false}$ 
3: for  $B_1 = 0$  to  $3^{|\Phi_1|}$  (sorted in increasing order of size) do
4:   for  $B_2 = 0$  to  $3^{|\Phi_2|}$  (sorted in increasing order of size) do
5:     for  $i = 1$  to  $2$  do
6:        $\hat{B}_i \leftarrow \text{MIXED\_SUPPORT}(B_i)$ 
7:       if there is a solution  $(\beta_1(p_1^1), \dots, \beta_1(p_1^k), \beta_1(p_2^1), \dots, \beta_1(p_2^\ell), x_1, x_2)$  to
          $\text{SE\_NUM}(\hat{G}, \hat{B}_1, \hat{B}_2)$  then
8:          $\text{behavioural\_ne} \leftarrow \text{true}$ 
9:         output  $(\beta_1, \beta_2)$ 
10:      // 'Break' at this point in BNE_SE_NUM_SOME.
11: if not  $\text{behavioural\_ne}$  then
12:   output "No behavioural Nash equilibria."

```

---

that strategy will receive zero probability in every mixed equilibrium with that support. Hence, there are no mixed equilibria with that support, and the latter can duly be ignored in our search for behavioural equilibria.

Formally, given a mixed support  $S = (S_1, S_2)$ , we say that a pure strategy  $v_i \in S_i$  for player  $i$  is **weakly dominated with respect to**  $S_{-i}$  if there is some  $v'_i \in 2^{\Phi_i}$  such that  $u(v'_i, w_{-i}) \geq u(v_i, w_{-i})$  for all  $w_{-i} \in S_{-i}$ , and  $u(v'_i, w'_{-i}) > u(v_i, w'_{-i})$  for some  $w'_{-i} \in S_{-i}$ . Thus, the following corollary of the Indifference Principle provides the formal basis for our heuristic.

**Proposition 11.** *Let  $S = (S_1, S_2)$  be a mixed support and  $v_i$  a pure strategy for player  $i$  that is weakly dominated with respect to  $S_{-i}$ . Then, there is no mixed equilibrium  $\sigma^* = (\sigma_1^*, \sigma_2^*)$  with  $v_i \in S_{\sigma_i^*}$  and  $S_{\sigma_{-i}^*} = S_{-i}$ .*

*Proof:* Assume for contradiction that there is some mixed equilibrium  $\sigma^* = (\sigma_1^*, \sigma_2^*)$  with  $v_i \in S_{\sigma_i^*}$  and  $S_{\sigma_{-i}^*} = S_{-i}$ . Having assumed that  $v_i$  is weakly dominated in  $S$ , there is also some  $v'_i \in S_i$  such that  $u(v'_i, w_{-i}) \geq u(v_i, w_{-i})$  for all  $w_{-i} \in S_{-i}$ , and  $u(v'_i, w'_{-i}) > u(v_i, w'_{-i})$  for some  $w'_{-i} \in S_{-i}$ . For every

$p$	$q$	$r$	$p$	$q$	$r$	$p$	$q$	$r$	$p$	$q$	$r$	$p$	$q$	$r$	
$B_1^1$	0	0	0	$B_1^7$	0	2	0	$B_1^{13}$	1	0	2	$B_1^{19}$	2	0	0
$B_1^2$	0	0	1	$B_1^8$	0	2	1	$B_1^{14}$	1	1	0	$B_1^{20}$	2	0	1
$B_1^3$	0	0	2	$B_1^9$	0	2	2	$B_1^{15}$	1	1	2	$B_1^{21}$	2	0	2
$B_1^4$	0	1	0	$B_1^{10}$	1	0	0	$B_1^{16}$	1	2	0	$B_1^{22}$	2	1	0
$B_1^5$	0	1	1	$B_1^{11}$	1	0	0	$B_1^{17}$	1	2	1	$B_1^{23}$	2	1	1
$B_1^6$	0	1	2	$B_1^{12}$	1	0	1	$B_1^{18}$	1	2	2	$B_1^{24}$	2	1	2

Table 1: The 27 behavioural support functions for a player controlling variables  $p$ ,  $q$ , and  $r$ .

mixed strategy  $\sigma_{-i}$  with  $S_{\sigma_{-i}} = S_{-i}$ , we have  $\sigma_{-i}(w'_{-i}) > 0$ . It then follows  
470 that  $\sum_{w_{-i} \in 2^{\Phi_{-i}}} \sigma_{-i}(w_{-i}) u_i(v_i, w_{-i}) < \sum_{w_{-i} \in 2^{\Phi_{-i}}} \sigma_{-i}(w_{-i}) u_i(v'_i, w_{-i})$ , which is  
at variance with the Indifference Principle.  $\square$

Thus, when searching for mixed equilibria by mixed support enumeration,  
if a mixed support  $S = (S_i, S_{-i})$  contains a pure strategy  $v_i$  that is weakly  
dominated with respect to it, one can ignore any mixed support  $(S'_i, S_{-i})$  where  
475  $v_1 \in S'_1$ .

For an example, let player 1 control propositional variables  $p$ ,  $q$ , and  $r$ . Her  
possible behavioural supports are then enumerated in Table 1. Now suppose  
that  $pq\bar{r}$  is weakly dominated with respect to  $\hat{B}_2$ , where  $B_2$  is a behavioural  
support function for player 2. Now consider  $B_1^{14} = (1, 1, 0)$ , for which  $\hat{B}_1^{14} =$   
480  $\{pq\bar{r}\}$ . By Proposition 11, there is no behavioural equilibrium with support  
 $(B_1^{14}, B_2)$ . However, any behavioural strategy  $\beta_i$  with  $B_{\beta_i}(p) = 2$ ,  $B_{\beta_i}(q) = 1$   
and  $B_{\beta_i}(r) = 0$ , will also put positive probability on  $pq\bar{r}$ . Hence, neither is there  
a behavioural equilibrium with support  $(B_1^{22}, B_2)$ , which can thus be ignored  
accordingly. Similarly, one will find no behavioural equilibria with behavioural  
485 supports  $(B_1^{22}, B_2)$ ,  $(B_1^{15}, B_2)$ ,  $(B_1^{16}, B_2)$ ,  $(B_1^{18}, B_2)$ ,  $(B_1^{24}, B_2)$ ,  $(B_1^{25}, B_2)$ , or  
 $(B_1^{27}, B_2)$  either.

For both BNE\_SE\_LP and BNE\_SE\_NUM we have implemented this procedure  
and we refer to as the **support elimination heuristic**. This heuristic main-  
tains a list recording which pure strategies of each player are weakly dominated

490 with respect to which behavioural support of the other player. Initially this list  
 is empty. Both BNE\_SE\_LP and BNE\_SE\_NUM iterate through all behavioural sup-  
 ports  $(B_1, B_2)$ . The heuristic is called in every iteration before checking whether  
 there is a solution to SE\_LP( $\hat{G}, \hat{B}_1, \hat{B}_2$ ), respectively, to SE\_NUM( $\hat{G}, \hat{B}_1, \hat{B}_2$ ). In-  
 intuitively, the heuristic checks whether the list contains a records for  $B_i$  and some  
 495  $v_{-i} \in B_{-i}$  for either player  $i$ . Any such a record indicates that  $v_{-i}$  is weakly  
 dominated with respect to  $\hat{B}_i$  and that there is no behavioural equilibrium with  
 support  $(\hat{B}_1, \hat{B}_2)$ . The iteration is therefore then broken off and the algorithm  
 continues with the next behavioural support. If there is no such entry, the  
 heuristic computes all pure strategies that are weakly dominated with respect  
 500 to either  $\hat{B}_1$  and  $\hat{B}_2$ , *if this has not been done before in an earlier iteration*  
*already*, and adds corresponding records to the list. It then checks if among  
 the new records there are any for some pure strategy  $v_1$  or  $v_2$ . If so, iteration  
 is broken off and the algorithm continues with the next behavioural support.  
 Otherwise, the algorithm proceeds to the next line and tries to find a solution to  
 505 SE\_LP( $\hat{G}, \hat{B}_1, \hat{B}_2$ ), in the case of BNE\_SE\_LP, and to SE\_NUM( $\hat{G}, \hat{B}_1, \hat{B}_2$ ), in the  
 case of BNE\_SE\_NUM.

#### 4.4. Finding Behavioural Equilibria via Correlated Profiles: BNE\_CORR\_NUM

Our final approach is to exploit the well-known fact that correlated equilibria  
 can be found using a single linear feasibility program. Observe that the very  
 510 definition of correlated equilibrium (see page 18) almost immediately yields a  
 linear feasibility program when the right-hand term of the defining inequality CE  
 is brought to the left, and additional constraints are included that assure that  $\pi$   
 is a probability distribution over  $2^\Phi$ . By replacing the linear constraints by  
 non-linear ones involving propositional variables immediately, we obtain the  
 515 non-linear program CORR\_NUM depicted in Figure 4. In virtue of Corollary 7,  
 CORR\_NUM can only have *behavioural* profiles as solutions. We can find these  
 solutions by invoking a numerical solver, and the algorithm that does so, we  
 refer to by BNE\_CORR\_NUM.

The crucial difference between correlated and mixed strategies is that with  
 520 the former, the players can correlate the randomisations over their pure strate-  
 gies, whereas they are assumed to randomise independently when playing mixed  
 strategies. In Boolean games, mixed and behavioural strategies are related in  
 a much similar way. With mixed strategies players can correlate the probabili-  
 ties they assign to their own propositional variables, whereas with behavioural  
 525 strategies the randomisations over their propositional variables are independent.  
 Correlated equilibria can efficiently be found by solving a linear program. Al-  
 gorithms for computing mixed equilibria are computationally harder and have  
 to deal with the independence of mixed strategies played by different players.  
 Algorithms for computing behavioural equilibria, moreover, must not only deal  
 530 with the independence of the behavioural strategies played by different players,  
 but also with the independence of the randomisations over the propositional  
 variables of a single player. As they search for behaviourally independent mixed  
 equilibria, our first two approaches towards computing behavioural equilibria  
 could be seen as dealing with the two types of independence separately, whereas  
 535 BNE\_CORR\_NUM deals with both types of non-correlation simultaneously.

Another advantage of this approach, which BNE\_CORR\_NUM shares with  
 BNE\_SE\_NUM, is that it is not restricted to two-player games or games with  
 only a finite number of mixed equilibria. Similarly, BNE\_CORR\_NUM can also be  
 used to find solutions in games with an infinite number of mixed equilibria, in  
 540 particular, also in the non-trivial class of degenerate 01-Boolean games. The  
 numerical solver we invoke in our experiments (see Section 5), however, outputs  
 at most one solution. Accordingly, we can use this algorithm only to find some  
 behavioural equilibrium and cannot guarantee that we have found them all.

## 5. Experimental Evaluation

545 We conducted a detailed experimental study comparing the run-time perfor-  
 mance of BNE\_NAIVE, BNE\_SE\_LP, BNE\_SE\_NUM, and BNE\_CORR\_NUM, the four

$$\begin{aligned}
& \sum_{v_{-i} \in 2^{\Phi_{-i}}} \left( \left( \prod_{p \in (v_i)} \beta_i(p) \cdot \prod_{p \in 2^{\Phi} \setminus (v_i)} (1 - \beta_i(p)) \cdot \prod_{p \in (v_{-i})} \beta_i(p) \cdot \prod_{p \in 2^{\Phi} \setminus (v_{-i})} (1 - \beta_i(p)) \right) \cdot u_i(v_{-i}, v_i) \right) \\
& - \sum_{v_{-i} \in 2^{\Phi_{-i}}} \left( \left( \prod_{p \in (v_i)} \beta_i(p) \cdot \prod_{p \in 2^{\Phi} \setminus (v_i)} (1 - \beta_i(p)) \cdot \prod_{p \in (v_{-i})} \beta_i(p) \cdot \prod_{p \in 2^{\Phi} \setminus (v_{-i})} (1 - \beta_i(p)) \right) \cdot u_i(v_{-i}, w_i) \right) \geq 0 \\
& \qquad \qquad \qquad \forall i \in N, \forall v_i, w_i \in 2^{\Phi_i} \quad (12) \\
& \qquad \qquad \qquad \beta_i(p) \geq 0 \quad \forall i \in \{1, 2\}, p \in \Phi_i \quad (13) \\
& \qquad \qquad \qquad \beta_i(p) \leq 1 \quad \forall i \in \{1, 2\}, p \in \Phi_i \quad (14)
\end{aligned}$$

Figure 4: Non-linear feasibility program CORR\_NUM for finding behavioural equilibria via correlated profiles.

approaches towards computing behavioural equilibria in two-player weighted Boolean games presented in the previous section.

In our experiments, we measured the average running time of our algorithms  
550 on a library of 959 weighted Boolean games with a finite number of mixed  
equilibria. These games were obtained by first randomly generating about 40,000  
weighted Boolean games and then selecting from these those with a finite number  
of mixed equilibria (cf. Section 5.2). In addition, we also built two other libraries,  
one consisting of 959 weighted Boolean games and the other containing 440  
555 01-Boolean games. The games in these latter two libraries are not constrained to  
have a finite number of mixed equilibria, and on them we measured the average  
running time of BNE\_SE\_NUM and BNE\_CORR\_NUM to find a first behavioural  
equilibrium.

With our experiments we aim to obtain a clear picture of the relative advan-  
560 tages of the three different approaches towards computing behavioural equilibria  
in terms of average running time. In addition, we measure the impact of heuris-  
tics on the performance of our algorithms. Accordingly, we want to see how  
the optimisations of computing mixed equilibria as provided by the GAMBIT  
package affect the performance of BNE\_NAIVE, and to what extent the elimination  
565 heuristic accelerates the computations of BNE\_SE\_LP and BNE\_SE\_NUM. Finally,  
we want to see if—when put to use for the purposes of finding behavioural

equilibria in two-player games—numerical solvers can compete in practice with state-of-the-art LP solvers. As the constraints in `SE_LP` are linear for two-player games only, a positive answer to the latter question would improve the prospects  
570 for computing behavioural equilibria in many-player games.

To these ends, we implemented our algorithms in the programming language Python. To solve the linear program `SE_LP` invoked by `BNE_SE_LP` we used the LP solvers provided by the PuLP package [26], which was specially developed for Python. To perform the numerical computations of `SE_NUM` invoked by  
575 `BNE_SE_NUM`, we chose the default numerical solver *SLSQP* from the SciPy package [27]. We ran all our experiments on an Intel<sup>®</sup> Core<sup>®</sup> i5-6500 machine with 3.20GHz on each of the four cores, and cache and RAM memory of 6144 KB and 7.7 GB, respectively.

Our experimental results section concludes with a brief study of a question  
580 that is somewhat tangential to the main concerns of the section. Using the three game libraries that we use in our experiments, we investigate the frequency with which behavioural equilibria exist in games. We find that, in our experimental libraries at least, behavioural equilibria are rather common, which is an encouraging indication of their applicability.

### 585 5.1. Our Experiments

Our experiments are divided into two groups: Group I and Group II. In the first, we measure the *average running times* of our algorithms `BNE_NAIVE_ALL`, `BNE_SE_LP_ALL`, and `BNE_SE_NUM_ALL` to find *all* behavioural equilibria in a given weighted Boolean game (Section 5.3). In the second, we compare the *aver-*  
590 *age running times* of `BNE_NAIVE_SOME`, `BNE_SE_LP_SOME`, `BNE_SE_NUM_SOME`, as well as of `BNE_CORR_NUM` to find *a first* behavioural equilibrium (Section 5.4).<sup>3</sup>

---

<sup>3</sup>The SLSQP solver from the SciPy package only gives one solution. Accordingly, our implementation of `BNE_CORR_NUM` is not suited as an algorithm for finding *all* behavioural equilibria. The same concern, however, does not apply to `BNE_SE_NUM`, whose implementation also invokes the SLSQP solver every time it calls `SE_NUM`. Observe, that by virtue of Proposition 10, one need to find at most one behavioural equilibrium in each support in order to find all behavioural equilibria of the game, *provided that one restricts attention to games with*

In both groups of experiments, we run BNE\_NAIVE, BNE\_SE\_LP, and BNE\_SE\_NUM with (experiments I.1 and II.1) and without attendant heuristics (experiments I.2 and II.2). Without heuristics, we let BNE\_NAIVE invoke the support enumeration  
595 that solves SE\_LP for every *mixed support* when it calls MIXED\_EQUILIBRIA to compute all mixed equilibria. The performance of BNE\_NAIVE can be enhanced by invoking *EnumMixed* provided by the GAMBIT to this purpose instead. The support enumeration algorithms BNE\_SE\_LP and BNE\_SE\_NUM can be optimised by implementing the support elimination heuristic as described in Section 4.3.

600 To obtain a proper understanding of the computational ramifications of the essential differences between our algorithms, we first tested our algorithms without any heuristics. In this way, we aim to chart the benefits for BNE\_SE\_LP and BNE\_SE\_NUM of there being only  $3^{|\Phi|}$  behavioural supports to iterate through. In experiments I.2 and II.2 we compare the computational impact of the different  
605 heuristics that are available for our algorithms.

As each of BNE\_NAIVE\_ALL, BNE\_NAIVE\_SOME, BNE\_SE\_LP\_ALL, BNE\_SE\_LP\_SOME, and BNE\_SE\_NUM\_ALL requires games with only a finite number of mixed equilibria as input, experiments I.1, I.2, II.1, and II.2 have duly been performed on games from Library 1. However, when searching for a first behavioural equilibrium,  
610 BNE\_CORR\_NUM and BNE\_SE\_NUM\_SOME are not so constrained. Hence, we also performed a third set of experiments, II.3, in which the performance of these algorithms on games from Libraries 2 and 3 are compared, that is, on weighted Boolean games and 01-Boolean games with a potentially infinite number of mixed equilibria.<sup>4</sup>

615 Tables 2 and 3 give a schematic overview of our experiments.

---

*a finite number of mixed equilibria.* We observe the latter restriction in our experiments on library 1.

<sup>4</sup>As virtually all non-trivial 01-Boolean games are degenerate, they tend to have an infinite number of mixed equilibria. Testing BNE\_NAIVE and BNE\_SE\_LP on non-degenerate 01-Boolean games only is therefore bound to lead to rather distorted results. No such concerns apply to BNE\_SE\_NUM\_SOME and BNE\_CORR\_NUM as they do not require games with a finite number of mixed equilibria as input.



	algorithm	heuristics	Library
I.1	BNE_NAIVE_ALL	none	Library 1
	BNE_SE_LP_ALL	none	
	BNE_SE_NUM_ALL	none	
I.2	BNE_NAIVE_ALL	GAMBIT: <i>EnumMixed</i>	Library 1
	BNE_SE_LP_ALL	support elimination	
	BNE_SE_NUM_ALL	support elimination	

Table 2: Group I of experiments: searching for all behavioural Nash equilibria. Library 1: weighted Boolean games with a finite number of mixed equilibria.

### 5.2. Sampling of Weighted Boolean Games

For the purposes of our experiments, we have built three libraries of two-player weighted Boolean games:

**Library 1:** 959 weighted Boolean games with a finite number of mixed equilibria

620 **Library 2:** 950 weighted Boolean games with any number of mixed equilibria

**Library 3:** 440 01-Boolean games with any number of mixed equilibria

To build Library 1, we randomly generated about 40,000 weighted Boolean games and used GAMBIT [9] to compute the *extreme mixed Nash equilibria*, of which there are only a finitely many.<sup>5</sup> We checked whether any two of  
625 these extreme mixed Nash equilibria had the same support for one of the players. If so, by Proposition 10, the game in question has an infinite number of equilibria and was duly eliminated from the library. To generate the initial 40,000 weighted Boolean games, we developed a weighted Boolean game generator, which randomly assigns the control of propositional variables to the players and  
630 produces a random Boolean preference net  $\Gamma_i$ . To the latter end we deployed the Boolean formula generator *CNFGen* [28]. Our game generator takes three inputs: a number of variables (ranging from 2 to 8), a number of clauses (ranging from 1 to 10), and the clause width of the goal formulas (ranging from 1 to 10). It outputs a random weighted Boolean game selected uniformly at random according

<sup>5</sup>The **extreme mixed equilibria** of a strategic game are those mixed equilibria that cannot be defined as a convex combination of any other mixed equilibria in the game. Every finite strategic game has only a finite number of extreme equilibria.

	algorithm	heuristics	libraries
II.1	BNE_NAIVE_SOME	none	Library 1
	BNE_SE_LP_SOME	none	
	BNE_SE_NUM_SOME	none	
	BNE_CORR_NUM	none	
II.2	BNE_NAIVE_SOME	GAMBIT: <i>EnumMixed</i>	Library 1
	BNE_SE_LP_SOME	support elimination	
	BNE_SE_NUM_SOME	support elimination	
II.3	BNE_SE_NUM_SOME	none	Library 2
	BNE_SE_NUM_SOME	support elimination	Library 3
	BNE_CORR_NUM	none	

Table 3: Group II of experiments: searching for some (the first) behavioural equilibrium. Library 1: weighted Boolean games with a finite number of mixed equilibria. Library 2: weighted Boolean games with any number of mixed equilibria. Library 3: 01-Boolean games with any number of mixed equilibria.

635 to the parameters. Observe that due to these parameter setting our experiments are restricted to weighted Boolean games with no more than 8 propositional variables. Accordingly, a player can have at most 128 pure strategies at her disposal in any of these games.

The games in libraries 2 and 3 are generated by the same techniques, except  
640 that there was no need to filter out games with an infinite number of mixed equilibria.

### 5.3. Experiments Results: Finding All Behavioural Equilibria

In experiment I.1, we measured the average running time of, respectively, BNE\_NAIVE\_ALL, BNE\_SE\_LP\_ALL, and BNE\_SE\_NUM\_ALL, to compute all be-  
645 havioural equilibria of weighted Boolean games drawn from Library 1. The results are depicted graphically in Figure 5. We find that BNE\_NAIVE\_ALL starts solving  $2 \times 2$  games reasonably fast at about  $\frac{1}{10}$ th of a second per game, but quickly times out for larger games. The graph shows exponential growth in the number of outcomes, which is consistent with the asymptotic time complexity of  
650 our algorithm, given that the time taken to compute the mixed Nash equilibria is exponential in the number of outcomes of the game. All existing algorithms for computing mixed Nash equilibria in the general case either have unknown

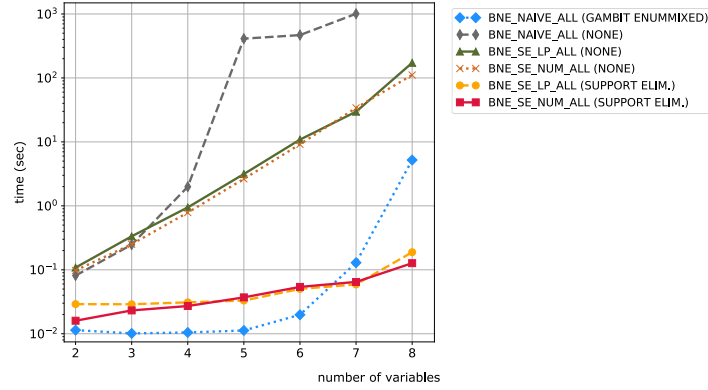


Figure 5: Experiments I.1 and I.2: the average run-time of finding all behavioural equilibria on Library 1, weighted Boolean games with a finite number of equilibria.

runtime or are exponential in the number of outcomes [29]. These two facts suggest that the computation of all mixed equilibria when MIXED\_EQUILIBRIA is called in BNE\_NAIVE\_ALL does indeed take time exponential in the number of outcomes.

The support enumeration algorithm BNE\_SE\_LP\_ALL basically only differs from BNE\_NAIVE\_ALL in that it iterates through behavioural supports, which are exponentially fewer in number than the mixed supports BNE\_NAIVE has to check. The gradient of BNE\_SE\_LP\_ALL's appears to be proportionate to the growth of the number of behavioural supports in a Boolean game as the number of variables increases. This impression is reinforced by the much similar average running times of BNE\_SE\_NUM\_ALL's on this library. We have seen that BNE\_SE\_LP\_ALL has to solve the linear program SE\_LP for an exponentially smaller number of supports than BNE\_NAIVE\_ALL. Our results seem to confirm our expectation that the number of behavioural supports being polynomial in the number of outcomes represents a major and tangible practical benefit for BNE\_SE\_LP\_ALL when computing all behavioural equilibria.

Figure 5 also depicts the results for the second set of experiments I.2. For these experiments, BNE\_NAIVE\_ALL, BNE\_SE\_LP\_ALL, and BNE\_SE\_NUM\_ALL are augmented with their respective heuristics. The graph for BNE\_NAIVE\_ALL(*EnumMixed*)

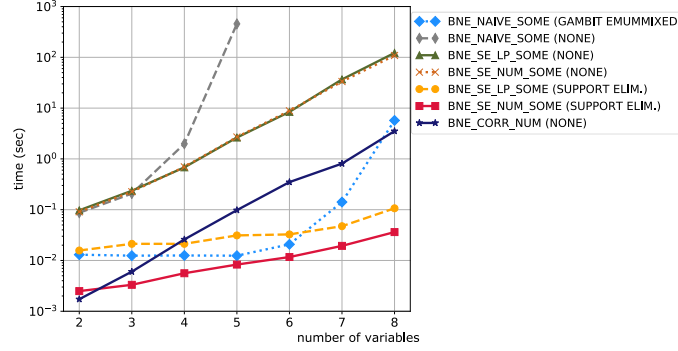


Figure 6: Experiments II.1 and II.2: The average run-time of finding the first behavioural equilibria on Library 1, weighted Boolean games with a finite number of equilibria.

shows the effectiveness of the optimisations provided by GAMBIT’s *EnumMixed* algorithm for finding all mixed equilibria. The support elimination heuristic can similarly be seen to provide a significant efficiency gain for both BNE\_SE\_LP\_ALL and BNE\_SE\_NUM\_ALL. Interestingly, the graphs of BNE\_SE\_LP\_ALL and BNE\_SE\_NUM\_ALL again virtually coincide. This indicates that the computational bottleneck of these support enumeration methods is indeed the sheer number of supports for which SE\_LP and SE\_NUM have to be solved. It also suggests that the difference between the time taken by PuLP to solve SE\_LP and that of SciPy’s *SLSQP* tool to solve SE\_NUM is negligible in comparison with the number of times these subroutines are called by BNE\_SE\_LP\_ALL and BNE\_SE\_NUM\_ALL, respectively.

Although on Boolean games with a small number of propositional variables, BNE\_NAIVE(*EnumMixed*) performs better than both support enumeration methods, our experimental data also suggest that this advantage quickly evaporates as the number of variables increases. Altogether, we venture the conclusion that the optimisations provided by the GAMBIT package for computing mixed equilibria do not offset the advantage BNE\_SE\_LP and BNE\_SE\_NUM gain by iterating through behavioural supports rather than mixed supports.

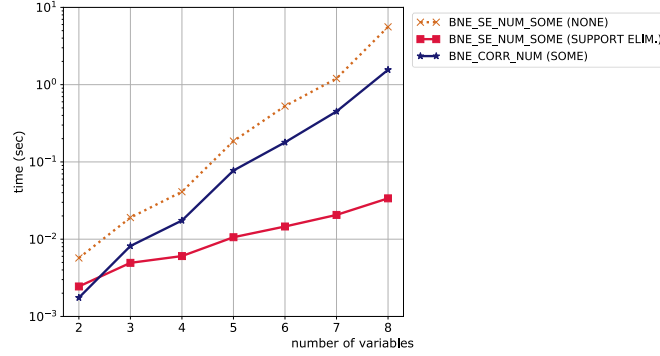


Figure 7: Experiments II.3: The average run-time of finding the first behavioural equilibria on Library 3, general weighted Boolean games.

#### 5.4. Experimental Results: Finding a First Behavioural Equilibrium

690 The results of the experiments II.1 and II.2 for finding a first behavioural equilibrium are graphically depicted in Figure 6. The first thing that stands out is that the overall picture that emerges is very similar to that for experiments I.1 and I.2 in the previous section. The fact that BNE\_SE\_LP\_SOME and BNE\_SE\_NUM\_SOME have to call the subroutines SE\_LP and SE\_NUM at most  $3^{|\Phi|}$  times gives them an exponential edge over BNE\_NAIVE\_SOME, irrespective of  
695 whether heuristics are considered or not. Again, the relative performance of the support enumeration algorithms BNE\_SE\_LP\_SOME and BNE\_SE\_NUM\_SOME is very similar. When the support elimination heuristic is applied, the small advantage of BNE\_SE\_NUM\_SOME over BNE\_SE\_LP\_SOME is noteworthy: SciPy's numerical solver *SLSQP* appears to outperform PuLP's LP solver. It is equally  
700 surprising that this difference in performance is not visible when the heuristic is dispensed with. Moreover, the data exhibit a marked and significant improvement in average running time when BNE\_SE\_LP\_SOME and BNE\_SE\_NUM\_SOME are augmented with the support elimination heuristic.

705 Algorithm BNE\_CORR\_NUM uses numerical methods to solve the non-linear program CORR\_NUM once for each weighed Boolean game. As such it is based on quite different principles to the other algorithms considered. Yet, its average

running time appears to be of a similar order to that of the support enumeration algorithms BNE\_SE\_LP\_SOME and BNE\_SE\_NUM\_SOME when run without the support elimination heuristic. Even though, with the heuristic, BNE\_SE\_LP\_SOME and BNE\_SE\_NUM\_SOME outperform BNE\_CORR\_NUM, the latter still seems to have an exponential advantage over BNE\_NAIVE\_SOME (both with and without *EnumMixed* heuristic). Due to the very definition of BNE\_CORR\_NUM, however, it seems unlikely that this advantage can directly be attributed to there being exponentially fewer behavioural supports than mixed supports.

The performance of BNE\_SE\_NUM\_SOME and BNE\_CORR\_NUM on the games from Library 1 is encouraging. As the soundness of these two algorithms does not depend on the input being a weighted Boolean game with a finite number of mixed equilibria, this may suggest their feasibility for finding a first behavioural equilibrium also in games without this restriction. In the last sets of experiments II.3, we therefore compare the average running times of BNE\_SE\_NUM\_SOME and BNE\_CORR\_NUM on games from Libraries 2 and 3. The experiments for general weighted Boolean games, as depicted in Figure 7, are reassuring: the run-time behaviour of both BNE\_SE\_NUM\_SOME and BNE\_CORR\_NUM seems to be hardly, if at all, affected by the constraint on the input games being lifted. It should be noted, however, that the games from Library 2 are more likely to have pure equilibria, which are generally easier to find. In future experiments, we hope to investigate the robustness of these findings when larger games are considered.

In the last set of experiments, we investigated 01-Boolean games from Library 3. Our data, as depicted in Figure 8, show a markedly faster computation time for BNE\_SE\_NUM\_SOME on this class of games, whereas the results for BNE\_CORR\_NUM are comparable with those on general weighted Boolean games. We suspect that this is due to the high likelihood of a randomly generated two-player 01-Boolean game having a *pure equilibrium*.<sup>6</sup> Moreover, as in our

---

<sup>6</sup>If payoffs to outcomes are chose uniformly at random and independently for both players, the probability of an outcome in which both players receive payoff 1 is one in four. Any such outcome would be a Nash equilibrium!

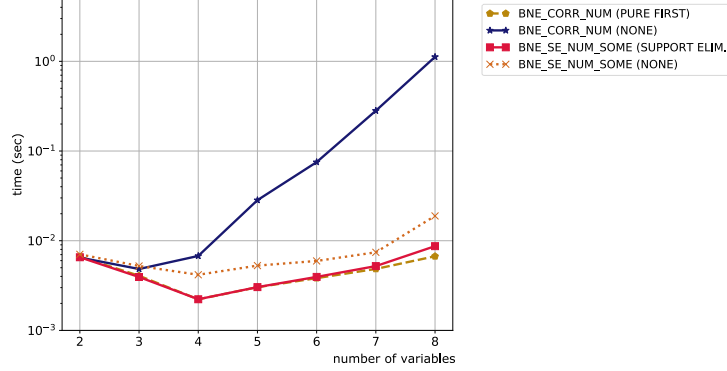


Figure 8: Experiments II.3: The average run-time of finding the first behavioural equilibria on Library 3, 01-Boolean games.

support enumeration algorithms, supports are sorted by increasing order of size, BNE\_SE\_NUM\_SOME will find pure equilibria first. As no such assumption can be made for BNE\_CORR\_NUM, this would explain BNE\_SE\_NUM\_SOME's superior performance. To test this hypothesis, we augmented BNE\_CORR\_NUM with a preprocessing procedure that searches for pure equilibria first. We found that the resulting algorithm BNE\_CORR\_NUM(PURE FIRST) has almost identical average run-times as BNE\_SE\_NUM with the support elimination heuristic, corroborating our conjecture.

### 5.5. Frequency of Behavioural Nash Equilibria

We conclude this section with an issue that is somewhat tangential to the main concerns of efficiently computing behavioural strategies, but that is nevertheless very relevant if one is interested in actually using behavioural strategies. The question is what proportion of weighted Boolean games actually have behavioural equilibria. We know that not all games have behavioural equilibria: but if we then discovered, for example, that behavioural equilibria are in fact very scarce, then that would greatly reduce their appeal as a practical solution concept.

To answer this question, we studied the games in the three libraries used in the experiments described above. For each game, we checked for the existence of a behavioural equilibrium: our findings are summarised in Table 4 (the 100% statistics for Library 3 are due to the high probability of randomly generating a

Library	# games with PNE		# games with BNE	
Library 1	818	(85.3%)	903	(94.2%)
Library 2	917	(96.5%)	936	(98.5%)
Library 3	440	(100%)	440	(100%)

Table 4: Frequency of pure and behavioural equilibria in Libraries 1, 2, and 3.

01-Boolean game with a pure equilibrium). As this table shows for Libraries 1 and 2, behavioural equilibria occur more frequently than pure Nash equilibria. This are encouraging results, although more work is needed to get precise understanding of the distribution of behavioural equilibria.

## 6. Summary, Conclusions, and Future Research

To date, relatively little research has considered the use of randomised strategies in (weighted) Boolean games. We have begun to rectify this omission. Mixed strategies in weighted Boolean games have prohibitive space requirements, motivating our introduction of behavioural strategies, in which players randomise at the level of individual variables. We have presented definitions for mixed and behavioural strategies for weighted Boolean games. Our main theoretical and algorithmic results are as follows: we have completely characterised the conditions under which a mixed strategy has an equivalent behavioural strategy; proven several theorems about behavioural Nash equilibria in Boolean games, including an analog to the Indifference Principle; and presented several algorithms and a complexity result for important computational problems in Boolean games.

Our theoretical results suggest three approaches towards computing behavioural equilibria in weighted Boolean games. The first approach is a naive one, the second is based on support enumeration methods, and the third is inspired by the concept of correlated equilibrium. On this basis, we formulated and implemented seven algorithms, each tailored to either computing all behavioural equilibria or finding a some behavioural equilibrium. In an extensive experimental study on two-player games, we found that in both settings, the naive algorithms are clearly outperformed by those based on the other two



780 approaches. Our experiments also showed that a support elimination heuristic  
 can significantly enhance the performance of the support enumeration algorithms,  
 whereas for the naive algorithms they suggest that optimising the computation  
 of mixed equilibria provides relief on small games only.

Our support enumeration algorithms we implemented both with numerical  
 785 and LP solvers to find behavioural equilibria with a given support. In our  
 experiments, the performance of the algorithms using the numerical solvers  
 was, quite surprisingly, at least comparable to that of the algorithms using  
 LP solvers. This observation is encouraging as it offers a perspective on the  
 computation of behavioural equilibria in many-player games, where the conditions  
 790 of the Indifference Principle can no longer be represented by a linear feasibility  
 program. An important question in this context is, however, how well the  
 algorithms using numerical solvers scale as the size of the games increases.  
 Similar considerations apply to the “correlated” algorithm BNE-CORR-NUM,  
 which also invokes a numerical solver.

795 Some of the algorithms we have presented for finding behavioural Nash equi-  
 libria require that the weighted Boolean game taken as input has finitely many  
 mixed Nash equilibria. While only degenerate games have infinitely many equi-  
 libria, it should be noted that degeneracy is far more common among 01-Boolean  
 games than among general strategic games. Future work may include producing  
 800 further algorithms for this degenerate case, which our experiments suggest may  
 involve the use of numerical methods. Another approach to dealing with general  
 weighted Boolean games could exploit the fact that every strategic game has  
 a finite number of *extreme* equilibria (cf, footnote 5). An easily checkable  
 characterisation of behavioural equilibria in terms of the *extreme* mixed equilibria  
 805 of a weighted Boolean game, could form the basis of algorithms for computing  
 behavioural equilibria in a similar way as the behavioural independence property  
 does in this paper.

The connection between weighted Boolean games with imperfect recall and  
 extensive-form games, as delineated in footnote 2, may furthermore provide an  
 810 important source of algorithms and complexity results for both of these models.

A similar remark applies to Nash equilibria in stationary strategies for games played on graphs (also see [30]).<sup>7</sup>

Finally, as behavioural equilibria are not guaranteed to exist, establishing the computational complexity of the decision problem whether a weighted Boolean game has a behavioural equilibrium is also an interesting question, which we had to leave open.

*Acknowledgements.* Paul Harrenstein and Michael Wooldridge were supported by the European Research Council via ERC Advanced Investigator grant 291528 (“RACE”) at Oxford. Jonathan Philpott is currently employed at VITOL, London. The authors are also indebted to Dima Pasechnik and Edith Elkind for fruitful discussions about previous versions of this paper and four anonymous reviewers for the SR 2017 workshop for helpful comments.

## References

- [1] P. Harrenstein, W. van der Hoek, J.-J. Meyer, C. Witteveen, Boolean games, in: J. van Benthem (Ed.), *Proceeding of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, 2001, pp. 287–298.
- [2] E. Bonzon, M. Lagasquie, J. Lang, B. Zanuttini, Boolean games revisited, in: *Proceedings of the Seventeenth European Conference on Artificial Intelligence (ECAI-2006)*, 2006, pp. 265–269.
- [3] E. Bonzon, M.-C. Lagasquie-Schiex, J. Lang, B. Zanuttini, Compact preference representation and Boolean games, *Autonomous Agents and Multi-Agent Systems* 18 (1) (2009) 1–35. doi:10.1007/s10458-008-9040-2.
- [4] M. Wooldridge, U. Endriss, S. Kraus, J. Lang, Incentive engineering for Boolean games, *Artificial Intelligence* 195 (2013) 418–439. doi:10.1016/j.artint.2012.11.003.

---

<sup>7</sup>The authors thank one of the reviewers for the Strategic Reasoning Workshop 2018 for this suggestion.

- [5] J. Grant, S. Kraus, M. Wooldridge, I. Zuckerman, Manipulating Boolean games through communication, in: T. Walsh (Ed.), Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI-2011), 2011, pp. 210–215. doi:10.5591/978-1-57735-516-8/IJCAI11-046.
- [6] P. E. Dunne, S. Kraus, W. van der Hoek, M. Wooldridge, Cooperative Boolean games, in: Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2008), Estoril, Portugal, 2008, pp. 1015–1022. doi:10.1145/1402298.1402363.
- [7] M. Mavronicolas, B. Monien, K. W. Wagner, Weighted boolean formula games, in: Algorithms, Probability, Networks, and Games—Scientific Papers and Essays Dedicated to Paul G. Spirakis on the Occasion of His 60th Birthday, 2015, pp. 49–86. doi:10.1007/978-3-319-24024-4\_6.
- [8] E. Iarovski, L. Ong, EGuaranteeNash for Boolean games is NEXP-hard, in: Proceedings of the Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR-2014), 2014, pp. 210–217.
- [9] R. D. McKelvey, A. M. McLennan, T. L. Turocy, Gambit: Software Tools for Game Theory, Version 15.1.0 (2014).  
URL [www.gambit-project.org](http://www.gambit-project.org)
- [10] P. Harrenstein, Logic in conflict, Ph.D. thesis, Utrecht University (2004).
- [11] S. Jeong, Y. Shoham, Marginal contribution nets: A compact representation scheme for coalitional games, in: Proceedings of the Sixth ACM Conference on Electronic Commerce (EC’05), Vancouver, Canada, 2005, pp. 193–202.
- [12] E. Elkind, M. Wooldridge, Hedonic coalition nets, in: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, 2009, pp. 417–424.
- [13] M. Maschler, E. Solan, S. Zamir, Game Theory, Cambridge U.P., 2013.

- [14] R. J. Aumann, Subjectivity and correlation in randomized strategies, *Journal of Mathematical Economics* 1 (1974) 67–96. doi:10.1016/0304-4068(74)90037-8.
- [15] J. F. Nash, Equilibrium points in  $n$ -person games, *Proceedings of the National Academy of Sciences (PNAS)* 36 (1950) 48–49. doi:10.2307/1969529.
- [16] J. F. Nash, Non-cooperative games, *Annals of Mathematics* 54 (2) (1951) 286–295. doi:10.2307/1969529.
- [17] B. von Stengel, Computing equilibria for two-player games, in: R. J. Aumann, S. Hart (Eds.), *Handbook of Game Theory with Economic Applications*, Elsevier, 2002, Ch. 45, pp. 1723–1759.
- [18] B. von Stengel, Equilibrium computation for two-player games in strategic and extensive forms, in: N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani (Eds.), *Algorithmic Game Theory*, Cambridge University Press: Cambridge, England, 2007, pp. 53–78.
- [19] E. Nudelman, J. Wortman, Y. Shoham, K. Leyton-Brown, Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms, in: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004, pp. 880–887. doi:10.1109/AAMAS.2004.238.
- [20] R. D. McKelvey, A. M. McLennan, T. L. Turocy, Gambit: Software Tools for Game Theory, Version 15.1.0, [www.gambit-project.org](http://www.gambit-project.org) (2014).
- [21] B. von Stengel, A. van den Velzen, D. Talman, Computing normal form perfect equilibria for extensive two-person games, *Econometrica* 70 (2) (2002) 693–715. doi:10.1111/1468-0262.00300.
- [22] J. C. Harsanyi, Oddness of the number of equilibrium points: A new proof, *International Journal of Game Theory* 2 (1) (1973) 235–250. doi:10.1007/BF01737572.

- 895 [23] T. Abbott, D. Kane, P. Valiant, On the complexity of two-player win-lose  
games, in: Proceedings of the 46th Symposium on the Foundations of  
Computer Science (FOCS), 2005, pp. 113–122. doi:10.1109/SFCS.2005.  
59.
- [24] R. Porter, E. Nudelman, Y. Shoham, Simple search methods for finding a  
Nash equilibrium, Games and Economic Behavior 63 (2) (2008) 642–662.  
doi:<https://doi.org/10.1016/j.geb.2006.03.015>.
- 900 [25] Y. Shoham, K. Leyton-Brown, Multiagent Systems: Algorithmic, Game-  
Theoretic, and Logical Foundations, Cambridge University Press, 2008.
- [26] S. Mitchell, A. Kean, M. O’Sullivan, A. Phillips, Optimization with PuLP  
(2018).  
URL <https://www.coin-or.org/PuLP/index.html>
- [27] The SciPy Community, SciPy Reference Guide. Release 1.0.0 (2017).  
905 URL [https://docs.scipy.org/doc/scipy/reference/generated/  
scipy.optimize.minimize.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html)
- [28] M. Lauria, CNFgen Documentation. Version 0.7.1 (2016).  
URL <http://cnfgen.readthedocs.io/en/latest/>
- [29] C. Daskalakis, P. W. Goldberg, C. H. Papadimitriou, The complexity of  
910 computing a Nash equilibrium, SIAM Journal on Computing 39 (1) (2009)  
195–259. doi:10.1137/070699652.
- [30] M. Ummels, Stochastic multiplayer games: theory and algorithms, Pallas  
Publications, Amsterdam University Press, 2010.