Efficient Computation of the Shapley Value for Centrality in Networks

Karthik V. Aadithya[1] Balaraman Ravindran[2] Tomasz Michalak[3] Nicholas R. Jennings[3]

[1]The University of California, Berkeley, USA
aadithya@eecs.berkeley.edu
[2]Indian Institute of Technology Madras, India
ravi@cse.iitm.ac.in
[3]School of Electronics and Computer Science, University of Southampton, UK
{tpm,nrj}@ecs.soton.ac.uk

Abstract. The Shapley Value is arguably the most important normative solution concept in coalitional games. One of its applications is in the domain of networks, where the Shapley Value is used to measure the relative importance of individual nodes. This measure, which is called node centrality, is of paramount significance in many real-world application domains including social and organisational networks, biological networks, communication networks and the internet. Whereas computational aspects of the Shapley Value have been analyzed in the context of conventional coalitional games, this paper presents the first such study of the Shapley Value for network centrality. Our results demonstrate that this particular application of the Shapley Value presents unique opportunities for efficiency gains. In particular, we develop exact analytical formulas for computing Shapley Value based centralities in both weighted and unweighted networks. These formulas not only provide an efficient (polynomial time) and error-free way of computing node centralities, but their surprisingly simple closed form expressions also offer intuition into why certain nodes are relatively more important to a network.

1 Introduction

The Shapley Value (SV) is a fundamental normative solution concept in coalitional games. Given a scenario where agents are allowed to realize collective payoffs through mutual co-operation, the SV postulates a fair method to evaluate each agent's individual contribution. One of the many applications of the SV is in the domain of networks, where it is used to measure the importance of individual nodes, which is known as *game theoretic network centrality* [1, 2]. Although centrality plays a key role in many real-life network applications, efficient algorithms for its measurement via the SV remain unknown.

In fact, to date, computational aspects of the SV have been studied only in the context of conventional coalitional games. To the best of our knowledge, this paper is the first study of the computational aspects of the SV, as applied to network centrality. We believe that the additional challenges that arise in this context can be overcome by harnessing the unique opportunities available for efficiency gains, stemming from the underlying network structure. In this paper, we address the question of how to take advantage of this special structure to compute SV-based centralities efficiently. Indeed, our research shows that it is possible to develop efficient (polynomial time) and exact algorithms for SV computation in the networks domain. By contrast, SV computation has been proved to be a hard problem for many other structured domains [3–5]. Therefore, we believe that the algorithms presented in this paper represent rare examples of practically relevant applications that nevertheless permit efficient SV solutions[1].

We now introduce the concept of "centrality". In the networks context, it is often paramount to determine which nodes and edges are more critical than others. Classic examples include identifying the most important highways in a road network, the most influential people in a social network or the most critical functional entities in a protein network. As a result, the concept of centrality, which aims to quantify the importance of individual nodes/edges in a network, has been extensively studied in network analysis.

Conventional centrality measures[2] usually work by assigning a score to each node in the network, which in some way corresponds to the importance of that node for the application at hand. For instance, if the application is to design an infrastructure network (such as a power transmission network or communication network) with minimum vulnerability to random node failures, a conventional centrality measure might work by analysing the consequences of failure at each individual node. The more adverse the consequences of failure, the higher the node centrality.

Such a conventional centrality metric, however, suffers from the following drawbacks:

1. By considering only the failure of individual nodes, it completely ignores real-world situations where multiple nodes can fail simultaneously. For example, if the network is so designed that no single node's failure carries any consequence, but the failure of certain specific *pairs of nodes* can bring down the entire network, the above centrality metric would fail to give a higher centrality score to the nodes belonging to these critical pairs.

[1] Other positive results on SV computation (e.g. MC-Nets [6]) are discussed in the Related Work section

[2] An overview of conventional centrality measures (such as degree centrality, betweenness centrality, closeness centrality and eigenvalue centrality) can be found in [7].

2. Because each node is treated separately, the hidden assumption is that node failures occur independently of each other. As a result, real-world phenomena such as cascading node failures, that have been known to precipitate widespread disruption in a very short time [8], are outside the scope of this centrality analysis.

In short, conventional centrality measures fail to recognize that in many network applications, it is not sufficient to merely understand the relative importance of nodes as stand-alone entities. Rather, the key requirement is to understand the importance of each node *in terms of its utility when combined with other nodes* [9]. For instance, in the above infrastructure network, an ideal centrality measure would assign a score to a node v based on the failure probabilities (and consequences thereof) of *every subset of nodes containing* v, rather than just failure of the single node v. This approach would automatically allow the ideal centrality measure to give due consideration to real-world failure patterns such as cascading failures and simultaneous multiple node failures. On the other hand, this flexibility, which comes from the ability to take into account the contributions of all possible combinations of nodes (rather than just one node at a time), is absent in conventional centrality measures, which is a crucial limitation in many applications.

Game theoretic network centrality [1,2] has been proposed as a framework that would address the above limitation. Given the network to be analysed, the idea is to define a co-operative game where the agents (players) are the nodes of the network. Then the SV of each agent (node) in this game is interpreted as a *centrality measure* because it represents the average marginal contribution made by each node to every possible combination of the other nodes. This powerful paradigm of SV-based network centrality thus confers a high degree of flexibility (which was completely lacking in traditional centrality metrics) to model real-world network phenomena. Indeed, this new paradigm has already been proved to be more useful than traditional centrality measures for certain real-life network applications [1,10].

From a computational perspective, however, evaluating game theoretic network centrality using the original SV formula involves an analysis of the marginal contribution of every node (i.e. player) to every coalition. Thus, given a network $G(V,E)$, a direct application of the SV formula involves considering $O(2^{|V(G)|})$ coalitions. Such an exponential computation is clearly prohibitive for bigger networks (of, e.g, 100 or 1000 nodes). For such networks, the only feasible approach currently outlined in the literature is Monte-Carlo sampling, which is not only inexact, but also very time-consuming.

The above problem of *exponential complexity in the number of agents* is a fundamental challenge associated with computing the SV. As a result, for conventional coalitional games, this issue has received considerable attention in the literature. As an alternative to the straightforward (but exponential) listing of all possible coalitions, some authors [3,6] have proposed more efficient representations for coalitional games. In addition to being concise for many games, these representations also possess desirable computational properties, including efficient SV computation[3]. Thus, the *choice of representation* has been the foremost consideration for efficient SV computation in the context of conventional coalitional games.

The networks domain, by contrast, poses a very different set of challenges. Unlike conventional coalitional games, conciseness is usually not an issue in the networks context. This is because the games that aim to capture network centrality notions already specify a *concise closed-form expression* for evaluating coalition values (please see next section for an example). But on the other hand, the exact functional forms of these expressions are dictated not by computational considerations, but by the real-world applications of *game theoretic network centrality*. Therefore, while the available representations for these games are concise, they are already fixed by the application at hand, i.e, beyond a point, they cannot be changed to suit computational convenience. Moreover, because these games are designed to reflect network centrality, their definition of coalition values often depends highly non-trivially on the underlying graph structure. For example, this definition may involve quantities such as shortest path lengths, adjacent vertex counts etc. with respect to the underlying graph.

Therefore, the challenge we face in this paper is to *efficiently compute the SV, given a network and a game defined over it, where coalition values for this game are given by a closed-form expression that depends non-trivially on the network*. The key question here is how to take advantage of (a) the network structure, and (b) the functional form for the coalition values, so as to compute SVs efficiently, i.e, without the need to enumerate all possible coalitions.

Against this background:

[1] Our key contribution in this paper is to demonstrate that it is possible to **exactly and efficiently compute SV-based network centralities** of practical interest **defined on large networks** which exceed thousands of nodes! By contrast, the only previously known method that scaled to such large networks was Monte-Carlo simulation, which was neither exact nor particularly efficient.

[2] For four different measures of network centrality, we develop **exact closed-form formulas** for the SVs. We present pseudo-codes of **linear and polynomial time algorithms** to implement these formulas.

[3] For more details, please see the Related Work section

[3] We develop **a closed-form polynomial time computable SV approximation** for a fifth measure of centrality defined on weighted networks.

[4] We test our algorithms on **two real-life examples**, (a) an infrastructure network representing the topology of the Western States Power Grid, and (b) a collaboration network from the field of astrophysics. The results show that the algorithms proposed in this paper are not only accurate but also deliver significant speedups (upto 550 times for the 16000+ node collaboration network!) over Monte-Carlo simulation.

The remainder of the paper is organized as follows. Section 2 presents an example of how a coalitional game may be used to capture the notion of network centrality. Section 3 analyses five types of centrality-related coalitional games and presents polynomial time SV algorithms for all of them. Section 4 discusses related work. Conclusions follow. Finally, the appendix presents numerical simulation results.

2 SV as a Centrality Measure

As mentioned in the introduction, the concept of game theoretic network centrality based on the SV has been proposed in [1, 2] and further explored in [10]. Here we illustrate this concept with an example.

Consider the notion of "closeness centrality" of a node in a graph $G(V, E)$, which is traditionally defined as the reciprocal of the average distance of that node from other (reachable) nodes in the graph [7]. This definition captures the intuitive idea that a node "in close proximity to many other nodes" is more valuable by virtue of its central location, and hence should be assigned a higher centrality score.

The above measure, however, fails to recognize the importance of combinations of nodes. For example, consider a typical real-world application of closeness centrality: that of disseminating a piece of information to all nodes in the network. At any time point t in the dissemination process, define the random variable C_t to be the subset of nodes most actively involved in propagating the information. In this situation, a new node added to C_t would make maximum contribution to the diffusion of information only if it is "in close proximity to nodes that are not currently in close proximity to any node in C_t". Thus, while conventional closeness centrality only takes into account *average proximity to all other nodes*, the actual importance of a node in the real-world application is based on a very different measure: *proximity to nodes that are not in close proximity to the random variable* C_t.

We now show how coalitional game theory can be used to construct a centrality measure that faithfully models the above *real-world importance* of a node. Let C be any subset of nodes from the given network $G(V, E)$. Then, for every such C, assign a value $\nu(C)$ given by

$$\nu(C) = \sum_{v \in V(G)} \frac{1}{1 + \min\{d(u, v) | u \in C\}}$$

where $d(u, v)$ is the distance between nodes u and v (measured as the shortest path length between u and v in graph G).

The map ν defined above captures a fundamental centrality notion: that the *intrinsic value* of a subset of nodes C in the context of a real-world application (such as information dissemination) is proportional to the overall proximity of the nodes in C to the other nodes in the network. In effect, the map ν carries the original definition of closeness centrality to a global level, where a measure of importance is assigned to every possible combination of nodes.

The map ν above is therefore a *characteristic function* for a coalitional game, where each vertex of the network is viewed as an agent playing the game. It follows that if a node v has a high SV in this game, it is likely that v would "contribute more" to an arbitrary randomly chosen coalition of nodes C in terms of increasing the proximity of C to other nodes on the network. Thus, computing the SVs of this game yields a centrality score for each vertex that is a much-improved characterization of closeness centrality.

The only difficulty in adopting such a game-theoretically inspired centrality measure is the previously mentioned problem of *exponential complexity in the number of agents*. In the next section, we show how to overcome this difficulty and compute the SV for many centrality applications (including the above formulation) in time polynomial in the size of the network.

3 Algorithms for SV-based network centrality

In this section, we present 5 characteristic function formulations $\nu(C)$, each designed for a different real-world application. While each formulation captures a different *flavor of centrality*, they all embrace one fundamental centrality idea: that the definition for $\nu(C)$ must somehow quantify the *sphere of influence* of the coalition C over the other nodes. For instance, in our first game formulation, we start with the simplest possible idea that the sphere of influence of a coalition of nodes C is the set of all nodes immediately reachable (within one hop) from C. Subsequent games further generalize this notion of sphere of influence. For example, the second formulation specifies a more sophisticated sphere of influence: one that includes only those nodes which are immediately reachable in at least k different ways from C. The other three

formulations extend the notion of sphere of influence to weighted graphs. The third game, for instance, defines sphere of influence as the set of all nodes within a cutoff distance of C (as measured by shortest path lengths on the weighted graph). The fourth formulation is an extreme generalization: it allows the sphere of influence of C to be specified by an arbitrary function $f(.)$ of the distance between C and the other nodes. The fifth and final formulation is a straightforward extension of the second, to the case of weighted networks.

Throughout this section, we assume the reader is familiar with concepts of graph theory, including weighted and unweighted graphs, vertex degrees, neighboring vertices and shortest paths. We do not define these concepts here but suggest the references [11, 12]. The terms "network" and "graph" are used interchangeably in this paper, as are the terms "node" and "vertex". All the weighted graphs considered in this paper are positive weighted. We do not use digraphs in this paper, so all graphs are assumed to be undirected.

We also assume familiarity with the concepts of co-operative game theory, including the definition of coalitional games in characteristic function form and the Shapley Value. We do not define these concepts here but suggest the references [13, 14].

We now set the notation for a general coalitional game played on a network. Given a graph $G(V, E)$ with vertex set V and edge set E, we use G to define a coalitional game $g(V(G), \nu)$ with set of agents $V(G)$ and characteristic function ν. Here the agents of the coalitional game are the vertices of the graph G. Thus a coalition of agents C is simply any subset of $V(G)$. The characteristic function $\nu : 2^{V(G)} \rightarrow \mathbb{R}$ can be any function that depends on the graph G as long as it satisfies the condition $\nu(\emptyset) = 0$. We use the phrase "value of coalition C" to informally refer to $\nu(C)$.

With the above notation, we now proceed to formally define and solve the 5 centrality-related coalitional games mentioned above.

3.1 Game 1: $\nu_1(C)$ = #agents at-most 1 degree away

Given an unweighted, undirected network $G(V, E)$. We first define "fringe" of a subset $C \subseteq V(G)$ as the set $\{v \in V(G) : v \in C$ (or) $\exists u \in C$ such that $(u, v) \in E(G)\}$, i.e, the fringe of a coalition includes all nodes reachable from the coalition in at most one hop.

Based on the fringe, we define the coalitional game $g_1(V(G), \nu_1)$ with respect to the network $G(V, E)$ by the characteristic function $\nu_1 : 2^{V(G)} \rightarrow \mathbb{R}$ given by

$$\nu_1(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ \text{size(fringe}(C)) & \text{else} \end{cases}$$

This coalitional game has been extensively discussed in [1], where the authors motivate the game by arguing that the SVs of nodes in this game constitute a centrality metric that is superior to degree centrality for some applications. It is therefore desired to compute the SVs of all nodes for this game. We shall now present an exact formula for this computation rather than obtaining it through Monte-Carlo simulation as was done in [1].

To evaluate the SV of node v_i, consider all possible permutations of the nodes in which v_i would make a positive marginal contribution to the coalition of nodes occurring before itself. Let the set of nodes occurring before node v_i in a random permutation of nodes be denoted C_i. Let the neighbors of node v_i in the graph $G(V, E)$ be denoted $N_G(v_i)$ and the degree of node v_i be denoted $deg_G(v_i)$.

The key question to ask is: what is the necessary and sufficient condition for node v_i to marginally contribute node $v_j \in N_G(v_i) \cup \{v_i\}$ to fringe(C_i)? Clearly this happens if and only if neither v_j nor any of *its* neighbors are present in C_i. Formally $(N_G(v_j) \cup \{v_j\}) \cap C_i = \emptyset$.

Given that permutations are chosen uniformly at random for computing the SV, combinatorial arguments can be used to show that the above condition is satisfied with probability $\frac{1}{1+deg_G(v_j)}$. Denote by B_{v_i,v_j} the Bernoulli random variable that v_i marginally contributes v_j to fringe(C_i). Thus:

$$E[B_{v_i,v_j}] = \Pr[(N_G(v_j) \cup \{v_j\}) \cap C_i = \emptyset] = \frac{1}{1 + deg_G(v_j)}$$

Therefore, the Shapley Value $SV_{g_1}(v_i)$, which is the expected marginal contribution of v_i, is given by:

$$SV_{g_1}(v_i) \;=\; \sum_{v_j \in \{v_i\} \cup N_G(v_i)} E[B_{v_i,v_j}] \;=\; \sum_{v_j \in \{v_i\} \cup N_G(v_i)} \frac{1}{1 + deg_G(v_j)}$$

which is an *exact closed-form expression* for computing the SV of each node on the network.

Algorithm 1 describes an $O(V+E)$ procedure that directly implements the above equation to compute the exact SVs of all nodes in the network. By contrast, Monte-Carlo simulation requires $O(V + E)$ operations for *every* iteration. Moreover, the results obtained using Monte-Carlo are statistical in nature and may not be sufficiently accurate unless a large number of iterations are carried out.

Algorithm 1: Computing SVs for Game 1

Input: Unweighted graph $G(V, E)$
Output: SVs of all nodes in $V(G)$ for game g_1

foreach $v \in V(G)$ **do**
  ShapleyValue[v] = $\frac{1}{1+deg_G(v)}$;
  foreach $u \in N_G(v)$ **do**
    ShapleyValue[v] += $\frac{1}{1+deg_G(u)}$;
  end
end
return ShapleyValue;

It is possible to derive some intuition from the above formula. If a node has a high degree, the number of terms in its SV summation above will also be high. But the terms themselves would be inversely related to the degree of neighboring nodes. This gives the intuition that a node will have high centrality not only when its degree is high, but also whenever its degree tends to be higher in comparison to the degree of its neighboring nodes. In other words, *power comes from being connected to those who are powerless*, a fact that is well-recognized [15] by the centrality literature.

3.2 Game 2: $\nu_2(C)$ = #agents with at-least k neighbors in C

We now consider a more general game formulation for an unweighted graph $G(V, E)$, where the value of a coalition includes the number of agents who are either in the coalition or are adjacent to at least k agents who are in the coalition. Formally, we consider game g_2 characterised by $\nu_2 : 2^{V(G)} \to \mathbb{R}$, where

$$\nu_2(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ |\{v : v \in C \text{ (or) } |N_G(v) \cap C| \geq k\}| & \text{else} \end{cases}$$

Note that this game reduces to game g_1 for $k = 1$.

The motivation for this generalization is that in many real-life networks, the value of a coalition is interpreted as the number of agents who can be "influenced" by the coalition. For instance, in a viral marketing or innovation diffusion analysis [16], it is usually assumed that an agent will "be influenced" only if atleast k of his neighbors have already been convinced, which suggests such a game formulation.

Adopting notation from the previous subsection, we again ask: what is the necessary and sufficient condition for node v_i to marginally contribute node $v_j \in N_G(v_i) \cup \{v_i\}$ to the value of the coalition C_i?

Clearly, if $deg_G(v_j) < k$, we have $E[B_{v_i,v_j}] = \delta(v_i, v_j)$, i.e, $E[B_{v_i,v_j}] = 1$ for $v_i = v_j$ and 0 otherwise.

For $deg_G(n_j) \geq k$, we split the argument into two cases. If $v_j \neq v_i$, the condition for marginal contribution is that exactly $(k-1)$ neighbors of v_j already belong to C_i and $v_j \notin C_i$. On the other hand, if $v_j = v_i$, marginal contribution happens if and only if C_i originally consisted of at most $(k-1)$ neighbors of v_j.

So for $deg_G(v_j) \geq k$ and $v_j \neq v_i$, we have

$$E[B_{n_i,n_j}] = \binom{deg_G(v_j) - 1}{k-1} \frac{(k-1)!(1 + deg_G(v_j) - k)!}{(1 + deg_G(v_j))!} = \frac{1 + deg_G(v_j) - k}{deg_G(v_j)(1 + deg_G(v_j))}$$

And for $deg_G(v_i) \geq k$ and $v_j = v_i$, we have

$$E[B_{v_i,v_i}] = \frac{k}{1 + deg_G(v_i)}$$

As before, the SVs are given by substituting the above formulas into:

$$SV_{g_2}(v_i) = \sum_{v_j \in N_G(v_i) \cup \{v_i\}} E[B_{v_i,v_j}]$$

Although this game is a generalization of game g_1, it can still be solved to obtain the SVs of all nodes in $O(V + E)$ time, as formalised by Algorithm 2.

Algorithm 2: Computing SVs for Game 2

Input: Unweighted graph $G(V, E)$, positive integer k
Output: SVs of all nodes in $V(G)$ for game g_2

foreach $v \in V(G)$ **do**
  ShapleyValue[v] = min(1, $\frac{k}{1+deg_G(v)}$);
  foreach $u \in N_G(v)$ **do**
    ShapleyValue[v] += max(0, $\frac{deg_G(u)-k+1}{deg_G(u)(1+deg_G(u))}$);
  end
end
return ShapleyValue;

An even more general formulation of the game is possible by allowing k to be a function of the agent, i.e, each node $v_i \in V(G)$ is assigned its own unique attribute $k(v_i)$. This translates to an application of the form: agent i is convinced if and only if atleast k_i of his neighbors are convinced, which is a frequently used model in the literature [16].

The above proof does not use the fact that k is constant across all nodes. So this generalized formulation can be solved by a simple modification to the original SV expression:

$$SV(v_i) = \frac{k(v_i)}{1 + deg_G(v_i)} + \sum_{v_j \in N_G(v_i)} \frac{1 + deg_G(v_j) - k(v_j)}{deg_G(v_j)(1 + deg_G(v_j))}$$

The above equation (which is also implementable in $O(V + E)$ time) assumes that $k(v_i) \leq 1 + deg_G(v_i)$ for all nodes v_i. This condition can be assumed without loss of generality because all cases can still be modeled (we set $k(v_i) = 1 + deg_G(v_i)$ for the extreme case where node v_i is never convinced no matter how many of its neighbors are already convinced).

3.3 Game 3: $\nu_3(C) = \#$agents at-most d_{cutoff} away

Hitherto, our games have been confined to unweighted networks. But in many applications, it is necessary to model real world networks as weighted graphs. For example, in a co-authorship network, each edge is often assigned a weight proportional to the number of joint publications the corresponding authors have produced [17].

This subsection extends the game g_1 to the case of weighted networks. Whereas game g_1 equates $\nu(C)$ to the number of nodes located within one hop of some node in C, our new formulation in this subsection equates $\nu(C)$ to the number of nodes located within a distance d_{cutoff} of some node in C. Here, distance between two nodes is measured as the length of the shortest path between the nodes in the given weighted graph $G(V, E, W)$, where $W : E \to \mathbb{R}^+$ is the weight function.

Formally, we define the game g_3, where for each coalition $C \subseteq V(G)$,

$$\nu_3(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ \text{size}(\{v_i : \exists v_j \in C \mid \text{distance}(v_i, v_j) \leq d_{\text{cutoff}}\}) & \text{else} \end{cases}$$

We shall now show that even this highly general centrality game g_3 is amenable to analysis which yields an exact formula for SVs. However, in this case the algorithm for implementing the formula is not linear in the size of the network, but has $O(VE + V^2\log(V))$ complexity.

Before deriving the exact SV formula, we introduce some extra notation. Define the *extended neighborhood* $N_G(v_j, d_{\text{cutoff}}) = \{v_k \neq v_j : \text{distance}(v_k, v_j) \leq d_{\text{cutoff}}\}$, i.e, the set of all nodes whose distance from v_j is at most d_{cutoff}. Denote the size of $N_G(v_j, d_{\text{cutoff}})$ by $deg_G(v_j, d_{\text{cutoff}})$.

With this notation, the necessary and sufficient condition for node v_i to marginally contribute node v_j to the value of coalition C_i is: $\text{distance}(v_i, v_j) \leq d_{\text{cutoff}}$ and $\text{distance}(v_j, v_k) > d_{\text{cutoff}}\ \forall v_k \in C_i$. That is, neither v_j nor any node in *its* extended neighborhood should be present in C_i. But from the discussion of previous subsections, we know that the probability of this event is exactly $\frac{1}{1+deg_G(v_j, d_{\text{cutoff}})}$. Therefore, the exact formula for SV of node v_i in game g_3 is:

$$SV_{g_3}(v_i) = \sum_{v_j \in \{v_i\} \cup N_G(v_i, d_{\text{cutoff}})} \frac{1}{1 + deg_G(v_j, d_{\text{cutoff}})}$$

Algorithm 3: Computing SVs for Game 3

Input: Weighted graph $G(V, E, W)$, $d_{\text{cutoff}} > 0$
Output: SVs of all nodes in G for game g_3

```
foreach v ∈ V(G) do
    DistanceVector D = Dijkstra(v,G);
    extNeighbors(v) = ∅; extDegree(v) = 0;
    foreach u ∈ V(G) such that u ≠ v do
        if D(u) ≤ d_cutoff then
            extNeighbors(v).push(u);
            extDegree(v)++;
        end
    end
end
foreach v ∈ V(G) do
    ShapleyValue[v] = 1/(1+extDegree(v));
    foreach u ∈ extNeighbors(v) do
        ShapleyValue[v] += 1/(1+extDegree(u));
    end
end
return ShapleyValue;
```

Algorithm 3 works as follows: for each node v in the network $G(V, E)$, the extended neighborgood $N_G(v, d_{\text{cutoff}})$ and its size $deg_G(v, d_{\text{cutoff}})$ are first computed using Dijkstra's algorithm in $O(E + V\log(V))$ time [18]. The results are then used to directly implement the above equation, which takes maximum time $O(V^2)$. In practice this step runs much faster because the worst case situation only occurs when every node is reachable from every other node within d_{cutoff}. Overall the complexity is $O(VE + V^2\log(V))$.

We make one final observation: that the above proof does not depend on d_{cutoff} being constant across all nodes. Indeed, each node $v_i \in V(G)$ may be assigned its own unique value $d_{\text{cutoff}}(v_i)$, where $\nu(C)$ would be the number of agents v_i who are within a distance $d_{\text{cutoff}}(v_i)$ from C. For this case, the above proof gives:

$$SV(v_i) = \sum_{\substack{v_j : \text{distance}(v_i, v_j) \\ \leq d_{\text{cutoff}}(v_j)}} \frac{1}{1 + deg_G(v_j, d_{\text{cutoff}}(v_j))}$$

3.4 Game 4: $\nu_4(C) = \sum_{v_i \in V(G)} f(\text{distance}(v_i, C))$

This subsection further generalizes game g_3, again taking motivation from real-life network problems. In game g_3, all agents at distances $d_{\text{agent}} \leq d_{\text{cutoff}}$ contributed equally to the value of a coalition. However, this assumption may not always hold true because in some applications, we intuitively expect agents closer to a coalition to contribute more to its value. For instance, we expect a Facebook user to exert more influence over his immediate circle of friends than over "friends of friends", even though both may satisfy the d_{cutoff} criterion. Similarly, we expect a virus-affected computer to infect a neighboring computer more quickly than a computer two hops away.

In general, we expect that an agent at distance d from a coalition would contribute $f(d)$ to its value, where $f(.)$ is a positive valued decreasing function of its argument. More formally, we define the game g_4 where the value of a coalition C is given by:

$$\nu_4(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ \sum_{v_i \in V(G)} f(d(v_i, C)) & \text{else} \end{cases}$$

where $d(v_i, C)$ is the minimum distance $\min\{\text{distance}(v_i, v_j) | v_j \in C\}$.

The key question to ask is: what is the expected value of the marginal contribution of v_i through node $v_j \neq v_i$ to the value of coalition C_i? Let this marginal contribution be denoted $MC(v_i, v_j)$. Clearly:

$$MC(v_i, v_j) = \begin{cases} 0 & \text{if distance}(v_i, v_j) \geq d(v_j, C_i) \\ f(\text{distance}(v_i, v_j)) - f(d(v_j, C_i)) & \text{else} \end{cases}$$

Let $D_{v_j} = \{d_1, d_2 ... d_{|V|-1}\}$ be the distances of node v_j from all other nodes in the network, sorted in increasing order. Let the nodes corresponding to these distances be $\{w_1, w_2 ... w_{|V|-1}\}$ respectively. Let $k_{ij} + 1$ be the number of nodes (out of these $|V| - 1$) whose distances to v_j are $\leq \text{distance}(v_i, v_j)$. Let $w_{k_{ij}+1} = v_i$ (i.e, among all nodes that have the same distance from v_j as v_i, v_i is placed last in the increasing order).

We use *literal* w_i to mean $w_i \in C_i$ and the literal $\overline{w_i}$ to mean $w_i \notin C_i$. Define a sequence of boolean variables $p_k = \overline{v_j} \wedge \overline{w_1} \wedge \overline{w_2} \wedge ... \wedge \overline{w_k}$ for each $0 \leq k \leq |V| - 1$. Finally denote expressions of the form $MC(v_i, v_j|F)$ to mean the marginal contribution of v_i to C_i through v_j given that the coalition C_i satisfies the boolean expression F.

$$\begin{aligned} MC(v_i, v_j | p_{k_{ij}+1} \wedge w_{k_{ij}+2}) &= f(d_{k_{ij}+1}) - f(d_{k_{ij}+2}) \\ MC(v_i, v_j | p_{k_{ij}+2} \wedge w_{k_{ij}+3}) &= f(d_{k_{ij}+1}) - f(d_{k_{ij}+3}) \\ \vdots \qquad & \vdots \qquad \vdots \\ MC(v_i, v_j | p_{|V|-2} \wedge w_{|V|-1}) &= f(d_{k_{ij}+1}) - f(d_{|V|-1}) \\ MC(v_i, v_j | p_{|V|-1}) &= f(d_{k_{ij}+1}) \end{aligned}$$

With this notation, we obtain expressions for $MC(v_i, v_j)$ by splitting over the above *mutually exclusive* and *exhaustive* (i.e, covering all possible non-zero marginal contributions) cases.

The probabilities $\Pr(p_k \wedge w_{k+1})$ are found by elementary combinatorics which gives:

$$\Pr(p_k \wedge w_{k+1}) = \frac{k!}{(k+2)!} = \frac{1}{(k+1)(k+2)} \;\forall\; 1 + k_{ij} \leq k \leq |V| - 2$$

Using the $MC(v_i, v_j)$ equations and the probabilities $\Pr(p_k \wedge w_{k+1})$:

$$\begin{aligned} E[MC(v_i, v_j)] &= \left[\sum_{k=1+k_{ij}}^{|V|-2} \frac{f(\text{distance}(v_i, v_j)) - f(d_{k+1})}{(k+1)(k+2)} \right] + \frac{f(\text{distance}(v_i, v_j))}{|V|} \\ &= \frac{f(\text{distance}(v_i, v_j))}{k_{ij} + 2} - \sum_{k=k_{ij}+1}^{|V|-2} \frac{f(d_{k+1})}{(k+1)(k+2)} \end{aligned}$$

Algorithm 4: Computing SVs for Game 4

Input: Weighted graph $G(V, E, W)$, function $f : \mathbb{R}^+ \to \mathbb{R}^+$
Output: SVs of all nodes in G for game g_4

Initialise: $\forall v \in V(G)$ **set** ShapleyValue[v] = 0;
foreach $v \in V(G)$ **do**
  [Distances D, Nodes w] = Dijkstra(v,G);
  sum = 0; index = |V|-1; prevDistance = -1, prevSV = -1;
  while *index > 0* **do**
    if *D(index) == prevDistance* **then**
      currSV = prevSV;
    else
      currSV = $\frac{f(D(index))}{1+index} - sum$;
    end
    ShapleyValue[w(index)] += currSV;
    sum += $\frac{f(D(index))}{index(1+index)}$;
    prevDistance = D(index), prevSV = currSV;
    index--;
  end
  ShapleyValue[v] += f(0) − sum;
end
return ShapleyValue;

For $v_i = v_j$, a similar analysis produces:

$$E[MC(v_i, v_i)] = f(0) - \sum_{k=0}^{|V|-2} \frac{f(d_{k+1})}{(k+1)(k+2)}$$

Finally the exact SVs are given by:

$$SV_{g_4}(v_i) = \sum_{v_j \in V(G)} E[MC(v_i, v_j)]$$

Algorithm 4 implements the above formulas. For each vertex v, a vector of distances to every other vertex is first computed using Dijkstra's algorithm [18]. This yields a vector D_v that is already sorted in increasing order. This vector is then traversed in reverse, to compute the backwards cumulative sum $\sum \frac{f(d_{k+1})}{(k+1)(k+2)}$. At each step of the backward traversal, the SV of the appropriate node w is updated according to the $E[MC(w, v)]$ equation. After the traversal, the SV of v itself is updated according to the $E[MC(v, v)]$ equation. This process is repeated for all nodes v so that at the end of the algorithm, all SVs have been computed exactly in $O(VE + V^2 \log(V))$ time.

3.5 Game 5: $\nu_5(C)$ = #agents with $\sum$(weights inside C) $\geq W_{\text{cutoff}}$(agent)

In this subsection, we generalize the game g_2 for the case of weighted networks. Given a positive weighted network $G(V, E, W)$ and a value $W_{\text{cutoff}}(v_i)$ for every node $v_i \in V(G)$, we first define $W(v_j, C) = \sum_{v_i \in C} W(v_j, v_i)$ for every coalition C, where $W(v_i, v_j)$ is the weight of the edge between nodes v_i and v_j (or 0 if there is no such edge). With this notation, we define the game g_5 by the characteristic function:

$$\nu_5(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ \text{size}(\{v_i : v_i \in C \ (or) \ W(v_i, C) \geq W_{\text{cutoff}}(v_i)\}) & \text{else} \end{cases}$$

The formulation above has applications in the analysis of information diffusion, adoption of innovations etc. Indeed, many cascade models of such phenomena on weighted graphs have been proposed [19] [20] [21], which work by assuming that an agent will change state from "inactive" to "active" if and only if the sum of the weights to all active neighbors is at least equal to an agent-specific cutoff.

Although we have not been able to come up with an exact formula for SVs in this game, our analysis yields an approximate formula which was found to be quite accurate in practice.

We observe that node v_i marginally contributes node $v_j \in N_G(v_i)$ to the value of coalition C_i if and only if $v_j \notin C_i$ and $W_{\text{cutoff}}(v_j) - W(v_i, v_j) \leq W(v_j, C_i) < W_{\text{cutoff}}(v_j)$. Denote by B_{v_i,v_j} the Bernoulli random variable corresponding to this event.

Let $N_G(v_j) = \{v_i, w_1, w_2 ... w_{deg_G(v_j)-1}\}$. Let the weights of edges between v_j and each of the nodes in $N_G(v_j)$ be $W_j = \{W(v_i, v_j), W_1, W_2 ... W_{deg_G(v_j)-1}\}$ in that order. Also, let α_j be the sum of all the weights in W_j and β_j be the sum of the squares of all the weights in W_j.

Let k_{ij} be the number of nodes of $N_G(v_j)$ that occur before v_i in C_i. Let X_t^{ij} be the sum of a t-subset of $W_j \setminus \{W(v_i, v_j)\}$ drawn uniformly at random from the set of all such possible t-subsets. Let Y_m^{ij} be the event $\{k_{ij} = m \ \wedge \ v_j \notin C_i\}$. Then:

$$E[B_{v_i,v_j}] = \sum_{m=0}^{deg_G(v_j)-1} \Pr(Y_m^{ij}) \ \Pr\{X_m^{ij} \in [W_{\text{cutoff}}(v_j) - W(v_i, v_j), W_{\text{cutoff}}(v_j))\}$$

where $\Pr(Y_m^{ij})$ can be analytically evaluated to be

$$\Pr(Y_m^{ij}) = \binom{deg_G(v_j) - 1}{m} \frac{m! \ (deg_G(v_j) - m)!}{(deg_G(v_j) + 1)!} = \frac{deg_G(v_j) - m}{deg_G(v_j)(deg_G(v_j) + 1)}$$

Evaluating $\Pr\{X_m^{ij} \in [W_{\text{cutoff}}(v_j) - W(v_i, v_j), W_{\text{cutoff}}(v_j))\}$ is much more difficult because the distribution of X_m^{ij} is a complicated function of the $deg_G(v_j) - 1$ numbers in $W_j \setminus \{W(v_i, v_j)\}$. However, we can obtain analytical expressions for the mean $\mu(X_m^{ij})$ and variance $\sigma^2(X_m^{ij})$. These are given by:

$$\mu(X_m^{ij}) = \frac{m}{deg_G(v_j) - 1} \ (\alpha_j - W(v_i, v_j))$$

$$\sigma^2(X_m^{ij}) = \frac{m(deg_G(v_j) - 1 - m)}{(deg_G(v_j) - 1)(deg_G(v_j) - 2)} \ (\beta_j - W(v_i, v_j)^2 - \frac{(\alpha_j - W(v_i, v_j))^2}{deg_G(v_j) - 1})$$

Knowing only the mean and variance (not the exact distribution) of X_m^{ij}, we propose the approximation:

$$X_m^{ij} \sim \mathcal{N}(\mu(X_m^{ij}), \sigma^2(X_m^{ij}))$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the Gaussian random variable with mean μ and variance σ^2. This approximation is similar to the randomised approach that has been proposed and tested in [22].

With this approximation, we have $Z_m^{ij} = \Pr\{X_m^{ij} \in [W_{\text{cutoff}}(v_j) - W(v_i, v_j), W_{\text{cutoff}}(v_j))\}$ given by

$$Z_m^{ij} \approx \frac{1}{2} \left[\text{erf}\left(\frac{W_{\text{cutoff}}(v_j) - \mu(X_m^{ij})}{\sqrt{2}\sigma(X_m^{ij})} \right) - \text{erf}\left(\frac{W_{\text{cutoff}}(v_j) - W(v_i, v_j) - \mu(X_m^{ij})}{\sqrt{2}\sigma(X_m^{ij})} \right) \right]$$

This allows us to write: $E[B_{v_i,v_j}] = \sum_{m=0}^{deg_G(v_j)-1} \frac{deg_G(v_j) - m}{deg_G(v_j)(deg_G(v_j) + 1)} \ Z_m^{ij}$

The above equations are true only for $v_j \neq v_i$. For $v_j = v_i$ we have:

$$E[B_{v_i,v_i}] \approx \frac{1}{1 + deg_G(v_i)} \sum_{m=0}^{deg_G(v_i)} \Pr\{\mathcal{N}(\mu(X_m^{ii}), \sigma^2(X_m^{ii})) < W_{\text{cutoff}}(v_i)\}$$

$$\text{where} \quad \mu(X_m^{ii}) = \frac{m}{deg_G(v_i)} \alpha_i \quad \text{and} \quad \sigma^2(X_m^{ii}) = \frac{m \ (deg_G(v_i) - m)}{deg_G(v_i) \ (deg_G(v_i) - 1)} \ (\beta_i - \frac{\alpha_i^2}{deg_G(v_i)})$$

Finally the SV of node v_i is given by $\sum_{v_j \in \{v_i\} \cup N_G(v_i)} E[B_{v_i,v_j}]$.

Algorithm 5 implements an $O(V + \sum_{v_i \in V(G)} deg_G^2(v_i)) \leq O(V + E^2)$ algorithm to compute SVs of all agents in game g_5 using the above approximation.

We make one final observation: that the approximation of the discrete random variable X_m^{ij} as a continuous Gaussian random variable is good only when $deg_G(v_j)$ is large. For small $deg_G(v_j)$, one might as well use the brute force computation to determine $E[B_{v_i,v_j}]$ in $O(2^{deg_G(v_j)-1})$ time.

4 Related Work

Three related bodies of literature can be distinguished, (a) game theory literature on the SV for graph-based games, (b) network analysis literature on centrality measures, and (c) computer science literature on concise representation of coalitional games and the complexity of SV computation.

Game theory literature: Myerson [23] introduced graph-based games in which each allowable (or feasible) coalition is induced by a subgraph of a graph, and for these games he redefined the concept of SV. In a number of follow-up works, the SV was studied for games in which the set of feasible coalitions was based on mathematical structures such as distributive lattices [24], convex geometries [25], antimatroids [26] and augmenting systems [27].

Algorithm 5: Computing SVs for Game 5

Input:
- ◇ Weighted network $G(V, E, W)$
- ◇ Cutoffs $W_{\text{cutoff}}(v_i)$ for each $v_i \in V(G)$

Output: SVs of all nodes in G for game g_6

```
foreach v_i ∈ V(G) do
    compute and store α_i and β_i;
end
foreach v_i ∈ V(G) do
    ShapleyValue[v_i] = 0;
    foreach m in 0 to deg_G(v_i) do
        compute μ = μ(X_m^ii), σ = σ(X_m^ii);
        compute p = Pr{N(μ, σ^2) < W_cutoff(v_i)};
        ShapleyValue[v_i] += p / (1+deg_G(v_i));
    end
    foreach v_j ∈ N_G(v_i) do
        p = 0;
        foreach m in 0 to deg_G(v_j) − 1 do
            compute μ = μ(X_m^ij), σ = σ(X_m^ij);
            compute z = Z_m^ij;
            p += z (deg_G(v_j) − m) / (deg_G(v_j)(deg_G(v_j)+1));
        end
        ShapleyValue[v_i] += p;
    end
end
return ShapleyValue;
```

Network analysis literature: Freeman [28] formalised the notion of centrality in network analysis, by presenting three different conventional centrality measures: degree, closeness and betweenness. Many authors have since worked on (a) developing new centrality measures for specific applications [29–31], and (b) developing algorithms for efficient centrality computation [32, 33].

Gòmez et al. [2] combined Myerson's idea of graph-restricted games with the concept of centrality from network analysis to propose new SV-based network centrality measures. Suri and Narahari [1] proposed a new SV-based centrality measure which was fundamentally different from the existing literature because it permitted all coalitions to be feasible. They have also explored the application of such centrality measures to determine the most important people in a social network [10].

Computer science literature: While the game theory literature focused on mathematical properties of games, the computer science literature was primarily concerned with algorithmic and computational aspects. Hence a major emphasis was placed on developing concise representations for coalitional games.

Such representations fall into two categories [34]:

1. Those which give the characteristic function a specific interpretation in terms of combinatorial structures such as graphs. This is, for instance, the approach adopted in [3, 34, 35] and its advantage is that the resulting representation is always guaranteed to be succinct. However, the disadvantage is that it is not always fully expressive.

2. Those which try to find a succinct but still fully expressive representation (see, for instance, [6, 36, 37]). These representations are more general (in that they completely capture all coalitional games of interest), although they are not always guaranteed to be succinct.

The computational complexity of the SV for succinctly representable games has also been considered by the computer science literature. Unfortunately, SV computation has been proved NP-Hard (or even worse, #P-Complete) for many domains, including weighted voting games [3], threshold network flow games [5] and minimum spanning tree games [4].

Fortunately, some positive results have also been discovered. The most prominent among these is due to Ieong and Shoham [6], who developed a representation consisting of a finite set of logical rules of the form *Boolean Expression* → *Real Number*, with agents as the atomic boolean variables. In this representation, the value of a coalition is equal to the sum of the right sides of those rules whose left sides are satisfied by the coalition. The rules have an interesting game-theoretic interpretation, as each rule directly specifies an incremental marginal contribution made by agents featured in that rule. This representation, called *marginal contribution networks* (MC-Nets) is (i) fully expressive (i.e, it can be used to model any game),

(ii) exponentially more concise for some games, and most importantly, (iii) allows for computing the SV in time linear in the size of the representation, provided the boolean expressions in all rules are conjunctions of (either positive or negative) atomic literals. Elkind et al. [37] developed extensions to more sophisticated (read-once) boolean expressions while Michalak et al. [38, 39] developed generalizations to coalitional games with externalities.

While MC-Nets offer a fully-expressive representation that works for arbitrary coalitional games, the question of whether SV computation can be speeded up (to polynomial time) by restriction to specific (not necessarily fully expressive) classes of games or real-life applications has received comparatively little attention. The one specific class of games/applications that has been investigated in detail is weighted voting, for which both approximate (but strictly polynomial) [22] and exact (but pseudo-polynomial) algorithms [40, 41] have been proposed. Aside from MC-Nets (and extensions thereof) and weighted voting games, the only other positive results known for SV computation are the restricted cases (of more general game classes) that can be solved in polynomial time, to which many papers [5, 22] draw attention.

5 Summary and conclusions

Game	Graph	$\nu(C)$	Complexity
g_1	UW	≤ 1 degree away	$V+E$
g_2	UW	$\geq k$ neighbors $\in C$	$V+E$
g_3	W	$\leq d_{\text{cutoff}}$ away	$VE+V^2\log V$
g_4	W	$\sum_{v_i} f(d(v_i, C))$	$VE+V^2\log V$
g_5	W	$\lvert\{v_i : W(v_i, C) \geq W_{\text{cutoff}}(v_i)\}\rvert$	$V+E^2$

Second Column $\implies$ $\{W =$ weighted, $UW =$ unweighted$\}$

The table to the left presents a brief summary of the SV algorithms discussed in this paper. These algorithms enable efficient centrality computation for many real-world applications including the analysis of social networks, information diffusion, spread of epidemics, biological and biochemical networks, viral marketing and internet/web phenomena.

The conclusion is that many centrality-related co-operative games of interest played on real-life networks can in fact be solved for SVs analytically. The resulting algorithms are not only error-free but also run in polynomial time and in practice, much faster than Monte-Carlo methods. Approximate closed-form expressions and algorithms can also be constructed for some classes of games played on weighted networks. Simulation results (please see appendix) show that these approximations are quite acceptable.

References

[1] N.R. Suri and Y. Narahari. Determining the top-k nodes in social networks using the Shapley Value. In *AAMAS '08: Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1509–1512, 2008.

[2] D. Gómez, E. González-Arangüena, C. Manuel, G. Owen, M. Del Pozo, and J. Tejada. Centrality and power in social networks: A game theoretic approach. *Mathematical Social Sciences*, 46(1):27–54, 2003.

[3] X. Deng and C. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.

[4] J. Bilbao, J. Fernández, and J.J. López. Complexity in cooperative game theory. `http://www.esi2.us.es/~mbilbao/pdffiles/compgame.pdf`.

[5] Y. Bachrach and J.S. Rosenschein. Power in threshold network flow games. *Autonomous Agents and Multi-Agent Systems*, 18(1):106–132, 2009.

[6] S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *EC '05: Proceedings of the Sixth ACM Conference on Electronic Commerce*, pages 193–202, 2005.

[7] D. Koschützki, K.A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski. *Centrality indices. Network analysis*, volume 3418 of *Lecture Notes in Computer Science*, pages 16–61. Springer, 2005.

[8] D.J. Watts. *Six degrees: The science of a connected age.* W.W. Norton & Company, 2004.

[9] D. Kempe, J. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. *Automata, Languages and Programming*, 2005.

[10] N.R. Suri and Y. Narahari. A Shapley Value based approach to discover influential nodes in social networks. *To appear in IEEE Transactions on Automation Science and Engineering*, 2010. `http://clweb.csa.iisc.ernet.in/nrsuri/jsac-nrsuri-2008.pdf`.

[11] J.A. Bondy and U.S.R. Murty. *Graph theory and its applications.* Macmillan, 1976.

[12] D.B. West. *Introduction to graph theory.* Prentice Hall, Inc., 2001.

[13] R. Gibbons. *A primer in game theory.* Harvester Wheatsheaf, 1994.

[14] A. Rapoport. *N-person game theory: Concepts and applications.* Dover Publications, 2001.

[15] P. Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[16] T.W. Valente. Social network thresholds in the diffusion of innovations. *Social Networks*, 18(1):69–89, 1996.

[17] M.E.J. Newman. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132(1–7), 2001.

[18] T.H. Cormen. *Introduction to algorithms.* MIT Press, 2001.

[19] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[20] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.
[21] H.P. Young. *The diffusion of innovations in social networks. Economy as an evolving complex system III*, pages 267–281. Oxford University Press, 2005.
[22] S.S. Fatima, M. Wooldridge, and N.R. Jennings. A randomized method for the Shapley Value for the voting game. In *AAMAS '07: Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 959–966, 2007.
[23] R.B. Myerson. Graphs and cooperation in games. *Mathematics of Operations Research*, 2(3):225–229, 1077.
[24] U. Faigle. Cores of games with restricted cooperation. *Mathematical Methods of Operations Research*, 33(6):405–422, 1989.
[25] J.M. Bilbao and P.H. Edelman. The Shapley Value on convex geometries. *Discrete Applied Mathematics*, 103(1-3):33–40, 2000.
[26] E. Algaba, J.M. Bilbao, R. van den Brink, and A. Jiménez-Losada. Cooperative games on antimatroids. *Discrete Mathematics*, 282(1-3):1–15, 2004.
[27] J.M. Bilbao and M. Ordóñez. Axiomatizations of the Shapley Value for games on augmenting systems. *European Journal of Operational Research*, 196(3):1008–1014, 2009.
[28] L.C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1(3):215–239, 1979.
[29] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120, 1972.
[30] J.D. Noh and H. Rieger. Random walks on complex networks. *Physical Review Letters*, 92(11):118701(1–4), 2004.
[31] K. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1–37, 1989.
[32] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.
[33] D. Eppstein and J. Wang. Fast approximation of centrality. In *SODA '01: Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 228–229, 2001.
[34] M. Wooldridge and P. Dunne. On the computational complexity of coalitional resource games. *Artificial Intelligence*, 170(10):835–871, 2006.
[35] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of compact coalitional games. In *IJCAI '09: Proceedings of the Twenty First International Joint Conference on Artifical Intelligence*, pages 147–152, 2009.
[36] V. Conitzer and T. Sandholm. Computing Shapley Values, manipulating value division schemes and checking core membership in multi-issue domains. In *AAAI '04: Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 219–225, 2004.
[37] E. Elkind, L.A. Goldberg, P.W. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. *Mathematical Logic Quarterly*, 55(4):362–376, 2009.
[38] T.P. Michalak, D. Marciniak, M. Samotulski, T. Rahwan, P. McBurney, M. Wooldridge, and N. Jennings. A logic-based representation for coalitional games with externalities. In *AAMAS '10: Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 125–132, 2010.
[39] T. Michalak, T. Rahwan, D. Marciniak, M. Szamotulski, and N. Jennings. Computational aspects of extending the Shapley Value to coalitional games with externalities. In *ECAI '10: Proceedings of the Nineteenth European Conference on Artificial Intelligence (to appear)*, 2010. `http://eprints.ecs.soton.ac.uk/21218/1/ECAI%2D139.pdf`.
[40] I. Mann and L.S. Shapley. Values of large games, VI: Evaluating the electoral college exactly. RAND Research Memorandum, 1962.
[41] T. Matsui and Y. Matsui. A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan*, 43:71–86, 2000.
[42] D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

Appendix A: Simulation Results

This section presents results obtained by simulating games g_1 to g_5 on networks that have been well-studied in the literature.

For games $g_1 - g_3$ (which can be played on unweighted networks[4]), we present simulations on an undirected, unweighted network representing the topology of the Western States Power Grid. This network (which has 4940 nodes and 6594 edges) has been studied in many contexts before (for example see [42]) and has been made freely available on the web. For games $g_3 - g_5$ (played on weighted networks), we have used the network of astrophysics collaborations between Jan 1, 1995 and December 31, 1999. This network (which has 16705 nodes and 121251 edges) is also freely available on the web and has been used in previous studies like [17].

The results for games g_1 to g_4 were generated as follows.

1. First we decide which game to play on the network and the parameters required if any.
2. Then we use exact algorithms proposed in this paper to solve for SVs. The runtime for the exact algorithm is denoted t_{exact}.
3. We now run Monte-Carlo simulations of the game. After each Monte-Carlo run, we compare the Monte-Carlo SVs with the true SVs. If the error is more than a tolerance level, we increase the number of iterations and repeat until the error is within tolerance limits. We denote by *minIter* the minimum number of iterations required for Monte-Carlo to yield results within the required tolerance. The runtime for *minIter* iterations is denoted t_{MC}.
   Implementation note: The tolerance criterion we have used here is that for each node, the SV computed by Monte-Carlo should be within 10% of the exact SV computed in Step 2. The initial number of iterations was fixed at 100 for all simulations. All simulations were carried out in Java.
4. We present $\frac{t_{\text{MC}}}{t_{\text{exact}}}$ as a measure of the speedup offered by our proposed algorithm over Monte-Carlo.

Table 1. Simulation results for games g_1 to g_4

Game	Parameters	*minIter*	MC Error	t_{exact}	t_{MC}	Speedup
g_1	–	2843	9.87%	7.97 *ms*	2.72 *s*	341
g_2	$k = 2$	1625	8.55%	7.38 *ms*	1.6 *s*	217
g_2	$k_i = \frac{deg_i}{2}$	2233	8.75%	7.76 *ms*	2.14 *s*	276
g_2	$k_i = \frac{3}{4}\, deg_i$	1406	9.79%	7.89 *ms*	1.41 *s*	178
g_3	$k_{\text{cutoff}} = 2$	14950	7.56%	10.96 *s*	3.03 *min*	16.5
g_3	$k_{\text{cutoff}} = 3$	31027	8.57%	12.24 *s*	6.53 *min*	32
g_3	$d_{\text{cutoff}} = \frac{d_{avg}}{8}$	> 100000	11.35%	5.3 *min*	59.22 *min*	11
g_3	$d_{\text{cutoff}} = \frac{d_{avg}}{4}$	> 100000	15.28%	6.2 *min*	97.11 *min*	15.6
g_4	$f(d) = \frac{1}{1+d}$	12480	9.89%	44 *sec*	4.55 *hrs*	372
g_4	$f(d) = \frac{1}{1+d^2}$	15340	9.64%	45.1 *sec*	5.51 *hrs*	423
g_4	$f(d) = e^{-d}$	> 20000	16.64%	46.3 *sec*	7.12 *hrs*	553.6

d_{avg} is the average distance of a node from all other reachable nodes in the network

Table 1 shows that the algorithms proposed in this paper deliver much better performance than Monte-Carlo simulation for both networks. Moreover, even with a generous 10% margin for error, we find that *minIter* varies widely for different games. In some cases (like game g_3), *minIter* was not reached even after 100000 Monte-Carlo iterations.

Table 2. Game g_5 - Approximation errors for small networks

Graph	$W_{\text{cutoff}} = \frac{1}{4}\alpha$	$W_{\text{cutoff}} = \frac{1}{2}\alpha$	$W_{\text{cutoff}} = \frac{2}{3}\alpha$	$W_{\text{cutoff}} = \frac{3}{4}\alpha$	$W_{\text{cutoff}} = \frac{4}{5}\alpha$
K_5	1.13%	1.47%	0.91%	1.85%	2.46%
K_6	1.37%	1.82%	2.14%	2.42%	2.25%
K_7	2.35%	2.49%	3.17%	2.83%	2.61%
K_8	4.11%	1.82%	3.14%	3.50%	2.86%
K_9	4.19%	4.27%	2.18%	2.69%	4.42%
K_{10}	6.14%	2.48%	2.22%	1.91%	2.54%
K_{11}	2.47%	2.54%	1.03%	1.52%	0.95%
K_{12}	2.77%	1.62%	1.48%	1.53%	1.07%

For game g_5 we performed two levels of simulation: one for small networks and one for large networks.

1. For small networks, we first compute exact SVs using the exponential algorithm. We then use our approximate algorithm for computing SVs. Hence we determine the error in our approximations.

[4] Replacing the distance threshold d_{cutoff} with a hop-threshold k_{cutoff} enables the game g_3 to be played on an unweighted network

Implementation note: The small networks we have used are randomly weighted complete graphs from K_5 to K_{12}. The cutoff for each node $W_{\text{cutoff}}(v_i)$ is assigned as a fraction of $\alpha(v_i)$ (please see the analysis of game g_5 for definitions).

2. For large networks, we run a fixed number (here 10000) of Monte-Carlo iterations (runtime t_{MC}). We also separately run our proposed algorithm (runtime t_{approx}) and present the speedups obtained.

We see from Table 2 that the approximation error in our proposed algorithm is quite well-contained for small networks. In large networks the error may be higher but we still believe it will lie within practical tolerance bounds, especially if we adopt the mixed strategy that uses the approximation only for large degree vertices (as discussed in Section 3).

Table 3. Game g_5 - Speedups on large networks

Parameters	t_{approx}	t_{MC}	*Speedup*
$W_{\text{cutoff}} = \frac{1}{4}\alpha$	4 *mins*	2.2 *hours*	33
$W_{\text{cutoff}} = \frac{1}{2}\alpha$	4.2 *mins*	2.2 *hours*	31.4
$W_{\text{cutoff}} = \frac{2}{3}\alpha$	4.4 *mins*	2.2 *hours*	30
$W_{\text{cutoff}} = \frac{3}{4}\alpha$	5 *mins*	2.2 *hours*	26.4
$W_{\text{cutoff}} = \frac{4}{5}\alpha$	6.1 *mins*	2.2 *hours*	21.6

Table 3 shows that speedups obtained for the proposed algorithm are quite significant. Moreover, the earlier results suggest that the proposed algorithm is likely to be more accurate than the Monte-Carlo simulation.