Vertex Cover Games and Weighted Voting Games

Erasmus Traineeship Report
Georgia Maria Fotopoulou
Supervisor: Maria Serna

June 26, 2023

1 Introduction

This is the report regarding the research on the subjects of the Vertex Cover Game (VCG) and Weighted Voting Games (WVG). The matters mentioned concerning the above subjects will be the basic explanation of the WVGs and coalitional games, the use of power indices in such games that influence the player's power and how much a player can change their power and manipulate the game using these indices as a measure. Finally, we will explain the Vertex Cover Games and there will be discussed some ways that they can be manipulated.

2 Weighted Voting Games

In this section we are going to present the cooperative games in the field of Game Theory in which players, by forming coalitions, can compete or cooperate with each other to get their desired outcome. That outcome is judged from certain power indices that we compute in order to see the increase, decrease or maintenance of a player's pay-off.

2.1 Coalitional Games

Coalition: An alliance for combined action (a temporary alliance of political parties forming a goverment)
Coalitional Game: A coalitional game G = (N,v) is given by a set of players N = {1,...,n}, and a characteristic function $v : 2^N \rightarrow R$, which maps any subset, or coalition, of players to a real value. This value is the total utility these players can guarantee to themselves when working together.

2.2 Simple Games vs Weighted Voting Games

Definition. A game v is a **simple game** if it satisfies the relation $v(S) \in \{0, 1\} \forall S \subseteq N$ coalitions S such that $v(S) = 1(0)$ are called winning (losing) coalitions. A simple game (N, v) is given by a set N of n players and a partition of 2^N into a set L of losing coalitions L with value v(L) = 0 that is closed undertaking subsets and a set W of winning coalitions W with v(W) = 1. Simple games with

$$a = min_{p\geq 0} max_{W\in W, L\in L} \frac{p(L)}{p(W)} < 1$$

are known as **weighted voting games**.

2.3 Weighted Voting Games

Weighted voting is a classic model of cooperation among agents in decision-making domains. In such games, each player has a weight, and a coalition of players wins the game if its total weight meets or exceeds a given quota. The agents' weights may correspond to the number of resources (time, money, or battery power) that they contribute, and the quota may indicate the amount of resources needed to complete a given task. Alternatively, the weight may be an indicator of an agent's experience or seniority, and a voting procedure may be designed to take into account these characteristics. Having a larger weight allows the player to affect the outcome more easily. Weighted voting games provide a model of decision-making in many political and legislative and have also been investigated in the context of multi agent systems.

Definition. A **weighted voting game** G is a simple game that is described by a vector of players' weights $w = (w_1, ..., w_n) \in (R^+)^n$ and a quota q $\in R^+$. We write $G = [q; w_1, ..., w_n]$, or G = [q; w]. Let v be a game on a set of players N = {1,2,. . . ,n}. For any $S \subseteq N$ we have v(S) = 1 if

$$\sum_{i \in S} w_i \geq q$$

and v(S) = 0 otherwise.

- w(S): the total weight of a coalition S: $w(S) = \sum_{i \in S} w_i$
- $w(N) \geq q$: the grand winning coalition
- q = w(N): then any player $i \in N$ is a veto player

Types of players:
- *Veto player: in a simple game player i is called a veto player if* $i \notin S \Rightarrow v(S) = 0$
- *Dummy player: player i is a dummy player if* $i \notin S \Rightarrow v(S \cup \{i\}) = v(S)$

2.4 Shapley-Shubik and Banzhaf power indices

Power indices methodology is widely used to measure an a priori voting power of members of a committee. With the use of a power index we measure a player's influence in a weighted voting game. A player's power in such games is usually not directly proportional to his weight, and is measured by a power index and in these cases we have studied the Shapley- Shubik index and the Banzhaf index. Power indices measure the player's power and can be used to determine their payoffs.

2.4.1 Shapley-Shubik Index (SSI)

It is a classic method of distributing the gains of the grand coalition in general coalitional games. We the SS index we take into consideration the order in which the players join the coalition, therefore we consider the **sequential coalitions**. The SS power index determines how likely a player is to be **pivotal**.

Pivotal player: is the player in a sequential coalition that changes a losing coalition to a winning one. There can only be one pivotal player in a sequential coalition.

Definition 1. Let Π_n be the set of all possible permutations (orderings) of n players. Each $\pi \in \Pi_n$ is a one-to-one mapping from {1,...,n} to {1,...,n}. Denote by $S_\pi(i)$ the set of predecessors of player i in π, i.e., $S_\pi(i) = \{j|\pi(j) < \pi(i)\}$. The Shapley value of the i-th player in a game G = (N,v) is denoted by $\phi_i(G)$ and is given by the following expression:

$$\varphi_i(G) = \frac{1}{n!} \sum_{\pi \in \Pi_n} [v(S_\pi(i) \cup \{i\}) - v(S_\pi(i))]$$

In such games the value of each coalition is either 0 or 1, so the formula above counts the fraction of all orderings of the players in which player i is critical for the coalition formed by his predecessors. The SSI thus reflects the assumption that when forming a coalition, any ordering of the players entering the coalition has an equal probability of happening and expresses the probability that player i is critical.

Definition 2. Let $n = |N|$ be the number of players in G and $i \in N$.
The Shapley-Shubik power index of player i in G is defined by

$$\phi(G, i) = \frac{\sum_{S \subseteq N \setminus \{i\}} |S|!(n-1-|S|)!(v(S \cup \{i\}) - v(S))}{n!}$$

We assume that there is some communication structure among the players and define the corresponding notion of graph-restricted weighted voting games.

2.4.2 Banzhaf Index (BI)

The Banzhaf index is a power index defined by the probability of changing an outcome of a vote where voting rights are not necessarily equally divided among the voters. Here we don't take into consideration the ordering of the players. The BI determines how likely a voter is to be **critical**.

Critical player: if this player changes their vote from 'yes' to 'no' the measure can fail.
The The BI computes the probability that the player is critical under the assumption that all coalitions of the players are equally likely. Formally, given a game G = (N,v), for each $i \in N$ we denote by $\eta_i(G)$ the number of coalitions for which i is critical in a game G.
Definition 3. The Banzhaf index of a player i in a weighted voting game G = (N,v) is

$$\beta_i(G) = \frac{\eta_i(G)}{\sum_{i \in N} \eta_j(G)}$$

Definition 4. Let $n = |N|$ be the number of players in G and $i \in N$.
The Penrose-Banzhaf power index of player i in G is defined by

$$\beta(G, i) = \frac{\sum_{S \subseteq N \setminus \{i\}} (\nu(S \cup \{i\}) - \nu(S))}{2^{n-1}}$$

2.5 False name manipulations in Weighted Voting Games

A false name manipulation is the weight-splitting technique of a player among two or more identities. Weight-splitting can be advantageous, disadvantageous or even neutral for the player. After splitting, naturally the payoff would be distributed among the players according to their power in the new game. The total payoff of the new split identities would be equal to the original payoff before the split, but as the manipulation technique can also be disadvantageous this is not always the case according to the SS and BI indices.

2.6 Weight Splitting

Players can divide their weights arbitrarily among new identities and thus the payoff is distributed among the other players according to their power in the final game. The total payoff of the new identities should be equal to the payoff of the original player before the split. Weight splitting can be **advantageous, disadvantageous** and even **neutral**. The results are judged according to the change of the SS and BI indices which declare the increase or decrease of the manipulator's power.

2.6.1 Bounds of manipulation in Splitting

A player can increase but also decrease their total payoff by weight-splitting. For that reason, there are upper and lower bounds on how much the player's payoff can be changed. Let's say we have a game $G = [q; w_1, \dots, w_n]$ where the manipulator is player n and he splits into two identities n' and n" resulting in a new game G'.

Shapley-Shubik: $\phi'_n(G') + \phi''_n(G') \leq \frac{2n}{n+1}\phi_n(G)$. The manipulator can't gain more than $\frac{2n}{n+1} < 2$. The bound is tight.

Banzhaf: $\beta'_n(G') + \beta''_n(G') \leq 2\beta_n(G)$. The bound is asymptotically tight.

Bound of loss of the manipulator: For any game $G = [q; w_1, ..., w_n]$ and any split of n into n' and n", we have $\phi_{n'}(G') + \phi_{n''}(G') \geq \frac{n+1}{2}\phi_n(G)$ (tight). The player cannot lose more than $\frac{n+1}{2}$.

2.6.2 Complexity of Beneficial Splitting

From the computational perspective, we have seen that computational complexity acts as a barrier to manipulative behavior. Below we display the hardness for the SS, BI indices for the increase of the final payoff.
Types of beneficial splits:
Beneficial SS split
Instance: (G, l) where $G = [q; w_1, ..., w_n]$ is a weighted voting game and $l \in 1, ..., n$.

Question: Is there a way for player l to split his weight w between sub-players $l_1, ..., l_m$ so that in the new game G' it holds that $\sum_{j=1}^{m} \phi_{l_j}(G') > \phi(G)$?
Beneficial-SS-Split is **NP-hard**, and remains NP-hard even if the player can only split into two players with equal weights.
Beneficial BI split
Instance: (G, l) where $G = [q; w_1, ..., w_n]$ is a weighted voting game and $l \in 1, ..., n$.
Question: Is there a way for player l to split his weight w between sub-players $l_1, ..., l_m$ so that in the new game G' it holds that $\sum_{j=1}^{m} \beta_{l_j}(G') > \beta(G)$?
Beneficial-SS-Split is **NP-hard**, and remains NP-hard even if the player can only split into two players with equal weights.

2.7 Merging and Annexation

The act of players merging into a single entity. Annexation: One player takes the voting weight of other players. Merging (voluntary): Players merge to become a bloc so that their total payoff exceeds their original individual payoffs.

Beneficial SS Merge: NP-Hard

Beneficial BI Merge: NP-Hard

Beneficial SS Annexation: *never* disadvantageous
For any weighted voting game G with the set of players N, any $i \in N$, and any $S \subseteq N \backslash \{i\}$ we have $\phi_i(G) \leq \phi_{\&(S \cup \{i\})}(G\&(S \cup \{i\}))$.

Beneficial BI Annexation: NP-Hard

2.8 Graph Restricted Weighted Voting Games

A graph-restricted weighted voting game is a weighted voting game $G = (w_1, ..., w_n; q)$ together with a graph G = (N,E), where

$$v(S) = \begin{cases} 1 & \text{if S has a connected part S' with } w_{S'} \geq q \\ 0 & \text{otherwise} \end{cases}$$

Graph-restricted weighted voting games generalize weighted voting games, which are the special cases with a complete graph as their communication structures. In this situation, whether a coalition wins or loses is determined only by its total weight. However, if we limit the possibilities in communication among players, a coalition's weight alone is not enough. Before we define appropriate power indices in graph-restricted weighted voting games, let us present a few useful notions referring to coalitions in the sense of graph restrictions.

2.8.1 Manipulation in Communication Structures of Graph-Restricted Weighted Voting Games

By manipulating a communication-structure we focus on the change of the BI and SSI as we are interested in their values, how they can change in different situations, and in the computational complexity of problems defined on them.

2.8.2 Adding Edges to a Communication Graph

On graph-restricted weighted voting games we consider the impact of adding new edges on the Banzhaf and Shapley-Shubik indices. By doing that, we allow players to communicate with each other.
To increase a power index:
Given: A graph-restricted WVG (Γ, G), where Γ is the game and G is the communication stracture, with players $N = \{1, ..., n\}$, G = (N,E), $|E| < \binom{n}{2}$, a player $p \in N$ and a non-negative integer k.
Question: Is it sufficient to add k or fewer edges $E' \subseteq E^c = \{(x, y) \in N \times N | (x, y) \notin E\}$ to G to

obtain a new game $(\Gamma, G_{\cup E'})$ for which it holds that the power index $((\Gamma, G_{\cup E'}), p) >$ from the power index $((\Gamma, G), p)$?

We also define similar decision problems for decreasing and maintaining the player's power index by replacing " $>$ " with " $<$ " and " $=$ ".

Theorem 1. Let (Γ, G) be a graph-restricted weighted voting game and let $E', |E'| = m$, be a set of edges that are to be added to G, between a player i and other players, creating a new game $(\Gamma, G_{\cup E'})$. For player i and for $\chi \in \{\beta, \phi\}$, let $diff_\chi(\Gamma, G, G \cup E', i) = \chi((\Gamma, G), i) - \chi((\Gamma, G \cup E'), i)$. The old and the new Penrose-Banzhaf index and the old and the new Shapley-Shubik index of player i can differ as follows:

$$-1 + 2^{-(|N(i)|+m)} \leq diff_\beta(\Gamma, G, G_{\cup E'}, i) \leq (1 - 2^{-m}\beta((\Gamma, G), i)$$
$$-1 + \frac{1}{|N(i)+m+1|} \leq diff_\varphi(\Gamma, G, G_{\cup E'}, i) \leq (1 - \frac{(n-m)!}{n!})\varphi((\Gamma, G), i)$$

Theorem 2. Control by textbfadding edges between players to *decrease* a distinguished player's Penrose-Banzhaf index in a graph-restricted weighted voting game is **NP-hard**, to *maintain* this index is **coNP-hard**, and to *increase* it is **DP-hard**.

Theorem 3. Control by **adding edges** between players to *decrease or to increase* a distinguished player's Shapley-Shubik index in a graph-restricted weighted voting game is **NP-hard** and to *maintain* the index is **coNP-hard**.

2.8.3 Deleting Edges from a Communication Graph

To increase a power index:
Given: A graph-restricted WVG (Γ, G), where Γ is the game and G is the communication structure, with players $N = \{1, ..., n\}$, G = (N,E), a player $p \in N$ and a positive integer $k \leq |E|$.
Question: Can at most k edges $E' \subseteq E$ be deleted from G such that for the new game $(\Gamma, G_{\setminus E'})$ it holds that the power index $((\Gamma, G_{\setminus E'}), p) >$ from the power index $((\Gamma, G), p)$?

Theorem 4. Let (Γ, G) be a graph-restricted weighted voting game and let $E', |E'| = m$, be the set of those edges between a player i and his or her neighbors that are to be deleted from G, creating a new game $(\Gamma, G_{\setminus E'})$. The old and the new Penrose-Banzhaf index and the old and the new Shapley-Shubik index of player i can differ as follows:

$$(1 - 2^m)\beta((\Gamma, G), i) \leq \beta((\Gamma, G), i) - \beta((\Gamma, G_{\setminus E'}), i) \leq \beta((\Gamma, G), i);$$
$$(1 - \frac{n!}{(n-m)!})\phi((\Gamma, G), i) \leq \phi((\Gamma, G), i) - \phi((\Gamma, G_{\setminus E'}), i) \leq \phi((\Gamma, G), i).$$

Theorem 5. Control by **deleting edges** between players to *decrease or to increase* a distinguished player's Penrose-Banzhaf index in a graph-restricted weighted voting game is **DP-hard**.

Theorem 6. Control by **deleting edges** between players to *decrease or to increase* a distinguished player's Shapley-Shubik power index is **NP-hard**.

Theorem 7. Control by **deleting edges** between players to *maintain* a distinguished player's Penrose-Banzhaf index in a graph-restricted weighted voting game is **coNP-hard**.

3 Vertex Cover Games

In this section we are going to focus on Vertex Cover Games and how to manipulate them by changing the form of some graph structures for the benefit (or disadvantage) of a single player by increasing the number of the player's vertex covers and the increase (or decrease) of the Shapley-Shubik and Banzhaf indices in each form of manipulation. Before our main goal, we give some general definitions of the Vertex Cover and the Vertex Cover Problem as well as some ways to manipulate vertex cover games regarding the minimum vertex cover.

3.1 Vertex Cover

Definition: A vertex cover of an undirected graph is a subset of its vertices such that for every edge (u,v) of the graph, either 'u' or 'v' is in the vertex cover. Although the name is Vertex Cover, the set covers all edges of the given graph. Given an undirected graph, the vertex cover problem is to find minimum size vertex cover. For a graph G = (V,N) S is a vertex cover (VC) if $\forall e \in E, e = \{u, v\}, \{u, v\} \cap S \neq \emptyset$.

3.2 Cooperation through social influence

Influence Game: A cooperative simple game in which a team (or coalition) of players succeeds if it is able to convince enough agents to participate in the task (to vote in favor of a decision).

In social networks people influence each other through their interactions, and through cooperative game theory, cooperation is achieved by splitting agents into teams so that according to the task of each agent, the pay-off is split among the team members.
Notations used:

- U: finite set
- P(U): U's power set
- $|U|$: U's cardinality
- For any $0 \leq k \leq |U|, P_k(U)$ denotes the subsets of U with exactly k elements
- For any G = (V,E), $n = |V|, m = |E|$
- $G[S]$: the subgraph induced by $S \subseteq V$
- For a vertex $u \in V, N(u) = \{v \in V | (u, v) \in E\}$

Influence Graph: it is a tuple (G, w, f), where G = (V, E) is a weighted, labeled and directed graph (without loops). As usual V is the set of vertices or agents, E is the set of edges and $w : E \rightarrow N$ is a weight function. Finally, $f : V \rightarrow N$ is a labeling function that quantifies how influence-able each agent is. An agent $i \in V$ has influence over another agent $j \in V$ if and only if $(i, j) \in E$. We also consider the family of unweighted influence graphs (G, f) in which every edge has weight 1. Therefore, for a VCG (G,w,f,q):

- w(e) = 1, $\forall e \in E(G)$, the weights of the edges of the graph equal to 1
- $f(v) = deg(v), \forall v \in E(v)$, where f equals to the degree of each vertex v and the influence of the player
- $q = |V|$

3.3 Vertex Cover Game (VCG)

Undirected Graph: Let $G = \langle N, E \rangle, E \neq \emptyset$ an undirected graph, M(G) is the set of least vertex covers of the graph G. A simple game N, v is a vertex cover game of G if $W^m(v) = M(G)$, that is
$v(K) = \begin{cases} 1 & if \exists A \in M(G) : A \subseteq K \\ 0 & \text{otherwise} \end{cases} \quad \forall K \subseteq N.$

3.3.1 VCG

A Vertex Cover Game (N,G) is:

$$V(S) = \begin{cases} 1 & \text{if S is VC} \\ 0 & \text{else} \end{cases}$$

A VC is a set $S \subseteq V$ where every edge is either $u \in S$ or $v \in S$ or both. We use Figure 1 as an example.

In this example, (u,y,z) is a VC and a winning coalition as they cover all edges in the graph. When it concerns vertex covers we don't take the quota into account. The goal is to find a critical graph S inside G so that we consider the coalition critical and therefore winning, i.e.:
α-Critical Graphs: Graphs are called critical if they are minimal with regards to a certain measure. More precisely, an edge of a given graph G is called a critical element iff its deletion would decrease the measure. G is then called edge-critical (or simply critical) iff every edge is a critical element. The concept of critical graphs is mostly used in works on the chromatic number. Though, it can also be of significant interest for the MVC (Minimum Vertex Cover) problem, where we call a graph critical iff

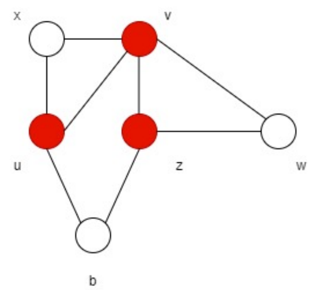

Figure 1: The vertices in red form the minimum vertex cover

the deletion of any edge would decrease the size of the minimal cover.
Vertex Cover Problem (VCP). The VCP is a known NP-hard problem in computer science and graph theory that covers the minimum number of vertices of a general network where at least one endpoint of each edge of the network is covered. It involves finding the minimum number of vertices needed to cover all the edges of a graph.
Goal: Select minimum-size subset $C \subseteq V$ such that for each edge $(u, v) \in E$ we have $u \in C$ and/or $v \in C$.

Definition. (Vertex cover problem) Given an undirected graph Ξ = (V, E) with the set of vertices V = {1, 2, . . . , n} and the set of edges $E = \{e_{ij}\}(i, j \in V, i \neq j)$, if there exists an edge from vertex i to vertex j, then $e_{ij} = 1$; otherwise, $e_{ij} = 0$. A vertex cover is defined as a set V_{VC} such that each $e_{ij}(e_{ij} \in E)$ has at least one endpoint (vertex) in the set V_{VC}. A minimum vertex cover V_{MVC} is a vertex cover with the minimum cardinality, i.e., the number of covered vertices is minimum.
Using the information abive, in image 1, node u is critical for $S \in G$. $u \in S$ and S is a VC but S-u is not. We consider each node an agent and each edge as the influence of each agent over the other. By manipulating the graph and therefore the VC we can change each agents influence. In the next subsection, we will see some of the ways to manipulate the VCP.

3.3.2 To Manipulate into the minimum Vertex Cover

1. **Adding or removing edges** One way to manipulate the vertex cover problem is to add or remove edges from the graph. Adding edges can make the problem harder, while removing edges can make it easier. For example, let's say we have the original graph below:

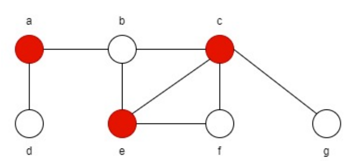

Figure 2.

It is obvious that the MVC for the graph is (a,c,e). If we remove the edges between (b,e) and (e,f) we have the following:

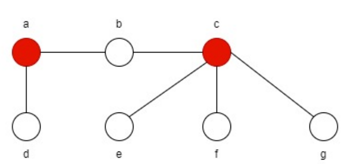

Figure 3.

Where the MVC has been reduced to (a,c). Furthermore, for e second example, we could remove from the original graph the edge between (a,d) end (e,f) while adding another edge between (b,d) and we have:

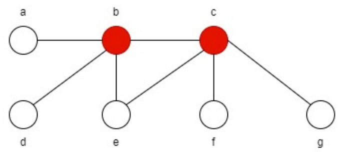

Figure 4.

Where the MVC again has been reduced to (b,c) as it covers all the edges.

2. **Creating a false identity** By using the example of Image 1 we can manipulate the VC by adding a new node to the graph replicating an already existing one, i.e creating a false identity.

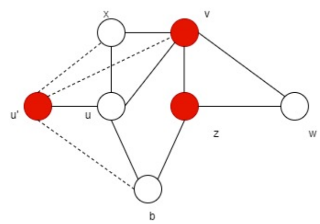

Figure 5.

The new graph is G'. In Figure 1 u was critical for $S \in G$, $u \in S$ was a VC but S-u was not because there exists a $v \in N(u)$ while $v \notin S$. Now, $S \in G'$ is a VC, S-u is not while u is critical for $S \in G'$. In the new graph, the $S' = S + u' \in G'$ is a vertex cover while S'-u is not while u is again critical for $S' \in G'$. The new false identity u' is critical for S if $u \notin S$ and S-u' is a VC of G.

3. **Changing the graph structure** Similar to creating a false identity, another way to manipulate the vertex cover problem is to change the structure of the graph itself. For example, you could add or remove nodes, or change the degree of nodes to change the difficulty of the problem. Using Figure 2 as an example:

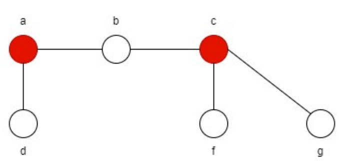

Figure 6.

Here, we have removed node e and reduced the minimum vertex cover to (a,c).

3.3.3 To change the number of VCs of a single player

We take as an example the graph of Figure 6 below for the increase of vertex covers of player u and changing the Shapley-Shubik and Banzhaf indices.

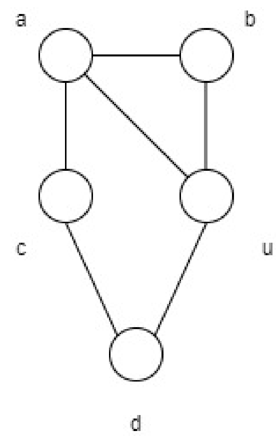

Figure 7.

In the graph above, the VCs for the players are (u,a,c), (u,a,d), (u,b,c,d), (u,a,b,d), (u,b,c), (u,a,b,c), (u,a,b,c,d), (u,a,c,d), (a,b,d) and (a,b,c,d). We can see that player u is not contained in some of the coalitions. To measure the change of a player's power we are going to use the Banzhaf index as the SS requires more complexity and we don't take the players' order in a coalition into account for the sake of finding the pivotal player.
As it has already been mentioned, the BI is measured by the subtraction of the sum of coalitions that player u is not a part of, and therefore the *losing* coalitions, from the coalitions that they are part of, meaning the *winning* coalitions. The Banzhaf index has been widely used, though primarily for the purpose of measuring individual power in a weighted voting system. However, it can also easily be applied to any simple coalitional game.
The Banzhaf index depends on the number of coalitions in which an agent is a swinger, out of all possible coalitions. As mentioned in 2.4.2, the Banzhaf index is given by $\beta(v) = (\beta_1(v), ..., \beta_n(v))$ where

$$\beta_i(v) = \frac{1}{2^{n-1}} \sum_{S \subset N | i \in S} [v(S) - v(S \backslash \{i\})].$$

where n is the number of players.

Different probabilistic models on the way a coalition is formed yield different appropriate power indices. The BI reflects the assumption that the agents are independent in their choices.
Now, to calculate the Banzhaf index of the graph above, we realize that the number of winning coalitions that u is part of equals to 7, meanwhile the number of losing coalition that do not contain u is 2. Consequently, the value of the index equals to $\beta_u(G) = \frac{8-2}{2^4} = \frac{6}{2^4}$. We will use this result to compare the new results of the newly formed graphs in our examples of manipulation.
Below have been proposed some ways to manipulate the Vertex Cover:

1. **Adding or removing edges** By adding or removing an edge from the graph we could help the player be a part of a coalition that they were not before. For example, in the graph of Figure 7 there is the coalition (b,d,a) in which player u is not included, therefore, to make them part of said coalition we add a new edge between c and themselves and as a result we create the new coalition (u,b,d,a) as in Figure 8.

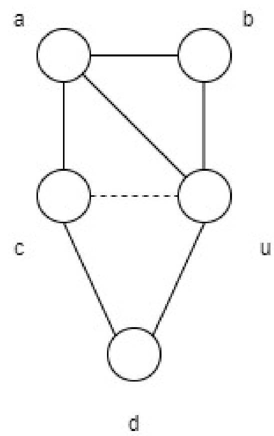

Figure 8.

The new coalitions are: (u,a,d), (u,a,c), (u,b,c), (u,a,b,d), (u,a,b,c), (u,b,c,d), (u,a,b,c,d) of which u is a part of and the only coalition of which they are not is (a,b,c,d). To calculate the BI, there still are 7 winning coalitions but one losing, meaning that the new BI of the new graph G' will be: $\beta'_u(G') = \frac{7-1}{2^4} = \frac{6}{2^4}$. It is obvious that the BI and the player's power have been maintained. That being the case, in our example by adding an edge between u and c the manipulation has not changed the outcome.

Below, we present the code in python used in order to calculate all the possible VCs of a graph. Now, we took as an example the graph of Figure 2.

```
import networkx as nx
import matplotlib.pyplot as plt
from itertools import combinations

def find_vertex_covers(G):    #takes the graph object G as input and returns a list of all VCs
    vertex_covers = []  #store VCs
    vertices = set(G.nodes()) #store vertices
    edges = set(G.edges())    #store edges
    for r in range(1, len(vertices) + 1):  #iteration of different sizes of combs of vertices
        vertex_combinations = combinations(vertices, r) #generate combs 'r'
        for combination in vertex_combinations:
            is_cover = True  #if current comb is VC
            for edge in edges:
                if edge[0] not in combination and edge[1] not in combination: #checks if bot endpoints are not present in the comb
                    is_cover = False #is not VC
                    break
            if is_cover:
                vertex_covers.append(combination)
    return vertex_covers


#Graph
G=nx.Graph()

#vertices
G.add_node("a")
G.add_node("b")
G.add_node("c")
```

```
G.add_node("d")
G.add_node("e")
G.add_node("f")
G.add_node("g")

#edges
G.add_edge("a", "d")
G.add_edge("a", "b")
G.add_edge("b", "c")
G.add_edge("b", "e")
G.add_edge("c", "e")
G.add_edge("c", "f")
G.add_edge("c", "g")
G.add_edge("e", "f")

#fixed position
pos = {
    "a": (1, 3),
    "b": (4, 3),
    "c": (7, 3),
    "d": (1, 1),
    "e": (4, 1),
    "f": (7, 1),
    "g": (10, 1)
}

all_vertex_covers = find_vertex_covers(G)

#print all the vcs
for vertex_cover in all_vertex_covers:
    print(vertex_cover)

nx.draw(G, pos=pos, with_labels=True, node_color='r', node_size=2000, width=5)
plt.margins(0.2)
```

We tested the code according to player c. At first, we printed the results that contained the edge between vertices c and e. The results are as follows:

```
('c', 'e', 'a')
('g', 'c', 'e', 'a')
('c', 'e', 'a', 'f')
('c', 'e', 'a', 'b')
('c', 'e', 'a', 'd')
('c', 'e', 'b', 'd')
('c', 'a', 'f', 'b')
('c', 'f', 'b', 'd')
('g', 'c', 'e', 'a', 'f')
('g', 'c', 'e', 'a', 'b')
('g', 'c', 'e', 'a', 'd')
('g', 'c', 'e', 'b', 'd')
('g', 'c', 'a', 'f', 'b')
('g', 'c', 'f', 'b', 'd')
('g', 'e', 'a', 'f', 'b')
('g', 'e', 'f', 'b', 'd')
('c', 'e', 'a', 'f', 'b')
('c', 'e', 'a', 'f', 'd')
('c', 'e', 'a', 'b', 'd')
```

```
('c', 'e', 'f', 'b', 'd')
('c', 'a', 'f', 'b', 'd')
('g', 'c', 'e', 'a', 'f', 'b')
('g', 'c', 'e', 'a', 'f', 'd')
('g', 'c', 'e', 'a', 'b', 'd')
('g', 'c', 'e', 'f', 'b', 'd')
('g', 'c', 'a', 'f', 'b', 'd')
('g', 'e', 'a', 'f', 'b', 'd')
('c', 'e', 'a', 'f', 'b', 'd')
('g', 'c', 'e', 'a', 'f', 'b', 'd')
```

And the total number of coalitions is **29**. Then, we removed the edge between c and e from the graph and it printed the following results:

```
('a', 'c', 'e')
('f', 'g', 'a', 'b')
('f', 'g', 'd', 'b')
('f', 'a', 'c', 'e')
('f', 'a', 'c', 'b')
('f', 'c', 'd', 'b')
('g', 'a', 'c', 'e')
('a', 'c', 'e', 'd')
('a', 'c', 'e', 'b')
('c', 'e', 'd', 'b')
('f', 'g', 'a', 'c', 'e')
('f', 'g', 'a', 'c', 'b')
('f', 'g', 'a', 'e', 'b')
('f', 'g', 'a', 'd', 'b')
('f', 'g', 'c', 'd', 'b')
('f', 'g', 'e', 'd', 'b')
('f', 'a', 'c', 'e', 'd')
('f', 'a', 'c', 'e', 'b')
('f', 'a', 'c', 'd', 'b')
('f', 'c', 'e', 'd', 'b')
('g', 'a', 'c', 'e', 'd')
('g', 'a', 'c', 'e', 'b')
('g', 'c', 'e', 'd', 'b')
('a', 'c', 'e', 'd', 'b')
('f', 'g', 'a', 'c', 'e', 'd')
('f', 'g', 'a', 'c', 'e', 'b')
('f', 'g', 'a', 'c', 'd', 'b')
('f', 'g', 'a', 'e', 'd', 'b')
('f', 'g', 'c', 'e', 'd', 'b')
('f', 'a', 'c', 'e', 'd', 'b')
('g', 'a', 'c', 'e', 'd', 'b')
('f', 'g', 'a', 'c', 'e', 'd', 'b')
```

Where the total number of coalitions is **31**.

It is obvious that the total number of players is 7.
Case 1: The edge between c and e exists. The number of coalitions that player c is a part of equals to 26, meanwhile they are not part of 3. Therefore, the BI of player c in that case is $\beta_c(G_{\{ce\}}) = \frac{26-3}{2^{7-1}} = \frac{23}{2^6}$.
Case 2: Removing the edge between c and e. The number of coalitions that player c is a part of equals to 26 and the number that it's not part of to 5. Therfore, the BI equals to $\beta_c(G_{\setminus\{ce\}}) = \frac{21}{2^6}$.
From these two cases, it is apparent that by removing the edge the Banzhaf Index for 'c' will decrease.

Case 3: Adding an edge between e and g In the following case we tried to test the possibility of increasing another player's index in order to put player c in disadvantage. The results are the ones below:

```
('a', 'e', 'c')
('b', 'a', 'e', 'c')
('b', 'e', 'c', 'd')
('f', 'a', 'e', 'c')
('a', 'e', 'g', 'c')
('a', 'e', 'c', 'd')
('b', 'f', 'a', 'e', 'g')
('b', 'f', 'a', 'e', 'c')
('b', 'f', 'a', 'g', 'c')
('b', 'f', 'e', 'g', 'd')
('b', 'f', 'e', 'c', 'd')
('b', 'f', 'g', 'c', 'd')
('b', 'a', 'e', 'g', 'c')
('b', 'a', 'e', 'c', 'd')
('b', 'e', 'g', 'c', 'd')
('f', 'a', 'e', 'g', 'c')
('f', 'a', 'e', 'c', 'd')
('a', 'e', 'g', 'c', 'd')
('b', 'f', 'a', 'e', 'g', 'c')
('b', 'f', 'a', 'e', 'g', 'd')
('b', 'f', 'a', 'e', 'c', 'd')
('b', 'f', 'a', 'g', 'c', 'd')
('b', 'f', 'e', 'g', 'c', 'd')
('b', 'a', 'e', 'g', 'c', 'd')
('f', 'a', 'e', 'g', 'c', 'd')
('b', 'f', 'a', 'e', 'g', 'c', 'd')
```

The total number of coalitions equals to **26**.
Player c is part of 23 total coalitions, thus their BI equals to $\frac{23-3}{2^6} = \frac{20}{2^6}$. However, we can see that the same stands now for player e with the same BI (while before adding the edge player e's BI was $\beta_e(G) = \frac{17}{2^6}$).

2. **Creating a false identity** We can manipulate the VC by adding a new node to the graph replicating an already existing one, i.e creating a false identity.

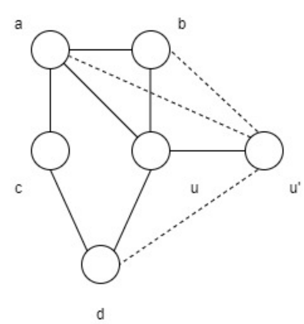

Figure 9.

The new graph is G'. In Figure 7 u was critical for $S \in G$, $u \in S$ was a VC but S-u was not because there exists a $v \in N(u)$ while $v \notin S$. Now, $S \in G'$ is a VC, S-u is not while u is critical for $S \in G'$. In the new graph, the $S' = S + u' \in G'$ is a vertex cover while S'-u is not while u is again critical for $S' \in G'$. The new false identity u' is critical for S if $u \notin S$ and S-u' is a VC of G.
By using the code from the previous example, we created the graph of Figure 7.

```
    #fig 7 graph
G.add_edge("a","b")
G.add_edge("a","c")
G.add_edge("a","u")
G.add_edge("b","u")
G.add_edge("c","d")
G.add_edge("u","d")

#fixed position
pos = {
    "a": (1, 7),
    "b": (4, 7),
    "c": (1, 4),
    "d": (2.3, 1),
    "u": (4,4)
}
```

and the results are as we have already calculated at the beginning of the section

```
('b', 'd', 'a')
('b', 'c', 'u')
('d', 'a', 'u')
('c', 'a', 'u')
('b', 'd', 'c', 'a')
('b', 'd', 'c', 'u')
('b', 'd', 'a', 'u')
('b', 'c', 'a', 'u')
('d', 'c', 'a', 'u')
('b', 'd', 'c', 'a', 'u')
```

and the index as previously calculated is $\frac{6}{2^4}$. Later on, we create the graph of Figure 9 to see how the index has changed for player 'u'.
The new results of graph G' are:

```
('u', 'd', 'u2', 'a')
('u', 'd', 'b', 'a')
('u', 'c', 'u2', 'b')
('u', 'c', 'u2', 'a')
('d', 'u2', 'b', 'a')
('u', 'd', 'c', 'u2', 'b')
('u', 'd', 'c', 'u2', 'a')
('u', 'd', 'c', 'b', 'a')
('u', 'd', 'u2', 'b', 'a')
('u', 'c', 'u2', 'b', 'a')
('d', 'c', 'u2', 'b', 'a')
('u', 'd', 'c', 'u2', 'b', 'a')
```

where u2 represents the false identity of u, u'. As it is obvious, now player u is part of every coalition of the graph. At first, they were part of 8 coalitions and after the manipulation they are part of 12 coalitions. Therefore, by creating a false identity we can manipulate the VC only for the benefit of the player.
Then, we took as an example the graph of Figure 5 without the false identity of u, u'.

```
    #fig 5
G.add_node("x")
G.add_node("v")
G.add_node("u")
G.add_node("z")
G.add_node("w")
```

```
G.add_node("b")

G.add_edge("x","v")
G.add_edge("x","u")
G.add_edge("u","v")
G.add_edge("u","b")
G.add_edge("b","z")
G.add_edge("z","v")
G.add_edge("z","w")
G.add_edge("w","v")

#fixed position fig 5
pos = {
    "x": (1,7),
    "v": (4,7),
    "u": (1,4),
    "z": (4,4),
    "w": (7,4),
    "b": (2.3,1)
}
```

with the coalitions

```
('u', 'v', 'z')
('u', 'x', 'v', 'z')
('u', 'x', 'w', 'z')
('u', 'v', 'b', 'w')
('u', 'v', 'b', 'z')
('u', 'v', 'w', 'z')
('x', 'v', 'b', 'w')
('x', 'v', 'b', 'z')
('u', 'x', 'v', 'b', 'w')
('u', 'x', 'v', 'b', 'z')
('u', 'x', 'v', 'w', 'z')
('u', 'x', 'b', 'w', 'z')
('u', 'v', 'b', 'w', 'z')
('x', 'v', 'b', 'w', 'z')
('u', 'x', 'v', 'b', 'w', 'z')
```

and we figured out that there are 15 total coalitions, of which "u" is part of 12 and the BI is calculated as $\beta_u(G) = \frac{9}{2^5}$. Meanwhile, in this graph (G), the player with the biggest partitions in coalitions is "v", with a BI of $\beta_v(G) = \frac{11}{2^5}$. Now, we create the graph G' of Figure 5 that will help player 'u' increase his index and surpass 'v'.

```
#fig 5
G.add_node("x")
G.add_node("v")
G.add_node("u")
G.add_node("z")
G.add_node("w")
G.add_node("b")
G.add_node("u2")

G.add_edge("x","v")
G.add_edge("x","u")
G.add_edge("u","v")
G.add_edge("u","b")
G.add_edge("b","z")
```

```
G.add_edge("z","v")
G.add_edge("z","w")
G.add_edge("w","v")

#false identity
G.add_edge("x","u2")
G.add_edge("v", "u2")
G.add_edge("b", "u2")
G.add_edge("u","u2")

#fixed position fig 5
pos = {
    "x": (1,7),
    "v": (4,7),
    "u": (1,4),
    "z": (4,4),
    "w": (7,4),
    "b": (2.3,1),
    "u2": (-4, 4)
}
```

with coalitions

```
('z', 'u2', 'v', 'u')
('z', 'w', 'x', 'u2', 'u')
('z', 'w', 'u2', 'v', 'u')
('z', 'b', 'x', 'u2', 'v')
('z', 'b', 'x', 'v', 'u')
('z', 'b', 'u2', 'v', 'u')
('z', 'x', 'u2', 'v', 'u')
('w', 'b', 'x', 'u2', 'v')
('w', 'b', 'x', 'v', 'u')
('w', 'b', 'u2', 'v', 'u')
('z', 'w', 'b', 'x', 'u2', 'v')
('z', 'w', 'b', 'x', 'u2', 'u')
('z', 'w', 'b', 'x', 'v', 'u')
('z', 'w', 'b', 'u2', 'v', 'u')
('z', 'w', 'x', 'u2', 'v', 'u')
('z', 'b', 'x', 'u2', 'v', 'u')
('w', 'b', 'x', 'u2', 'v', 'u')
('z', 'w', 'b', 'x', 'u2', 'v', 'u')
```

With this change, as a result we have 18 coalitions of which u, with their false identity u' (which is presented as 'u2'), is part of all. The new BI is $\beta'_{\{u\&u2\}}(G') = \frac{18}{2^5}$. At the same time, the BI of player 'v' was also increased at $\beta'_v(G') = \frac{14}{2^5}$, without being the winning player anymore.

3. **Changing the graph structure** Similar to what was stated previously, we can change the graph structure by adding or removing nodes to change the difficulty of the problem.
   Using the new graph of Figure 3 and also changing the graph of Figure 2 in various ways by implementing the code, the new coalitions are:

```
('c', 'a')
('b', 'd', 'c')
('b', 'c', 'a')
('d', 'c', 'a')
('c', 'f', 'a')
('c', 'g', 'a')
('b', 'd', 'c', 'f')
```

```
('b', 'd', 'c', 'g')
('b', 'd', 'c', 'a')
('b', 'd', 'f', 'g')
('b', 'c', 'f', 'a')
('b', 'c', 'g', 'a')
('b', 'f', 'g', 'a')
('d', 'c', 'f', 'a')
('d', 'c', 'g', 'a')
('c', 'f', 'g', 'a')
('b', 'd', 'c', 'f', 'g')
('b', 'd', 'c', 'f', 'a')
('b', 'd', 'c', 'g', 'a')
('b', 'd', 'f', 'g', 'a')
('b', 'c', 'f', 'g', 'a')
('d', 'c', 'f', 'g', 'a')
('b', 'd', 'c', 'f', 'g', 'a')
```

We count 23 coalitions from which player 'c' is part of 20. Meanwhile, the total number of players (without vertex 'e') is 6. Therefore, the new BI is calculated as $\beta'_c(G') = \frac{17}{2^5} > \frac{23}{2^6}$ thus by removing node 'e' the BI for 'c' has increased.
We tested this case also by removing node 'd'. The results are the 18 following coalitions:

```
('e', 'b', 'c')
('e', 'c', 'a')
('f', 'b', 'c')
('e', 'f', 'b', 'g')
('e', 'f', 'b', 'c')
('e', 'f', 'c', 'a')
('e', 'b', 'g', 'c')
('e', 'b', 'c', 'a')
('e', 'g', 'c', 'a')
('f', 'b', 'g', 'c')
('f', 'b', 'c', 'a')
('e', 'f', 'b', 'g', 'c')
('e', 'f', 'b', 'g', 'a')
('e', 'f', 'b', 'c', 'a')
('e', 'f', 'g', 'c', 'a')
('e', 'b', 'g', 'c', 'a')
('f', 'b', 'g', 'c', 'a')
('e', 'f', 'b', 'g', 'c', 'a')
```

From which 'c' is part of 16 and their new BI is $\beta''_c(G'') = \frac{14}{2^5} > \frac{23}{2^6}$.
Finally, we tested the new BI for 'c' after removing vertex 'g' and by doing that, intuitively, we suppose that the index will be decreased as 'c' is the only connection there is to 'g'. The results are:

```
('e', 'a', 'c')
('e', 'a', 'd', 'c')
('e', 'a', 'b', 'f')
('e', 'a', 'b', 'c')
('e', 'a', 'f', 'c')
('e', 'd', 'b', 'f')
('e', 'd', 'b', 'c')
('a', 'b', 'f', 'c')
('d', 'b', 'f', 'c')
('e', 'a', 'd', 'b', 'f')
('e', 'a', 'd', 'b', 'c')
('e', 'a', 'd', 'f', 'c')
```

```
('e', 'a', 'b', 'f', 'c')
('e', 'd', 'b', 'f', 'c')
('a', 'd', 'b', 'f', 'c')
('e', 'a', 'd', 'b', 'f', 'c')
```

with 16 as the number of the new total coalitions and $\beta_c'''(G''') = \frac{10}{2^5} < \frac{23}{2^6}$ which proves our initial thought.

4. **Merging - Annexing**
   Instead of a player splitting into smaller players, some players may merge into a single entity to increase the number of their total VCs.

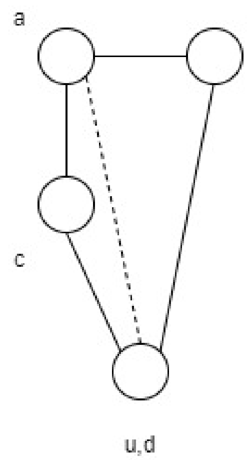

Figure 11.

In the example of Figure 10 by merging the players u and d we have created the extra participation of player u in the coalitions $(a, \{u, d\}, c), (a, b, \{d, u\}), (a, \{u, d\})$. We modified the code to get the results for the new graph of Figure 11 and the results were the following:

```
('{u,d}', 'a')
('c', 'b', '{u,d}')
('c', 'b', 'a')
('c', '{u,d}', 'a')
('b', '{u,d}', 'a')
('c', 'b', '{u,d}', 'a')
```

Which confirms the initial results. If we consider players u,d as one, provided that they share the same node, the new BI is $\beta_{\{u\&d\}}(G') = \frac{6-1}{2^{4-1}} = \frac{5}{2^3} > \frac{6}{2^4}$. Thus, the index for the merged player u has increased, even though now it is the same as of player d for whom the initial BI was $\beta_d(G) = \frac{4}{2^4} < \frac{5}{2^3}$ and as a consequence we have benefited both players.
We could also merge player u with player b and form the new graph and extra coalitions for u as in the graph below:

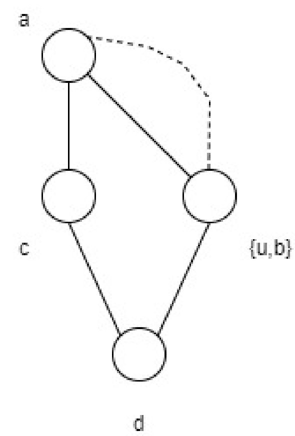

Figure 12.

Now, we have created for u the coalitions $(a, d, \{u, b\}), (c, d, \{u, b\}), (c, \{u, b\}), (c, a, \{u, b\})$. We confirm this by implementing again the code and the results are:

```
('c', '{u,b}')
('a', 'd')
('c', 'a', '{u,b}')
('c', 'a', 'd')
('c', '{u,b}', 'd')
('a', '{u,b}', 'd')
('c', 'a', '{u,b}', 'd')
```

The BI for u,b is $\beta_{u\&b}(G'') = \frac{3}{2^3} > \frac{5}{2^4}$, however it is less than $\beta_{u\&d}$. For that being the case, we can conclude that between the two mergings the merging of the edges u,d is more beneficial to player u.

4 Conclusions

In this report we have covered the general definitions of Weighted Voting Games on simple graphs and the representation of vertices as players and coalitions and how the Banzhaf Index indicates each player's influence in Vertex Cover Games. We have implemented various ways through code to manipulate the Vertex Cover Game for the advantage or disadvantage of the player.
In the simple graphs we have presented, it is obvious that the BI will most likely be increased or maintained for changes in the graph that concern our focused player. For example, by adding an edge between two edges, by creating a false identity of a vertex or by removing opposing vertices the index has a higher probability of increasing, whereas by removing an edge from the said player their index will most likely decrease. We have also shown that whether a beneficial manipulation exists through splitting or annexation is NP-hard.
However, as graph structures can get more complicated, there are other ways that can be studied in order to have more possible results and ways to manipulate the vertex cover and change the influence of a player amongst others.

References

Haris Aziz, Yoram Bachrach, Edith Elkind, Mike Paterson (2011). False-Name Manipulations in Weighted Voting Games.

Joanna Kaczmarek, Jörg Rothe (2021). Manipulation in Communication Structures of Graph-Restricted Weighted Voting Games.

Xavier Molinero, Fabián Riquelme, Maria Serna (2014). Cooperation through social influence.

Vasily V. Gusev (2019). The vertex cover game: Application to transport networks.

Andreas Jakoby, Naveen Kumar Goswami, Eik List, Stefan Lucks Faculty of Media, Bauhaus-Universitat Weimar, Bauhausstr (2018). Critical Graphs for Minimum Vertex Cover.