Boolean Combinations of Weighted Voting Games

Juan Pablo Royo Sales

Universitat Politècnica de Catalunya

January 18th, 2021

Agenda

Agenda

Introduction

Basic Notions

- Based on *Boolean Combinations of Weighted Voting Games paper* ***BWVG***[1]
- It is a natural Generalization over **Weighted Voting Games**
- Intuitively is a decision making process via multiple committees
- Each committee has the authority to decide the outcome *"yes" or "no"* about an issue.
- And each committee is a WVG
- Individuals can appear in multiple committees
- Different committees can have different Threshold values

[1] Piotr Faliszewski, Edith Elkind, and Michael Wooldridge. 2009. Boolean combinations of weighted voting games. In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1 (AAMAS '09). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 185–192.

Questions to be answered?

- Which coalitions might be able to bring the goal about?
- How important is a particular individual with respect to the achievement of the goal?

Introduction

Goals of the Paper

- Formal Definition of **BWVG**
- Investigate Computational Properties of **BWVG**

Agenda

Preliminary Definitions

Propositional Logic

- Let $\Phi = \{p, q, \dots\}$ be a fixed non-empty vocabulary of Boolean variables
- Let $\mathcal{L}$ denote the set of formulas of propositional logic over Φ, constructed using conventional Boolean operators: $\wedge, \vee, \implies, \iff, \neg$

Propositional Logic

- Let $\Phi = \{p, q, \dots\}$ be a fixed non-empty vocabulary of Boolean variables
- Let $\mathcal{L}$ denote the set of formulas of propositional logic over Φ, constructed using conventional Boolean operators: $\wedge, \vee, \implies, \iff, \neg$
- If "$\vee$" and "$\wedge$" are the only operators appearing in formula φ, se say that φ is **monotone**

Propositional Logic

- Let $\Phi = \{p, q, \dots\}$ be a fixed non-empty vocabulary of Boolean variables
- Let $\mathcal{L}$ denote the set of formulas of propositional logic over Φ, constructed using conventional Boolean operators: $\wedge, \vee, \implies, \iff, \neg$
- If "$\vee$" and "$\wedge$" are the only operators appearing in formula φ, se say that φ is **monotone**
- If $\xi \subseteq \Phi$, we write $\xi \models \varphi$ mean that φ is true satisfied by valuation ξ

Preliminary Definitions

Simple Games

- A coalitional game is Simple if $v(C) \in \{0, 1\} \forall C \subseteq N$
- C wins if $v(C) = 1$ and C losses otherwise.
- A Simple Game is **monotone** if $v(C) = 1 \implies v(C') = 1$ for any $C \subseteq C'$.
- In this paper authors consider both *monotone* and *non-monotone* Simple Games.
- They assume games with finite numbers of players $|N| = n$, $N = \{1, \ldots, n\}$

Preliminary Definitions

Weighted Voting Games

- Given $N = \{1, \dots, n\}$ players
- A list of n weights $w = (w_1, \dots, w_n) \in \mathbb{R}^n$
- A threshold $T \in \mathbb{R}$
- When N is clear from the context $q = (T; w_1, \dots, w_n)$ to denote a *WVG* g
- $w(C)$ total weight of coalition C, $w(C) = \sum_{i \in C} w_i$
- Characteristic function given by $v(C) = 1$ if $w(C) \geq T$ and $v(C) = 0$ otherwise.
- If all Weights are non-negative the game is monotone.

Preliminary Definitions

Computational Complexity

- P, NP, $coNP$, Σ_2^p, Π_2^p

Preliminary Definitions

Computational Complexity

- P, NP, $coNP$, Σ_2^p, Π_2^p
- D^p: A Language $L \in D^p$ if $L = L_1 \cap L_2$, for some language $L_1 \in NP$ and $L_2 \in coNP$

Computational Complexity

- P, NP, $coNP$, Σ_2^p, Π_2^p
- D^p: A Language $L \in D^p$ if $L = L_1 \cap L_2$, for some language $L_1 \in NP$ and $L_2 \in coNP$
- D_2^p: A Language $L \in D_2^p$ if $L = L_1 \cap L_2$, for some language $L_1 \in \Sigma_2^p$ and $L_2 \in \Pi_2^p$

Agenda

Boolean Weighted Voting Games

Definition

A **BWVG** is a tuple $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$, where:

- $N = \{1, \ldots, n\}$ is a set of players;
- $\mathcal{G} = \{g^1, \ldots, g^n\}$ is a Set of **WVG** over N, where jth game, g^j, is given by a vector of weights $w^j = (w^j_1, \ldots, w^j_n)$ and a Threshold T^j. $\mathcal{G}$ is called the **component games** of G;
- $\Phi = \{p^1, \ldots, p^n\}$ Set of Propositional Variables, in which each p^j correspond with the **component** g^j;
- φ is a propositional formula over Φ.

Shorthand Definition

Example:

- $g^1 \wedge g^2 \equiv \langle N, \{g^1, g^2\}, \{p^1, p^2\}, p^1 \wedge p^2 \rangle$

Winning Coalition

We say that C is a *wins* G if:

$$\exists \xi_1 \subseteq \Phi_C \;:\; \forall \xi_2 \subseteq (\Phi \setminus \Phi_C) \;:\; \xi_1 \cup \xi_2 \models \varphi \tag{1}$$

Boolean Weighted Voting Games

Winning Coalition

We say that C is a *wins* G if:

$$\exists \xi_1 \subseteq \Phi_C \ : \ \forall \xi_2 \subseteq (\Phi \setminus \Phi_C) \ : \ \xi_1 \cup \xi_2 \models \varphi \tag{2}$$

Intuitively 3

A coalition C wins if it is able to fix variables under its control in such a way that the goal formula φ is guaranteed to be **True**.

Boolean Weighted Voting Games

Winning Coalition

We say that C is a *wins* G if:

$$\exists \xi_1 \subseteq \Phi_C \ : \ \forall \xi_2 \subseteq (\Phi \setminus \Phi_C) \ : \ \xi_1 \cup \xi_2 \models \varphi \tag{3}$$

Intuitively 3

A coalition C wins if it is able to fix variables under its control in such a way that the goal formula φ is guaranteed to be **True**.

Notes

It is allowed ***BWVG*** to contain *negative* weights

Agenda

Preliminaries

- Any Simple Game with n players can be represented as a *K-Vector Weighted Voting Game* for $k = O(2^n)$, and therefore as a **BWVG** with $O(2^n)$ **component games** $\mathcal{G}$.

Representational Complexity

Preliminaries

- Any Simple Game with n players can be represented as a *K-Vector Weighted Voting Game* for $k = O(2^n)$, and therefore as a **BWVG** with $O(2^n)$ **component games** $\mathcal{G}$.
- That worst-case unfortunately cannot be improved in **BWVG**

Representational Complexity

Proposition

The total number of Boolean weighted voting games with $|N| = n$ and $|\varphi| = s$ is most $2^{O(sn^2 \log(sn))}$

Proposition

The total number of Boolean weighted voting games with $|N| = n$ and $|\varphi| = s$ is most $2^{O(sn^2 \log(sn))}$

Proof.

- Any weighted voting game[2] can be represented using Integer weights whose absolute values do not exceed $2^{O(n \log n)}$

$\square$

[2] S. Muroga. Threshold Logic and its Applications. Wiley, 1971.

Proposition

The total number of Boolean weighted voting games with $|N| = n$ *and* $|\varphi| = s$ *is most* $2^{O(sn^2 \log(sn))}$

Proof.

- Any weighted voting game[3] can be represented using Integer weights whose absolute values do not exceed $2^{O(n \log n)}$
- *w.l.g.* we assumed that $|\mathcal{G}| = |\Phi|$ and $|\Phi| \leq |\varphi| = s$

□

[3] S. Muroga. Threshold Logic and its Applications. Wiley, 1971.

Proposition

The total number of Boolean weighted voting games with $|N| = n$ and $|\varphi| = s$ is most $2^{O(sn^2 \log(sn))}$

Proof.

- Any weighted voting game[4] can be represented using Integer weights whose absolute values do not exceed $2^{O(n \log n)}$
- *w.l.g.* we assumed that $|\mathcal{G}| = |\Phi|$ and $|\Phi| \leq |\varphi| = s$
- Given a **BWVG** G with n players and $|\varphi| = s$, we can find a equivalent representation using $O(sn^2 \log n)$ bits to represent all weights in ALL components, plus another $O(s \log s)$ bits to represent $\mathcal{G}, \Phi$ and φ.

$\square$

[4] S. Muroga. Threshold Logic and its Applications. Wiley, 1971.

Representational Complexity

Proposition

The total number of Boolean weighted voting games with $|N| = n$ *and* $|\varphi| = s$ *is most* $2^{O(sn^2 \log(sn))}$

Proof.

- Any weighted voting game[5] can be represented using Integer weights whose absolute values do not exceed $2^{O(n \log n)}$
- *w.l.g.* we assumed that $|\mathcal{G}| = |\Phi|$ and $|\Phi| \leq |\varphi| = s$
- Given a **BWVG** G with n players and $|\varphi| = s$, we can find a equivalent representation using $O(sn^2 \log n)$ bits to represent all weights in ALL components, plus another $O(s \log s)$ bits to represent $\mathcal{G}, \Phi$ and φ.
- Therefore, the total number of **distinct games** can be represented as **BWVG** with $|N| = n$ and $|\varphi| = s$ is $2^{O(sn^2 \log(sn))}$

□

[5] S. Muroga. Threshold Logic and its Applications. Wiley, 1971.

Representational Complexity

Linear Representation - Specific Case

- We are going to show that for **some specific** instance that captures **realistic voting scenarios** that can be improve with linear representation.

Representational Complexity

Linear Representation - Specific Case

- We are going to show that for **some specific** instance that captures **realistic voting scenarios** that can be improve with linear representation.

Theorem

*Consider a **BWVG** $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ where $\mathcal{G} = \{g^1, g^2\}, g^1 = (k; 1, 0, \ldots, 1, 0), g^2 = (k; 0, 1, \ldots, 0, 1), |N| = 2k$ and $\varphi = p^1 \vee p^2$. To represent G as a conjunction of m weighted voting games requires $m \geq k/2$ **component games** $\mathcal{G}$*

Proof.

- Poof by contradiction

$\square$

Representational Complexity

Proof.

- Poof by contradiction
- A coalition C to win in G has to contain either *even players* or *odd players*

$\square$

Representational Complexity

Proof.

- Poof by contradiction
- A coalition C to win in G has to contain either *even players* or *odd players*
- Any *maximal losing coalition* (MLC) in G is of the form $N \setminus \{2i, 2j-1\}$ where $i, j \in \{1, \ldots, k\}$, denote as $C_{i,j}$

$\square$

Proof.

- Poof by contradiction
- A coalition C to win in G has to contain either *even players* or *odd players*
- Any *maximal losing coalition* (MLC) in G is of the form $N \setminus \{2i, 2j-1\}$ where $i, j \in \{1, \ldots, k\}$, denote as $C_{i,j}$
- There are exactly k^2 MLC

□

Representational Complexity

Proof.

- Poof by contradiction
- A coalition C to win in G has to contain either *even players* or *odd players*
- Any *maximal losing coalition* (MLC) in G is of the form $N \setminus \{2i, 2j-1\}$ where $i, j \in \{1, \dots, k\}$, denote as $C_{i,j}$
- There are exactly k^2 MLC
- 2 MLC $C_{i,j}$ and $C_{i',j'}$ clashes if $i = i'$ or $j = j'$, if $C_{i,j} \cup C_{i',j'} \neq N$

$\square$

Proof.

- Poof by contradiction
- A coalition C to win in G has to contain either *even players* or *odd players*
- Any *maximal losing coalition* (MLC) in G is of the form $N \setminus \{2i, 2j-1\}$ where $i, j \in \{1, \ldots, k\}$, denote as $C_{i,j}$
- There are exactly k^2 MLC
- 2 MLC $C_{i,j}$ and $C_{i',j'}$ clashes if $i = i'$ or $j = j'$, if $C_{i,j} \cup C_{i',j'} \neq N$
- **Suppose** that G can be represented as $\langle N, \{h^1, \ldots, h^m\}, \{q^1, \ldots, q^m\}, q^1 \wedge \cdots \wedge q^m \rangle$ with $m < k/2$

$\square$

Representational Complexity

Proof.

- Poof by contradiction
- A coalition C to win in G has to contain either *even players* or *odd players*
- Any *maximal losing coalition* (MLC) in G is of the form $N \setminus \{2i, 2j-1\}$ where $i, j \in \{1, \ldots, k\}$, denote as $C_{i,j}$
- There are exactly k^2 MLC
- 2 MLC $C_{i,j}$ and $C_{i',j'}$ clashes if $i = i'$ or $j = j'$, if $C_{i,j} \cup C_{i',j'} \neq N$
- **Suppose** that G can be represented as $\langle N, \{h^1, \ldots, h^m\}, \{q^1, \ldots, q^m\}, q^1 \wedge \cdots \wedge q^m \rangle$ with $m < k/2$
- Each component has to lose in at least one game $h^1, \ldots, h^m$. By **pigeonhole principle**, there must be at least 1 component game (w.l.g.) that is lost by at least $2k$ distinct MLC.

$\square$

Proof Cont.

- Fix an arbitrary MLC $C_{i,j}$ that loses in h^1

□

Representational Complexity

Proof Cont.

- Fix an arbitrary MLC $C_{i,j}$ that loses in h^1
- Among $2k$ MLCs that loses in h^1 there can be at most $k-1$ MLCs of the form $C_{i,j'}, j' \neq j$ and $C_{i',j}, i' \neq i$

$\square$

Proof Cont.

- Fix an arbitrary MLC $C_{i,j}$ that loses in h^1
- Among $2k$ MLCs that loses in h^1 there can be at most $k-1$ MLCs of the form $C_{i,j'}, j' \neq j$ and $C_{i',j}, i' \neq i$
- There must be a $C_{x,y}$ that loses in h^1 and don't clashes with $C_{i,j}$.

$\square$

Proof Cont.

- Fix an arbitrary MLC $C_{i,j}$ that loses in h^1
- Among $2k$ MLCs that loses in h^1 there can be at most $k-1$ MLCs of the form $C_{i,j'}, j' \neq j$ and $C_{i',j}, i' \neq i$
- There must be a $C_{x,y}$ that loses in h^1 and don't clashes with $C_{i,j}$.
- Let $h^1 = (T; w_1, \ldots, w_n)$, we have

$$w(N) - w_{2i} - w_{2j-1} < T; w(N) - w_{2x} - w_{2y-1} < T \tag{4}$$

$\square$

Proof Cont.

- Fix an arbitrary MLC $C_{i,j}$ that loses in h^1
- Among $2k$ MLCs that loses in h^1 there can be at most $k-1$ MLCs of the form $C_{i,j'}, j' \neq j$ and $C_{i',j}, i' \neq i$
- There must be a $C_{x,y}$ that loses in h^1 and don't clashes with $C_{i,j}$.
- Let $h^1 = (T; w_1, \ldots, w_n)$, we have

$$w(N) - w_{2i} - w_{2j-1} < T; w(N) - w_{2x} - w_{2y-1} < T \tag{5}$$

- Also, $C_{i,j} \setminus \{2y-1\} \cup \{2i\}$ and $C_{x,y} \setminus \{2y-1\} \cup \{2i\}$ are wining in G and hence in h^1

$$w(N) - w_{2j-1} - w_{2y-1} \geq T; w(N) - w_{2i} - w_{2x} \geq T \tag{6}$$

Equation 5 and 6 give a contradiction Therefore $m \geq k/2$. □

Agenda

Decision Problems in BWVG

Winning Coalitions

Given a game $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ *and a coalition* $C \subseteq N$*, deciding whether* C *wins in* G *is* Σ_2^p*-complete. This results holds even if there are 2 players and the weights of all players in all components are in* $\{0, 1\}$*. However, the problem is in* P *if the underlying formula is monotone.*

Decision Problems in BWVG

Winning Coalitions

Given a game $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ and a coalition $C \subseteq N$, deciding whether C wins in G is Σ_2^p-complete. This results holds even if there are 2 players and the weights of all players in all components are in $\{0, 1\}$. However, the problem is in P if the underlying formula is monotone.

Proof Sketch.

- By definition of Winning coalition of **BWVG** is easy to see that is in Σ_2^p for the general case

$\square$

Decision Problems in BWVG

Winning Coalitions

Given a game $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ *and a coalition* $C \subseteq N$*, deciding whether* C *wins in* G *is* Σ_2^p*-complete. This results holds even if there are 2 players and the weights of all players in all components are in* $\{0, 1\}$*. However, the problem is in* P *if the underlying formula is monotone.*

Proof Sketch.

- By definition of Winning coalition of **BWVG** is easy to see that is in Σ_2^p for the general case
- In the case of monotonicity of propositional formula testing whether a Coalition C is winning we need to set all all the controlled variables by C in **True**, while All others in $\perp$.

$\square$

Decision Problems in BWVG

Winning Coalitions

Given a game $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ and a coalition $C \subseteq N$, deciding whether C wins in G is Σ_2^p-complete. This results holds even if there are 2 players and the weights of all players in all components are in $\{0, 1\}$. However, the problem is in P if the underlying formula is monotone.

Proof Sketch.

- By definition of Winning coalition of **BWVG** is easy to see that is in Σ_2^p for the general case
- In the case of monotonicity of propositional formula testing whether a Coalition C is winning we need to set all all the controlled variables by C in **True**, while All others in $\perp$.
- With formulas with few variables we can enumerate all possible truth assignments.

□

Decision Problems in BWVG

Winning Coalitions

Given a game $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ *and a coalition* $C \subseteq N$*, deciding whether* C *wins in* G *is* Σ_2^p*-complete. This results holds even if there are 2 players and the weights of all players in all components are in* $\{0, 1\}$*. However, the problem is in* P *if the underlying formula is monotone.*

Proof Sketch.

- By definition of Winning coalition of **BWVG** is easy to see that is in Σ_2^p for the general case
- In the case of monotonicity of propositional formula testing whether a Coalition C is winning we need to set all all the controlled variables by C in **True**, while All others in $\perp$.
- With formulas with few variables we can enumerate all possible truth assignments.
- For the case of unrestricted formulas we do a reduction from $QSAT_2$

$\square$

Swing Player: Definition

i is a **swing player** for C in game G if C loses in G but $C \cup \{i\}$ wins in G. The problem of deciding if i is Swing Player or not, is easy if φ is monotone or its size is bounded by a constant, but in general is Computationally hard.

Swing Player: Definition

i is a **swing player** for C in game G if C loses in G but $C \cup \{i\}$ wins in G. The problem of deciding if i is Swing Player or not, is easy if φ is monotone or its size is bounded by a constant, but in general is Computationally hard.

Swing Player

SWINGPLAYER is D_2^p-complete. This holds even for 3 players and all components are of the form $\{0, 1\}$. However, the problem is in P if the underlying formula is monotone.

Proof Sketch.

- The case of monotone formulas follows directly from previous theorem of Winning Coalitions.

□

Decision Problems in BWVG

Proof Sketch.

- The case of monotone formulas follows directly from previous theorem of Winning Coalitions.
- We must exhibit 2 languages L_1 and L_2, such that $L_1 \in \Sigma_2^p$, $L_2 \in \Pi_2^p$ and $SWINGPLAYER = L_1 \cap L_2$.

$$L_1 = \{\langle G, C, i\rangle : C \cup \{i\} \text{ wins in } G\} \tag{7a}$$

$$L_2 = \{\langle G, C, i\rangle : C \text{ does not win in } G\} \tag{7b}$$

□

Proof Sketch.

- The case of monotone formulas follows directly from previous theorem of Winning Coalitions.
- We must exhibit 2 languages L_1 and L_2, such that $L_1 \in \Sigma_2^p$, $L_2 \in \Pi_2^p$ and $SWINGPLAYER = L_1 \cap L_2$.

$$L_1 = \{\langle G, C, i\rangle : C \cup \{i\} \text{ wins in } G\} \tag{8a}$$

$$L_2 = \{\langle G, C, i\rangle : C \text{ does not win in } G\} \tag{8b}$$

- Clearly $L_1 \in \Sigma_2^p$ and $L_2 \in \Pi_2^p$

□

Proof Sketch.

- The case of monotone formulas follows directly from previous theorem of Winning Coalitions.
- We must exhibit 2 languages L_1 and L_2, such that $L_1 \in \Sigma_2^p$, $L_2 \in \Pi_2^p$ and $SWINGPLAYER = L_1 \cap L_2$.

$$L_1 = \{\langle G, C, i \rangle : C \cup \{i\} \text{ wins in } G\} \tag{9a}$$

$$L_2 = \{\langle G, C, i \rangle : C \text{ does not win in } G\} \tag{9b}$$

- Clearly $L_1 \in \Sigma_2^p$ and $L_2 \in \Pi_2^p$
- By definition $SWINGPLAYER = L_1 \cap L_2$

$\square$

Proof Sketch.

- The case of monotone formulas follows directly from previous theorem of Winning Coalitions.
- We must exhibit 2 languages L_1 and L_2, such that $L_1 \in \Sigma_2^p$, $L_2 \in \Pi_2^p$ and $SWINGPLAYER = L_1 \cap L_2$.

$$L_1 = \{\langle G, C, i \rangle : C \cup \{i\} \text{ wins in } G\} \tag{10a}$$
$$L_2 = \{\langle G, C, i \rangle : C \text{ does not win in } G\} \tag{10b}$$

- Clearly $L_1 \in \Sigma_2^p$ and $L_2 \in \Pi_2^p$
- By definition $SWINGPLAYER = L_1 \cap L_2$
- To show D_2^p-hardness a reduction can be provided from D_2^p-complete problem $SAT_2^{\Sigma} - UNSAT_2^{\Sigma}$, which is a generalization of $SAT - UNSAT$ problem.

□

Decision Problems in BWVG

Dummy Player: Definition

i is a **dummy player** for C in game G if $v(C) = v(C \cup \{i\})$ for all $C \subseteq N \setminus \{i\}$.

Dummy Player

DUMMYPLAYER is coNP-hard even if all weights in all component games are in $\{0, 1\}$*, and* G *is in an* m*-vector weighed voting game.*

Decision Problems in BWVG

Proof Sketch.

- The proof can be done following a reduction from the clasic *NP*-complete problem **X3C (Exact Cover by 3-Sets)**
- An instance $\mathcal{E}$ of this problem is given

$\square$

Proof Sketch.

- The proof can be done following a reduction from the clasic *NP*-complete problem **X3C (Exact Cover by 3-Sets)**
- An instance $\mathcal{E}$ of this problem is given
- It is an "yes"-instance if $\mathcal{E}$ can be covered by K, "no"-otherwise

$\square$

Decision Problems in BWVG

Proof Sketch.

- The proof can be done following a reduction from the clasic *NP*-complete problem **X3C (Exact Cover by 3-Sets)**
- An instance $\mathcal{E}$ of this problem is given
- It is an "yes"-instance if $\mathcal{E}$ can be covered by K, "no"-otherwise
- Basically given this $\mathcal{E}$ a **BWVG** is constructed, where the games $\mathcal{G}$ are of size $3K + 1$.

$\square$

Decision Problems in BWVG

Proof Sketch.

- The proof can be done following a reduction from the clasic *NP*-complete problem **X3C (Exact Cover by 3-Sets)**
- An instance $\mathcal{E}$ of this problem is given
- It is an "yes"-instance if $\mathcal{E}$ can be covered by K, "no"-otherwise
- Basically given this $\mathcal{E}$ a **BWVG** is constructed, where the games $\mathcal{G}$ are of size $3K + 1$.
- A Coalition C wins the first $3K$ games if and only if corresponds to a valid cover of $\mathcal{E}$

$\square$

Decision Problems in BWVG

Proof Sketch.

- The proof can be done following a reduction from the clasic *NP*-complete problem **X3C (Exact Cover by 3-Sets)**
- An instance $\mathcal{E}$ of this problem is given
- It is an "yes"-instance if $\mathcal{E}$ can be covered by K, "no"-otherwise
- Basically given this $\mathcal{E}$ a **BWVG** is constructed, where the games $\mathcal{G}$ are of size $3K + 1$.
- A Coalition C wins the first $3K$ games if and only if corresponds to a valid cover of $\mathcal{E}$
- Analyzing the "no"-instance and the reminding players $+1$ establish the Dummy player or not.

$\square$

Decision Problems in BWVG

Proof Sketch.

- The proof can be done following a reduction from the clasic *NP*-complete problem **X3C (Exact Cover by 3-Sets)**
- An instance $\mathcal{E}$ of this problem is given
- It is an "yes"-instance if $\mathcal{E}$ can be covered by K, "no"-otherwise
- Basically given this $\mathcal{E}$ a **BWVG** is constructed, where the games $\mathcal{G}$ are of size $3K + 1$.
- A Coalition C wins the first $3K$ games if and only if corresponds to a valid cover of $\mathcal{E}$
- Analyzing the "no"-instance and the reminding players $+1$ establish the Dummy player or not.
- Therefore, a "no"-instance of $X3C$ is an "yes"-instance of DUMMYPLAYER.

$\square$

Agenda

Shapley Value

Shapley Value in BWVG

- In **WVG** it is known that computing Shapley Value is hard ($\#P$-complete).

Shapley Value

Shapley Value in BWVG

- In **WVG** it is known that computing Shapley Value is hard ($\#P$-complete).
- This implies that the problem is as least as hard to **BWVG**

Shapley Value

Shapley Value in BWVG

- In **WVG** it is known that computing Shapley Value is hard ($\#P$-complete).
- This implies that the problem is as least as hard to **BWVG**
- However there is a *poly-time* algorithm for computing Shapley Value in **WVG** with unary-encoded weights.

Shapley Value in BWVG

- In **WVG** it is known that computing Shapley Value is hard ($\#P$-complete).
- This implies that the problem is as least as hard to **BWVG**
- However there is a *poly-time* algorithm for computing Shapley Value in **WVG** with unary-encoded weights.
- But this is not true for **BWVG**

Shapley Value

Shapley Value

Computing a player's Shapley value in a ***BWVG*** *is* $\#P$*-hard even if the game in question is a* ***VWVG*** *and all weights in all component games are in* $\{0, 1\}$.

Shapley Value

Shapley Value

*Computing a player's Shapley value in a **BWVG** is $\#P$-hard even if the game in question is a **VWVG** and all weights in all component games are in $\{0, 1\}$.*

Proof.

- For the probe it is use a reduction of $X3C$ (Exact Cover by 3-Sets), where an instance of this problem is giving and a **BWVG** is constructed based on this.
- Given that there is a q which is a swing player for exactly N_k combinations, where N_k is the number of exact covers of $\mathcal{E}$, and the size of each such coalition is exactly K.
- Hence the Shapley Value for the q player is exactly $N_k \frac{K!(\ell+1-K)!}{(\ell+1)!}$
- N_K can be compute given sh_q^G, ℓ, and K
- As computing N_K is $\#P$-complete, it follows the statement.

$\square$

Shapley Value

Conclusion

- Shapley value requires to decide whether a given players is a SWINGPLAYER or not.
- Since SWINGPLAYER is in D_2^p-complete as we have seen, this hardness cannot be improved as it seems.

Shapley Value

Conclusion

- Shapley value requires to decide whether a given players is a SWINGPLAYER or not.
- Since SWINGPLAYER is in D_2^p-complete as we have seen, this hardness cannot be improved as it seems.

Poly-time

- Shapley value can be still computed in *poly-time* if both the weights are given in unary and the number of component games is bounded by a constant.

Shapley Value

Theorem: Shapley Value in Poly-Time

Given a BWVG $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ *and a player* $p \in N$*, Shapley value of* p *can be computed in time* $O((n^2 + s)(4nW)^m)$*, where* $|\Phi| = m, |\varphi| = s, |W| = max_{i,j}|w_i^j|$

Shapley Value

Theorem: Shapley Value in Poly-Time

Given a BWVG $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ *and a player* $p \in N$, *Shapley value of* p *can be computed in time* $O((n^2 + s)(4nW)^m)$, *where* $|\Phi| = m, |\varphi| = s, |W| = max_{i,j}|w_i^j|$

Proof.

- Proof using Dynamic Programming.
- Given an Integer vector $z = (z^1, \ldots, z^m) \in [-nW, nW]^m$, any $k = 1, \ldots, n-1$, and $t = 1, \ldots, k$

□

Shapley Value

Theorem: Shapley Value in Poly-Time

Given a BWVG $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ *and a player* $p \in N$*, Shapley value of* p *can be computed in time* $O((n^2 + s)(4nW)^m)$*, where*
$|\Phi| = m, |\varphi| = s, |W| = max_{i,j}|w_i^j|$

Proof.

- Proof using Dynamic Programming.
- Given an Integer vector $z = (z^1, \ldots, z^m) \in [-nW, nW]^m$, any $k = 1, \ldots, n-1$, and $t = 1, \ldots, k$
- Let $N(z, t, k)$ be the number of coalitions $C \subseteq N$, such that $w^j(C) = z^j$ for $j = 1, \ldots, m$, $|C| = t$, and $C \subseteq \{1, \ldots, k\}$.

$\square$

Shapley Value

Theorem: Shapley Value in Poly-Time

Given a BWVG $G = \langle N, \mathcal{G}, \Phi, \varphi \rangle$ *and a player* $p \in N$*, Shapley value of* p *can be computed in time* $O((n^2 + s)(4nW)^m)$*, where* $|\Phi| = m, |\varphi| = s, |W| = max_{i,j}|w_i^j|$

Proof.

- Proof using Dynamic Programming.
- Given an Integer vector $z = (z^1, \dots, z^m) \in [-nW, nW]^m$, any $k = 1, \dots, n-1$, and $t = 1, \dots, k$
- Let $N(z, t, k)$ be the number of coalitions $C \subseteq N$, such that $w^j(C) = z^j$ for $j = 1, \dots, m$, $|C| = t$, and $C \subseteq \{1, \dots, k\}$.
- For $k = 1$, $N(z, t, 1) = 1$ if $t = 1$ and $w_1^j = z^j$ and $N(z, t, 1) = 0$ otherwise.

$$N(z, t, k+1) = N(z, t, k) + N(z_{k+1}, t-1, k) \qquad (11)$$

□

Proof Cont. Poly-Time.

- First summand Coalitions without $k+1$ player, Second summand with $k+1$ player

$\square$

Proof Cont. Poly-Time.

- First summand Coalitions without $k+1$ player, Second summand with $k+1$ player
- So all $N(z,t,k)$ can be compute in $O(n^2(2nW)^m)$

$\square$

Proof Cont. Poly-Time.

- First summand Coalitions without $k+1$ player, Second summand with $k+1$ player
- So all $N(z,t,k)$ can be compute in $O(n^2(2nW)^m)$
- Let $I(z,t) = v(C \cup \{n\}) - v(C)$
- Computing the value of φ under a truth assignment can be done in $O(s)$, and for a fixed vector z in $O(s2^m)$. Hence all $I(z,t)$ requires $O(s2^m(2nW)^m)$

$\square$

Proof Cont. Poly-Time.

- First summand Coalitions without $k+1$ player, Second summand with $k+1$ player
- So all $N(z,t,k)$ can be compute in $O(n^2(2nW)^m)$
- Let $I(z,t) = v(C \cup \{n\}) - v(C)$
- Computing the value of φ under a truth assignment can be done in $O(s)$, and for a fixed vector z in $O(s2^m)$. Hence all $I(z,t)$ requires $O(s2^m(2nW)^m)$
- Shapley value can be computed as follows:

$\square$

Shapley Value

Proof Cont. Poly-Time.

- First summand Coalitions without $k+1$ player, Second summand with $k+1$ player
- So all $N(z,t,k)$ can be compute in $O(n^2(2nW)^m)$
- Let $I(z,t) = v(C \cup \{n\}) - v(C)$
- Computing the value of φ under a truth assignment can be done in $O(s)$, and for a fixed vector z in $O(s2^m)$. Hence all $I(z,t)$ requires $O(s2^m(2nW)^m)$
- Shapley value can be computed as follows:

$$sh_n = \frac{1}{n!} \sum_{z \in [-nW, nW]^m} \sum_{t=1}^{n-1} N(z,t,n-1) I(z,t) t!(n-1-t)! \tag{12}$$

- Therefore, overall running time: $O(n^2(2nW)^m) + s2^m(2nW)^m) = O((n^2+s)(4nW)^m)$

□

Agenda

The Core and BWVG

- Problem **InCore** we are given a BWVG G and a payoff vector x and we are asked if x belongs to G's core.
- Problem **CoreNonEmpty** we are given a BWVG G and we ask if its core is nonempty
- Problem **Veto** we are given a BWVG G and a player i and we ask if i is a veto player in G

The Core

InCore, CoreNonEmpty and Veto

InCore, CoreNonEmpty and Veto *are Π_2^p-complete even if $|N| = 2$ and all weights in all components games are either 0 or 1. However for non-negative weights these problems are in P if the underlying formulas are monotone.*

Proof

Authors Do Not provide any proof due to space restrictions.

Agenda

Conclusions

Conclusions

- **BWVG** are more expressive than **VWVG**, in particular when formulas are monotone

Conclusions

Conclusions

- **BWVG** are more expressive than **VWVG**, in particular when formulas are monotone
- Although expressiveness gain, **BWVG** are worst in terms of Computational Complexity

Conclusions

Conclusions

- **BWVG** are more expressive than **VWVG**, in particular when formulas are monotone
- Although expressiveness gain, **BWVG** are worst in terms of Computational Complexity
- Unrestricted **BWVG** leads to increase Complexity

Conclusions

Conclusions

- **BWVG** are more expressive than **VWVG**, in particular when formulas are monotone
- Although expressiveness gain, **BWVG** are worst in terms of Computational Complexity
- Unrestricted **BWVG** leads to increase Complexity
- As we have seen there are trade-off to deal with this increase of Complexity and gain in expressiveness

Thank you!!