Solving Influence Maximization Problem Using Methods from Cooperative Game Theory

Szymon Matejczyk[1] and Tomasz Michalak[2,3]

[1] Institute of Computer Science, Polish Academy of Sciences, ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
[2] Institute of Informatics, University of Warsaw, Poland
[3] Department of Computer Science, University of Oxford, UK

Abstract. The Shapley value—one of the fundamental game-theoretic solution concepts—has been recently proposed for the influence maximization problem, i.e. choosing the best seed nodes for the process of information diffusion [39, 40]. However, although applying game-theoretic solution concepts such as the Shapley value to tackle this problem seems to be a promising research direction, the performance of this approach has not yet been thoroughly evaluated due to inherent computational challenges. The contribution of this article is twofold. Firstly, we propose a new game-theoretic methods to tackle the problem of influence maximization: the Shapley value- and the Banzhaf index-based refinements of the Local DAG algorithm by Chen et al. [8]. Secondly, we provide the experimental evaluation of those new and previously known game-theoretic methods and compare its performance to other approaches from the literature. Our simulation results suggest that out method achieves competitive results when compared to the state-of-the-art heuristics.

1 Introduction

Diffusion in social networks is the process by which an idea, rumour, or disease is spread by people through links between them [26]. An interesting example of diffusion is the process of the public adopting a new technology. People recommend the innovation to their friends, and whether or not it eventually becomes a widely adopted technology depends on by how many people it has been adopted. Similarly, infectious diseases spread in the society between individuals that enter into physical contacts. The problem we are concerned with in this article is to find a set of initial nodes that will lead to the most widespread effect of diffusion in a network [29, 15, 42]. In other words, we look for a group of nodes (of a limited, *a priori* given size) that has the maximal influence on others. We will typically refer to nodes that have been successfully influenced by the diffusion process as *infected* or *active*.

The two most established mathematical models of diffusion are *Independent Cascade* and *Linear Threshold*. In short, in both models, as time passes in discrete time steps, new nodes are activated due to the influence of nodes that were active in the previous time step. The models differ in the way that the infection spreads through a particular link. In the first model, an active node has one chance to activate each of its neighbors with some pre-defined probability. In the second model, each node has some random pre-defined threshold and becomes active if it has enough active neighbors whose influence exceeds this threshold.

Unfortunately, the problem of finding the optimal set of initial seeds of limited size, called in the literature the *influence maximization problem* or the *top-k nodes* problem, is NP-hard in both models [29]. Given this negative result, recent research has focused on developing approximate solutions. In particular, Kempe et al. showed that the straightforward greedy algorithm that chooses a node that makes the largest contribution to the influence spread achieves an approximation ratio of $1 - 1/e$, where e is the base of the natural logarithm. Improvements to the greedy algorithm by Kempe et al. that focus on limiting the number of spread function computations were made by Leskovec et al. [34], Goyal et al. [23] resulting in *CELF++* algorithm that is current state-of-the-art.

The literature also contains a variety of heuristics that either modify spread function definition or focus on finding a solution that has no theoretical guarantees but performs well in practice. The former include PMIA [7] and SIMPATH [22] that simplify the spread to certain paths in the graph in Independent Cascade and Linear Threshold, respectively. Another interesting greedy algorithm of this type was proposed by Chen et al. [8] whose algorithm, in the search for the top k nodes, focuses on the influence spread only in local directed acyclic graphs (DAG). In simulations, this algorithm returns the solution that is of comparable quality to the original greedy algorithm that operates on the entire graph. Heuristics that focus on finding solution that has no theoretical guarantees but performs well in practice are usually built on various centrality measures that explore some properties of the network with the aim to estimate the importance of a node for information diffusion.[1] They include the standard centrality measures (such as degree, closeness, and betweenness centralities [19]), and also more novel heuristics designed especially for information diffusion (such as *Discount degree* heuristics [9]). The advantage of these heuristics is their time efficiency and scalability—they do not require the computation of the spread of diffusion in order to establish the seed set.

Both those latter algorithms are currently the state of the art. Their improved performance is achieved by limiting the calculations of the spread (only to a local neighbourhood of each node) and by decreasing the number of calls of the spread function.

More sophisticated heuristics to solve the influence maximization problem were proposed by Narayanam and Narahari [39, 40]. The authors developed

[1] Intuitively, in this approach, the seed is chosen following the ranking of nodes given by a chosen centrality measure. For instance, one can choose top k nodes with the highest degree centrality.

two algorithms based on the Shapley value-based centrality measures. These centralities belong to a wider class of advanced centrality measures that are based upon solution concepts from cooperative game theory. To construct such a centrality measure, we define a cooperative game over a network in which players are nodes, coalitions are subsets of the set of nodes, and payoffs of coalitions depend (in some way) on the network topology. Next, we use the Shapley value or other game-theoretic solution concepts that quantify the importance of a player in the game, as centrality measures that quantify the importance of nodes in the network. For instance, Narayanam and Narahari [39] construct their game-theoretic centrality as follows. Their payoff function assigns to each coalition the value that equals to the number of nodes this coalition is able to influence directly (i.e. the number of the coalition's neighbours). Then, the Shapley value is used to quantify the role of each node in influencing neighbours. Finally, all the nodes in the network are sorted according to their Shapley value and the first k of them are chosen as the approximate solution to the top-k nodes problem.

Unfortunately, unlike other heuristics for influence maximization, those based on the Shapley value-based centrality measures have not yet been extensively evaluated in the literature. In fact, the only experimental analysis that we are aware of is that of Narayanam and Narahari [39, 40]. These results, however, are very preliminary.

In this work, we reconsider the Shapley value-based approaches to the problem of influence maximization. We first propose a game-theoretic solution to the problem that is designed to fit parallel processing architectures and then we present a thorough evaluation of all the game-theoretic methods comparing them to other heuristics from the literature. In more detail, the contributions of this article can be summarized as follows:

- We show that the use of game-theoretic solution concepts for information diffusion is not restricted to the Shapley value and the cooperative games considered by Narayanam and Narahari [39, 40]. To this end, we combine both the Shapley value- and the Banzhaf index-based centrality measures with the Local DAG model [8]. Not only do our methods perform competitively (see the next bullet point), but we show that it is possible to parallelize the computation of both solution concepts in the Local DAG model. By doing so, we are able to fully utilize the computational advantages of local DAGs.
- In the simulation section, we compare all the above game-theoretic approaches to the aforementioned state-of-the-art heuristics, i.e. Greedy LDAG, CELF++ and *Discount Degree*. Importantly, in our simulations, whenever possible, we use the results from Aadithya et al. [1], who showed that some game-theoretic centrality measures, and the one from Narayanam and Narahari [39], in particular, can be computed in polynomial time. In other words, for game-theoretic centrality measures we consider exact solutions rather than Monte Carlo approximations.

Overall, the results of our experiments suggest that various approaches built upon cooperative game theory are competitive, even when compared to the

state-of-the art Greedy LDAG and CELF++ algorithms. In particular, our re-evaluation of the algorithm by Narayanam and Narahari [39] confirms that this approach outperforms the simple Degree Centrality heuristic. Moreover, also the Shapley value-based LDAG model proposed in this paper is competitive when compared to Greedy LDAG and CELF++. As for the comparison between the LDAG model based on the Shapley value with the one based on the Banzhaf index, the former turns out to achieve a more consistent performance than the latter.

The remainder of this article is organized as follows. In the next section, we formally introduce basic concepts from game theory and define influence models and the influence maximization problem. In Section 4 we describe game-theoretic solutions for the LDAG model by Chen et al. [8] and analyze their computational characteristic. Next, we present results of our experiments evaluating all the algorithms in Section 5. Conclusions follow.

2 Preliminaries

In this section we present the relevant concepts from cooperative game theory and graph theory, including the problem of influence maximization in networks. We also introduce notation used throughout the paper (comprehensive summary can be found in Appendix B). Finally, we discuss the basic idea behind the concept of game-theoretic centrality that connects cooperative game theory with graph theory.

2.1 Concepts from Cooperative Game Theory

Let $N = \{a_1, a_2, \ldots, a_{|N|}\}$ be the set of players that participate in a coalitional game. A *characteristic function* $\nu : 2^N \to \mathbb{R}$ assigns to every coalition $C \subseteq N$ a value (or a payoff) representing the quality of its performance, where it is assumed that $\nu(\emptyset) = 0$. The set of all players N is called the *grand coalition*. A coalitional game, also called the *characteristic function game*, is a pair (N, ν).

Let us now introduce the concepts of the Shapley value and the Banzhaf index. Both are intended to answer the following question: assuming that the grand coalition has formed, how should its value be distributed among the players?

The Shapley value: Shapley proposed to evaluate the role of each player in the game by considering marginal contributions that this player makes to all coalitions [43]. In more detail, the marginal contribution of a player a_i to coalition $C \subseteq N \setminus \{a_i\}$, denoted $\mu(a_i, C)$, is the difference between the value of C with and without a_i, i.e.:

$$\mu(a_i, C) = \nu(C \cup \{a_i\}) - \nu(C).$$

For example, when the player a_1 joins the empty coalition, the value of its marginal contribution, $\mu(a_1, \emptyset)$, is equal to $\nu(\{a_1\}) - \nu(\emptyset) = \nu(\{a_1\})$. Subsequently, when another player a_2 joins this newly created coalition $\{a_1\}$, its

marginal contribution, $\mu(a_2, \{a_1\})$, is equal to $\nu(\{a_1, a_2\}) - \nu(\{a_1\})$. The process continues until all the players have joined the coalition and the grand coalition emerges.

The order in which players build subsequent coalitions to finally form the grand coalition can be represented as a permutation of players. Let π denote such a permutation, $\pi(a_i)$ the position of player a_i in π, and $C_\pi(a_i)$ the coalition formed by all the predecessors of the player a_i in π. Formally, $C_\pi(a_i) = \{a_j : \pi(a_j) < \pi(a_i)\}$. For example, in the permutation $(a_5, a_3, a_7, \ldots, a_n, \ldots, a_2)$, $\pi(a_7) = 3$ and $C_\pi(a_7) = \{a_3, a_5\}$. Naturally, given n players, there are $n!$ permutations, i.e. ways to form the grand coalition. The Shapley value of player a_i, denoted $\mathrm{SV}(a_i)$, is the average marginal contribution of a_i to $C_\pi(a_i)$ over all permutations:

$$\mathrm{SV}(a_i) = \sum_{\pi \in \Pi(N)} \frac{1}{|N|!} (\nu(C_\pi(a_i) \cup \{a_i\}) - \nu(C_\pi(a_i))). \tag{1}$$

where $\Pi(N)$ is the set of all permutations of N.

The importance of the Shapley value stems from the fact that this division scheme is the only one that meets certain desirable criteria. We refer the reader to the overview by Colman [11] for more details on various axiomatizations of the Shapley value.

The Banzhaf index: The Banzhaf index [1965] is another solution concept for the problem of dividing the payoff of the grand coalition. It was initially proposed to measure the power of players in voting games, i.e., games in which each player has some given number of votes and a given quota of votes overall is needed to win. Such a scenario can be described as a simple coalitional game, where the characteristic function is binary—the coalition either has enough votes to win (value 1) or not (value 0). In simple coalitional games, the Banzhaf index of a player is equal to the number of all coalitions in which this player is indispensable to win divided by the number of all coalitions. This definition can be straightforwardly generalized to any characteristic function—here, the Banzhaf index is the expected marginal contribution made by the player a_i to a randomly chosen subset of $N \setminus \{a_i\}$. The formal definition of the Banzhaf index of player a_i is as follows:

$$\mathrm{BI}(a_i) = \frac{1}{2^{n-1}} \sum_{C \subseteq N \setminus \{a_i\}} \Big[\nu(C \cup \{a_i\}) - \nu(C)\Big]. \tag{2}$$

The difference between the Shapley value and the Banzhaf index can be intuitively described as follows: the Shapley value measures the player's expected marginal contribution if players form the grand coalition one by one in a random order, whereas the Banzhaf index measures the players expected marginal contribution if each player decides whether to take part in formation of the grand coalition independently with probability 1/2 [6].

More details on the Banzhaf index and its axiomatizations can be found in the works by Chalkiadakis et al. [6] and Lehrer [33].

2.2 Concepts from Graph Theory and the Influence Maximization Problem

A *graph* (or *network*) is a tuple $G = (V, E)$, where V denotes the set of *vertices* (or *nodes*) and $E \subset V \times V$ denotes the set of *edges*. Every edge from set E connects two different vertices in set V. By $(v, u) \in E$ we denote the directed edge connecting vertices from v to u, $v, u \in V$. In such a case, we call v an *in-neighbour* of u and to u as an *out-neighbour* of v. In this article, we consider weighted graphs, where additionally we have a function $w : E \to \mathbb{R}^+$ that assigns a weight to each edge. For simplicity we assume that $\forall_{v \in V} \sum_{u \in V \setminus \{v\}} w((u, v)) \leq 1$.

An important research direction in social network analysis is the study of how individuals influence each other and how this influence leads to a diffusion of a given phenomenon (such as information, innovation, or disease) throughout the network [21]. In this context, a social network is often called an *influence network*. In what follows, we introduce definitions of two most prominent models of influence in the literature: the *Independent Cascade* (IC) Model [20] and the *Linear Threshold* (LT) Model [29].

The influence network is typically modelled with a weighted, directed graph $G = (V, E, w)$, where an edge $e = (u, v)$ with weight $w(e)$ represents the strength of the influence that node u imposes on node v. In the IC and LT models time, t, runs in discrete time periods and each period is called a *round*. The set of nodes, $S \subset V$ "infected" in the first round is called the "seed set". With this notation we state the formal definition as follows:

Definition 1 (Independent Cascade Model [20]). *Let G be an influence network and let $S \subset V$ be the seed set that becomes infected in round $t = 1$. In each round $t \geq 2$, a node v that was infected in round $t - 1$, influences each of his out-neighbours, u, with probability $w(v, u)$. The process terminates at round t when no nodes are infected in this round.*

Intuitively, in the IC model, a newly infected node v has only one chance to infect each of its neighbours (each with independent probability). This happens in the round that immediately follows v's infection.

To define the LT model, we assign to each node $v \in V \setminus S$ an activation threshold, t_v, from the set $[0, 1]$ uniformly at random.

Definition 2 (Linear Threshold Model [29]). *Let G be an influence network with t_v sampled for each $v \in V \setminus S$. Furthermore, let $S \subset V$ be the seed set that becomes infected in round $t = 1$. In each round $t \geq 2$, a node v becomes infected if the influence from his infected in-neighbours exceeds t_v, i.e.:*

$$\sum_{u:u \text{ is infected}} w(u, v) \geq t_v.$$

The process terminates at round t when no nodes are infected in this round.

Unlike in the IC model, where an active node has only one chance to activate each of its neighbors, in the LT model, node v can contribute to activation of its neighbour at any time period.

Having defined both influence models, let us now introduce the function $\sigma : 2^V \to \mathbb{R}$ that, for each seed set $S \subset V$, returns the expected number of infected nodes in a given model (at the end of the process). We will call σ the *spread function* and write $\sigma_{IC}(S)$ and $\sigma_{LT}(S)$ to denote the spread function for the respective models. We will now formalize the top-k nodes problem.

Definition 3 (The influence maximization problem (top k-nodes problem)). *Let G be an influence network, $k \leq |V|$ a positive integer, and M an influence model (either IC or LT). The influence maximization problem is the problem of finding a seed set $S \subseteq V : |S| = k$ that maximizes the σ_M function. Formally:*

$$S \in 2^V : \sigma_M(S) = \max_{S' \in 2^V, |S'|=k} \sigma_M(S').$$

We conclude with a basic property of defined models: an additive decomposition of the spread function. To this end, we formalize first what the activation probability of a node is.

Definition 4 (Activation probability of a node). *Let G be an influence network, $S \subseteq V$ the initial seed set, and M an influence model (either IC or LT). We define the activation probability of a node v, denoted $\mathrm{ap}_M^G : (V, 2^V) \to [0, 1]$, as follows:*

$$\mathrm{ap}_M^G(v, S) = P(v \text{ is infected when } S \text{ is the seed set under } M),$$

where we will omit the superscript in $\mathrm{ap}_M^G(v, S)$, when G is clear from the context.

Kempe et al. [29] proved the following result.

Lemma 1 (Spread function node-additivity). *Given the model M (M is either IC or LT) the expected number of nodes infected if the infection starts from seed set $S \subset V$ is the sum of activation probabilities of all nodes:*

$$\sigma_M(S) = \sum_{v \in V} \mathrm{ap}_M(v, S).$$

2.3 Game-theoretic Centrality

Grofman and Owen [25] were the first to propose using game-theoretic solution concepts as measures of centrality. The basic idea behind this approach is as follows. First, we define a coalitional game over a network, where the nodes are treated as players and the value of each group of nodes depends in a certain way on the network topology. Next, we rank the nodes according to their payoffs assigned by the chosen solution concept, such as the Shapley value or the Banzhaf index. Intuitively, the nodes with the highest (weighted) contribution to various groups of nodes will be ranked top by such a centrality measure and *vice versa*.

In this article, we will consider a coalitional game (V, ν) in which vertices are players and the subsets of vertices are coalitions. Furthermore, the characteristic

function $\nu : 2^V \to \mathbb{R}$ will be a function that depends in some way on the graph G, with $\nu(\emptyset) = 0$. We will use the phrase "value of coalition C" to informally refer to $\nu(C)$, where $C \subseteq V$.

In the next section we discuss the Shapley value-based approach to the top-k nodes problem and introduce our refined algorithm.

3 Approaching the k-Nodes Problem with the Shapley Value-based Centrality

We start this section by presenting the approach by Narayanam and Narahari [39]—the first application of a Shapley value-based centrality to the top-k nodes problem. We also discuss the improvements made in a follow-up work by the same authors [40]. Finally, we present our algorithm.

The Shapley value-based approach by Narayanam and Narahari [39]: In the information diffusion context, it seems natural to think of the importance of a node in terms of its impact on neighboring nodes. In this spirit, Narayanam and Narahari defined the characteristic function as the number of nodes that are adjacent to the nodes in a given coalition. Formally, for all $C \subseteq N$:

$$\nu(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ size(neighbours(C)) & \text{otherwise.} \end{cases} \tag{3}$$

In order to approximate the Shapley value for the above characteristic function, Narayanam and Narahari [39] used a Monte Carlo approach.[2] In particular, the authors randomly choose a relatively small (linear with respect to the number of nodes) subset of all permutations of nodes. Next, for each permutation π, they compute the marginal contribution of each node v_i to the coalition of nodes preceding v_i in π, i.e. $\nu(C_\pi(v_i) \cup \{v_i\}) - \nu(C_\pi(v_i))$. Finally, the average marginal contribution of v_i over all randomly chosen permutations is the approximation of $\mathrm{SV}(v_i)$.

A key shortcoming of simulations performed by Narayanam and Narahari [39] is the limited number of Monte Carlo iterations—for instance the authors run only 10,000 iterations for a network of 8,319 nodes. Unfortunately, as shown by Aadithya et al. [1], the approximation error of the Shapley value for the characteristic function (3) caused by an insufficient number of iterations may be significant. Interestingly, however, Aadithya et al. proved that it is possible to compute the Shapley value for the influence game (3) exactly and this in $O(|V| + |E|)$ time (see Appendix A for more details). In particular, the closed-form formula for the Shapley value of node $v \in V$ in the game defined by the characteristic function (9) is as follows:

$$\mathrm{SV}(v) = \sum_{v_k \in \{v \cup N_{out}(v)\}} \frac{1}{1 + \deg_{in}(v_k)}. \tag{4}$$

[2] The issue of approximating the Shapley value with Monte Carlo techniques has been extensively studied in the literature.

In our simulations in Section 5, we use this positive result in order to compute the Shapley value-based centrality defined by Equation (9) exactly. By doing so, we present the first evaluation of Narayanam and Narahari's (2008) approach using the exact and not an approximate algorithm.

SPIN algorithm by Narayanam and Narahari [40]: [39] revised their approach by developing the SPIN algorithm [40]. As previously, SPIN relies on the Shapley value-based centrality to determine the most influential nodes. However, the characteristic function is changed as follows:

$$\nu(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ \sigma_{\mathrm{LT}}(C) & \text{otherwise.} \end{cases} \tag{5}$$

Unfortunately, this case is much more computationally challenging than previously. Firstly, no polynomial-time algorithm is known for the Shapley value given by this characteristic function; hence, one needs to rely on the Monte Carlo sampling. Secondly, even the Monte Carlo sampling is more challenging as the values of all coalitions, $\nu(C) = \sigma_{\mathrm{LT}}(C)$, have to be approximated [29]. Narayanam and Narahari proposed the following approximation procedure:

(1) Choose a random subset $\Pi' \subset \Pi$ of all possible permutations of players, where $|\Pi'| << |\Pi|$;
(2) For each $\pi \in \Pi'$ and each node $v_i \in \pi$ use a *nested Monte Carlo* technique to approximate the marginal contribution of v_i to the coalition of all nodes that precede it in π, i.e., $C_\pi(v_i)$. This step is necessary due to the intractability of $\nu(C_\pi(v_i) \cup \{v_i\})$ and $\nu(C_\pi(v_i))$. In particular:
   (2a) Sample the spread of influence of the LT model assuming that $C_\pi(v_i) \cup \{v_i\}$ and $C_\pi(v_i)$ are the seed sets.
   (2b) For each sample, compute the marginal contribution of v_i to $C_\pi(v_i)$.
   (2c) Repeat (2a) and (2b) either 6,000 or 10,000 times depending on the size of the network.
   (2d) Compute the approximate marginal contribution of v_i to $C_\pi(v_i)$ as the average marginal contribution of v_i across all samples;
(3) Compute the approximation of the Shapley value of v_i as the weighted average of average marginal contributions of v_i across all $|\Pi'|$ permutations;
(4) Sort the nodes according to their approximated Shapely value and return the resulting ranking.

As we can see, the above procedure involves one Monte Carlo sampling nested in another Monte Carlo sampling. Thus, the actual number of spread simulations that needs to be performed is the product of the number of iterations in both Monte Carlo samplings. In fact, because of this prohibitive computational cost, we are able to report in the simulation section results for this method only for networks with fewer than one hundred nodes and using a very small number of Monte Carlo iterations (see Section 5 for more details).

DSV algorithm by Adamczewski et al. [2]: Recently, Adamczewski et al. proposed a refinement of the algorithm by Narayanam and Narahari [39] called

DSV. They compute the Shapley value precisely using algorithm by Aadithya et al. [1], but improve the way the nodes are taken to the resulting seed set. The algorithm achieves results that are only slightly better than previous approaches [39, 40], but the algorithm is considerably faster.

4 The Shapley Value- and Banzhaf Index-Based Centralities In Local DAGs

In this section, we present our second approach to the influence maximization problem. It builds upon the work by Chen et al. [8], who proposed a greedy algorithm that, in the search for the top k nodes, focuses on the influence spread only in local directed acyclic graphs (DAGs). By the means of simulations, the authors showed that their algorithm returns solutions that are of comparable quality to the original greedy algorithm that operates on the entire graph. An important feature of this approach is that, from a computational point of view, each local DAG is typically much easier to deal with than the entire network. Furthermore, multiple local DAGs can be analysed in parallel. Since Chen et al. called their algorithm the Local DAG, we will abbreviate it to LDAG.

We extend the approach of Chen et al. [8] by introducing game-theoretic centrality measures—the Shapley value- and the Banzhaf index-based ones—to their model. Interestingly, as we will show below, it is possible to parallelize the computation of both of them. By doing so, we are able to fully utilize the advantages of considering local DAGs.

The remainder of this section is organized as follows. We first introduce the model by Chen et al. [8] and discuss their main computational results (subsection 4.1). Next, we present our refinement of this model based on the Shapley value and the Banzhaf index (subsection 4.2). Finally, we show that it is possible to significantly speed up our algorithm in the case of the Banzhaf index.

4.1 The LDAG Model by Chen et al.

In this subsection, we present the results of Chen et al. [8]. We first restate complexity results that motivate the LDAG approach. Next, we describe the LDAG model and conclude with the theoretic results of the influence maximization problem under this model.

The motivation behind Chen's et al. LDAG method comes from the observation that there is little one can do to tackle the complexity of the top k-nodes problem if the whole network is taken into account. In particular:

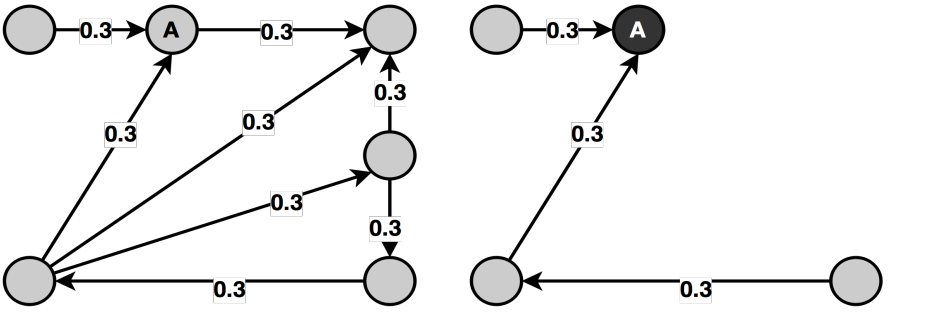
- Computing the spread function is #P-hard under both diffusion models;
- Moreover, even if, hypothetically, we could compute the spread function in polynomial time, then finding a set that maximizes the spread function would be still NP-hard.

Therefore, Chen et al. [8] propose to tackle the complexity of the problem by reducing the extent of the diffusion process. To this end, they observe that the

problem can be solved in linear time with respect to the number of nodes, when G is a directed acyclic graph. Precisely, they introduce a notion of *(Local) DAG rooted at* v, a directed acyclic subgraph of the original graph where:

- v has no out-neighbors; and
- every other node has at least one out-neighbor in LDAG; and
- it is a maximal subgraph fulfilling previous conditions.

Alternatively, one can think of DAG rooted at v as a subgraph of the original graph consisting of nodes that can be reached from v going in the reverse direction of the edges (see Figure 1 for an illustration).



(a) Original graph (b) LDAG rooted at A with threshold 0.08

Fig. 1. Example o LDAG constructed for a given graph.

Now, it's easy to observe that for a DAG rooted at v with vertex set V', we can compute the influence spread on v of the seed set S using any graph traversing algorithm starting from every $w \in S \cap V'$. Furthermore, the value of $\sigma(S)$ in the original graph is simply the sum of influence spreads on root node computed in LDAGs rooted at all nodes.

Acyclicity is a very strong assumption that is rarely observed in social networks. However, solution for the DAG case leads us to an approximation of the influence in the network (not necessary acyclic) on each individual node by assuming that diffusion takes place only in a particular local directed acyclic graph, precisely:

(1) For each node v in G, we choose a directed acyclic graph ldag(v) that is responsible for a high fraction of the influence on v[3];
(2) We use acyclicity to compute the approximated spread σ_v on every node in its LDAG (ldag(v));
(3) We obtain the approximated spread, σ', that is the sum of spreads of all the nodes.

[3] Chen et al. prove that choosing an optimal LDAG for a node is NP-hard, but they also observe that greedy algorithm for that performs very well in practice.

Algorithm 1: Computing the Shapley value in the LDAGs model

```
foreach v_i ∈ V do
    ldag[v_i] ← ldag rooted at v_i;
    for 1 to MCRuns do
        π ← random permutation of nodes in ldag;
        foreach node ∈ ldag do
            SV[node]+ = marginal contribution of node to v_i in π;
        end
    end
    SV[v_i] = SV[v_i]/MCRuns;
end
```

The choice of LDAGs leads us to the definition of new influence models, where influence spreads only in the given LDAGs. They are an approximation of the original IC and LT models. We call them LDAGs models and denote by σ_M (where $M \in \{\text{LDAG-LT, LDAG-IC}\}$) the expected number of infected nodes. However, as shown by Chen et al. [7], it is still NP-hard to compute the set that maximizes the σ function in these new models.

4.2 Refinement of the LDAG Model Based on the Game-Theoretic Centralities

Since we assume that we can reduce the entire network to a set of LDAGs, we find the initial seed set by analyzing the most influential nodes in all the LDAGs. While, in order to achieve this, Chen et al. [8] use a greedy approach, we propose an alternative approach that uses the Shapley value and the Banzhaf index as measures of node centrality in the LDAGs. Since the computation of both solution concepts is usually challenging, we use Monte Carlo simulations where we approximate the solution by sampling permutations. Furthermore, we take advantage of the LDAG structure, which is small relative to the size of the network. We also reduce the number of input nodes for the computation of the solution concepts (this reduces the number of necessary Monte Carlo simulations). The pseudocode for our approach using LDAGs is presented in Algorithm **??**. We precede it by the definitions and calculations that introduce it.

Definition 5 (Influence LDAG cooperative games). *Given an influence graph $G = (V, E, w)$, let V be a set of players, and the function σ_M (where $M \in \{\text{LDAG-LT}, \text{LDAG-IC}\}$) a characteristic function of a cooperative game. We call this game the cooperative influence game for a given model.*

We compute the Shapley value and the Banzhaf index assuming that the characteristic function ν is the $\sigma_M()$ influence function in LDAG approximation under both the IC and LT models. The theoretical analysis of the problem is presented

below, where we use the following additional notation: $X_v^G(S)$ denotes the probability that node v in graph G is infected, when S is the initial seed; $\mathrm{ldag}(v)$ denotes the LDAG for node v; and $\mathrm{ldag}_{+v}(u)$ and $\mathrm{ldag}_{-v}(u)$ denote *ancestors* and *predecessors*, respectively, of v in $\mathrm{ldag}(u)$. For marginal contribution μ of v to S we obtain:

$$\mu(v,S) = (\sigma(S \cup \{v\}) - \sigma(S)) \tag{6}$$

$$= \sum_{u \neq v} X_u^{\mathrm{ldag}(u)}(S \cup v) - X_u^{\mathrm{ldag}(u)}(S) \tag{7}$$

$$= \sum_{u \neq v} (1 - X_v^{\mathrm{ldag}(u)}(S)) \cdot X_u^{\mathrm{ldag}(u) \setminus S}(v). \tag{8}$$

Using Equation (1) we obtain the following expression for the Shapley value that we use directly:

$$\mathrm{SV}(v) = \sum_{\pi \in \Pi(N)} \frac{1}{|N|!} \sum_{u \neq v} (1 - X_v^{\mathrm{ldag}(u)}(S_\pi(v))) \cdot X_u^{\mathrm{ldag}(u) \setminus S_\pi(v)}(v).$$

The above analysis leads us to a Monte Carlo algorithm that computes the Shapley value and analogously the Banzhaf index (Algorithm **??**). We note that in our approach to the LDAG model based on the Shapley value and the Banzhaf index, we take advantage of the additive nature of these two solution concepts. Specifically, in order to calculate the Shapley value (or Banzhaf Index respectively) of a node, it is enough to add together the "partial Shapley values" for that node computed separately in each LDAG that contains this node. As a result, this game-theoretic approach, as opposed to the greedy approach in Chen et al. [8], is particularly suitable for distributed systems which facilitate and accelerate the computation of the most influential nodes. Unlike the game-theoretic approach, the greedy method of Chen et al. would require more substantial communication between the LDAGs after each iteration in order to see which node was the most influential, and then pass on the message of which node was added to the seed set. In Algorithm **??** we present a version of the LDAG Shapley value algorithm in the Map-Reduce [13] schema — currently the most popular parallel and distributed programming model in the industry. The Banzhaf index calculations are analogous, but can be further optimized using the structure of the LDAGs to obtain even better approximation.

5 Experimental Results

In this section we compare the performance of the following algorithms for the influence maximization problem:

(i) The algorithms from the literature:
- **Fringe game** — the Shapley-value based algorithm proposed by Narayanam and Narahari [39] but calculated using the polynomial time algorithm from Aadithya et al. [1] (see Section 3).

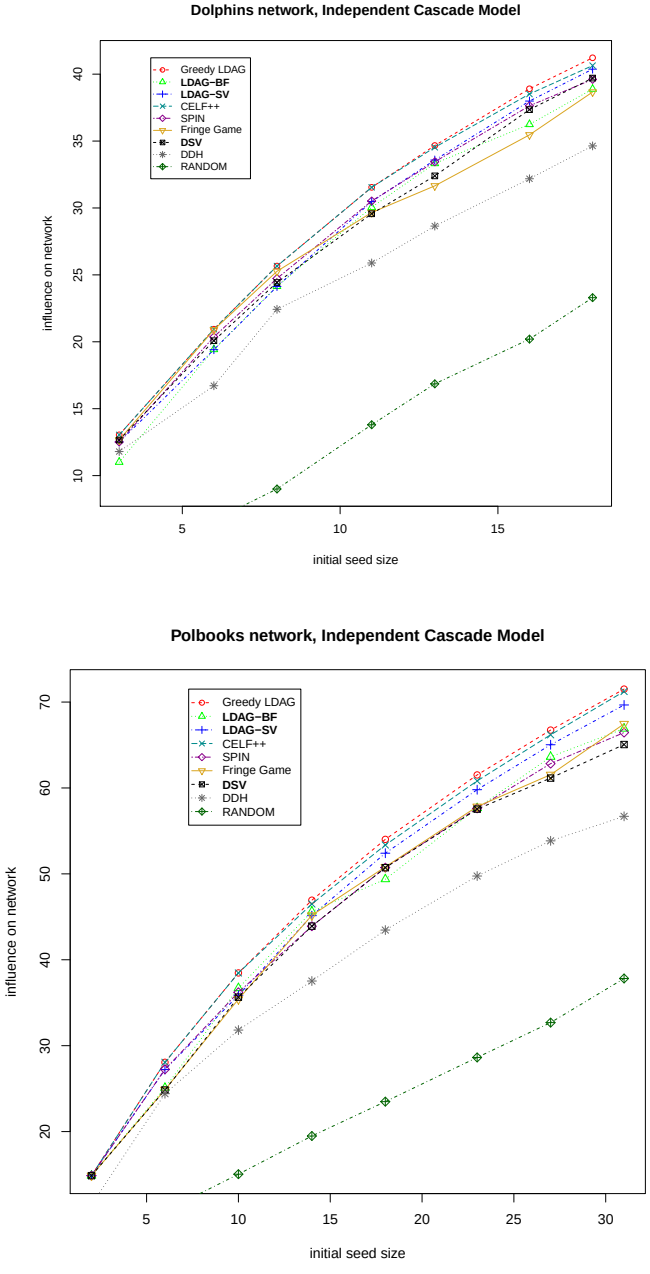


Fig. 2. Comparison of algorithms performance (influence vs. seed size) on Dolphins and Polbooks data

Algorithm 2: The LDAG Shapley Value - the Map Reduce schema

```
1 Function map(headNode, ldag(headNode))
2 |   forall the nodes in ldag(headNode) do
3 |   |   sv ← compute SV for node in ldag
4 |   |   Emit (node, sv)
5 |   end
1 Function reduce(node, list of SVs in LDAGs)
2 |   Emit sum of nodes SVs in LDAGs
```

- **Greedy LDAG** — the algorithm by Chen et al. [8] (see Subsection 4.1).
- **CELF++** — the algorithm by Goyal et al. [23]. This algorithm is an improvement of the traditional greedy algorithm that drastically limits the number of spread function calls by trying to look ahead and predict next nodes that will be added to the resulting initial seed set.
- **SPIN** — the algorithm by Narayanam and Narahari [40]. In order to perform our experiments we needed to reduce the MC repetitions parameters to 1000 for SV calculation and 4000 for spread function calculation.
- **DDH** - the Discount Degree heuristics by [9]. This algorithm works by iteratively adding to the seed set the node that has most neighbors that are neither already in the seed set nor neighbors of a node in a seed set.
- **RANDOM** — a random selection of the top k nodes (from the uniform distribution).
- **DSV** — the Discounted Shapley Value algorithm [2].

(ii) The algorithms proposed in this article:

- **LDAG-SV** — the refinement of the LDAG algorithm based on the Shapley Value (see Subsection 4.2).
- **LDAG-BF** — the refinement of the LDAG algorithm based on the Banzhaf index (see Subsections 4.2.

We conduct the experiments on both IC and LT diffusion models. Both models are described by means of a live graph[4], where edge weights stand for activation probabilities. In the case of Independent Cascade, each edge e is *live* (or *open*) with some probability p_e (drawn uniformly from the set $(0, 1)$) and *blocked* with probability $1 - p_e$. As shown in Kempe et al. [29], the Linear Threshold model can also be described by the live graph where only at most one incoming edge is live and the sum of incoming probabilities is less than or equal to 1. The nodes active at the end of the process are those reachable by live edges from the initial seed set. In our experiments we use a diverse group of real-world networks:

(a) American College Football network of games between Division IA colleges during the regular Fall 2000 season. The nodes are the teams and the edges are the games between the teams and weights are sampled from uniform distribution $\mathbb{U}[0, 1]$.

[4] A live graph is a subset of a graph, where every edge is taken with a given probability.

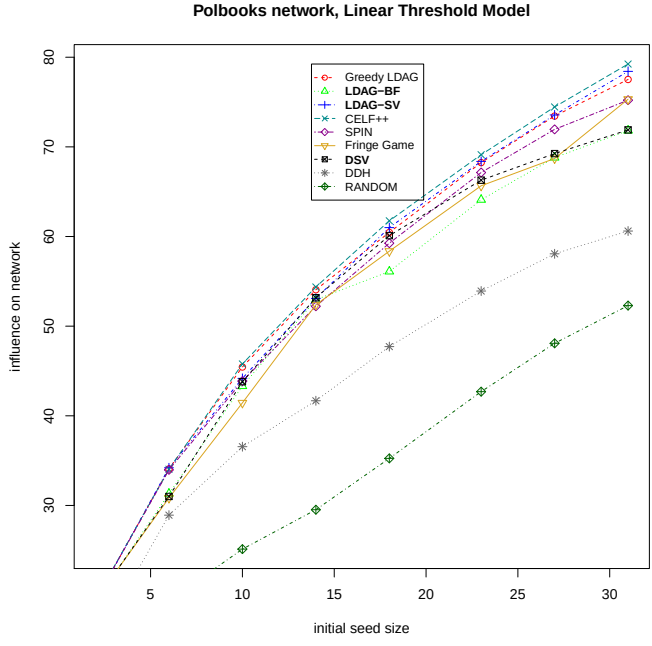


Fig. 3. Comparison of algorithms performance (influence vs. seed size) on Polbooks data for LT model

(b) An undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand (with weights sampled from a uniform distribution).
(c) Nodes represent books about US politics sold by the on-line bookseller Amazon.com. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also bought these other books" feature on Amazon [31].
(d) Co-appearance weighted network of characters in the novel Les Miserables.
(e) A collaboration network of scientists posting preprints on the high-energy theory archive at www.arxiv.org, 1995-1999, as compiled by M. Newman.
(f) Nodes represent pages from Stanford University (stanford.edu) and directed edges represent hyperlinks between them. The data was collected in 2002.

In order to compare algorithms on networks of similar size, we reduce the large networks (d) and (e) to a fraction of their sizes (using BFS walk from a random node and assuring that the obtained graph is not too small).

The experiments consist of two steps. In the first step, we employ the algorithms to find the seed set consisting of k nodes. This is the set of the most influential nodes according to each algorithm. We vary k between 2% and 30% of the total number of the nodes in the network. The second part tests the performance of the seed set by means of Monte Carlo simulations. As far as the quality of the seed set is concerned, the Greedy LDAG performs consistently best across all data sets, seed sizes and in both models (Chen et al. [7] only test it on the LT model). The performance of the CELF++ and Shapley value LDAG approaches is similar in the IC model (Figure **??**), where CELF++ performs slightly better for a smaller seed size and the roles reverse for a larger seed size (see Figure **??**). LDAG-SV performs better in the LT model, which makes sense since LDAG is designed for the LT model. DSV and fringe perform similarly in the IC model and DSV is slightly better in the LT model. SPIN algorithm performs worse than LDAG-SV and in most experiments better than LDAG-BF. We also note the weak performance of the Aadithya et al. [1] threshold model and very unstable performance of the Banzhaf index LDAG method across different datasets (compare Figure **??** and Figure **??**).

In larger networks, with thousands of nodes, the performance of the Shapley value LDAG is only preceded by the greedy LDAG. The three algorithms perform substantially better than the Degree Discount algorithm. It is also worth mentioning that all the Shapley value based heuristics perform better in the LT model than all fast heuristics (Degree Discount and degree), whilst we did not note significant differences in running times (see Figure **??**). The practicality of the heuristics when comparing the running times of the algorithms becomes more visible with the increasing size of the datasets. When only considering graphs with 100 nodes, CELF++ is about 400 times slower than fringe and DSV. The times of greedy approaches (excluding greedy LDAG) are prohibitively large. For large networks, the heuristics run in tens of milliseconds, while the longest LDAG-SV takes hours. Although greedy LDAG maintains competitive running time (a few seconds) for networks with a few thou-

sands nodes, LDAG-SV obtaining similar results has theoretical underpinnings for Map-Reduce distribution. We were unable to conduct experiments for the SPIN algorithm on bigger networks in a reasonable time (on day per dataset). We present only part of the data gathered in the experiments. However, the complete code of the experiments can be found in a repository available at `https://github.com/szymonm/CGMethodsForInfluence`.

6 Conclusions

We proposed a new game-theoretic method to the problem of influence maximization—refinement of the Local DAG algorithm by Chen et al. [8] based on the Shapley value and the Banzhaf index. We also verified the performance of the Shapley value-based centralities proposed earlier in the literature. The experimental results show that the greedy LDAG approach finds the highest quality seed set. Yet, our proposed heuristic based on the Shapley value performs almost as well as the greedy algorithm in terms of solution quality, and it can be easy adopted to the Map-Reduce or other parallel programming scheme.

Acknowledgements

The paper is co-founded by the European Union from resources of the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications", Agreement UDA-POKL.04.01.01-00-051/10-00.

APPENDIX A: The algorithm from Aadithya et al. [1]

Aadithya et al. considered the following characteristic function:

$$\nu^{fringe}(C) = \begin{cases} 0 & \text{if } C = \emptyset \\ size(fringe(C)) & \text{otherwise,} \end{cases} \tag{9}$$

where the *fringe* of a coalition is the set of all nodes in this coalition and its out-neighbours. Formally, $fringe(C)$ is a set $\{v \in V : v \in C$ or $\exists u \in C$ such that $(u, v) \in E\}$. It is easy to observe that both characteristic functions, the one presented in Equations (9) and the one (3), are very similar. The only difference is that the characteristic function in formula (9), in addition to the out-neighbors, also takes into account the number of nodes within the coalition. It is easy to check that this difference has no qualitative bearings on the rankings of the nodes. That is, both rankings will always be the same and both functions can be used interchangeably for the top k-nodes problem. Algorithm **??**, reproduced from Aadithya et al. [1, Algorithm 1], implements formula (4). Note that it has polynomial complexity in the size of the network. We use this positive result in our simulations in Section 5 in order to compute the Shapley value-based centrality defined by Equation (9) exactly. In other words, we present the first evaluation of Narayanam and Narahari's (2008) approach using the exact algorithm and not an approximate one.

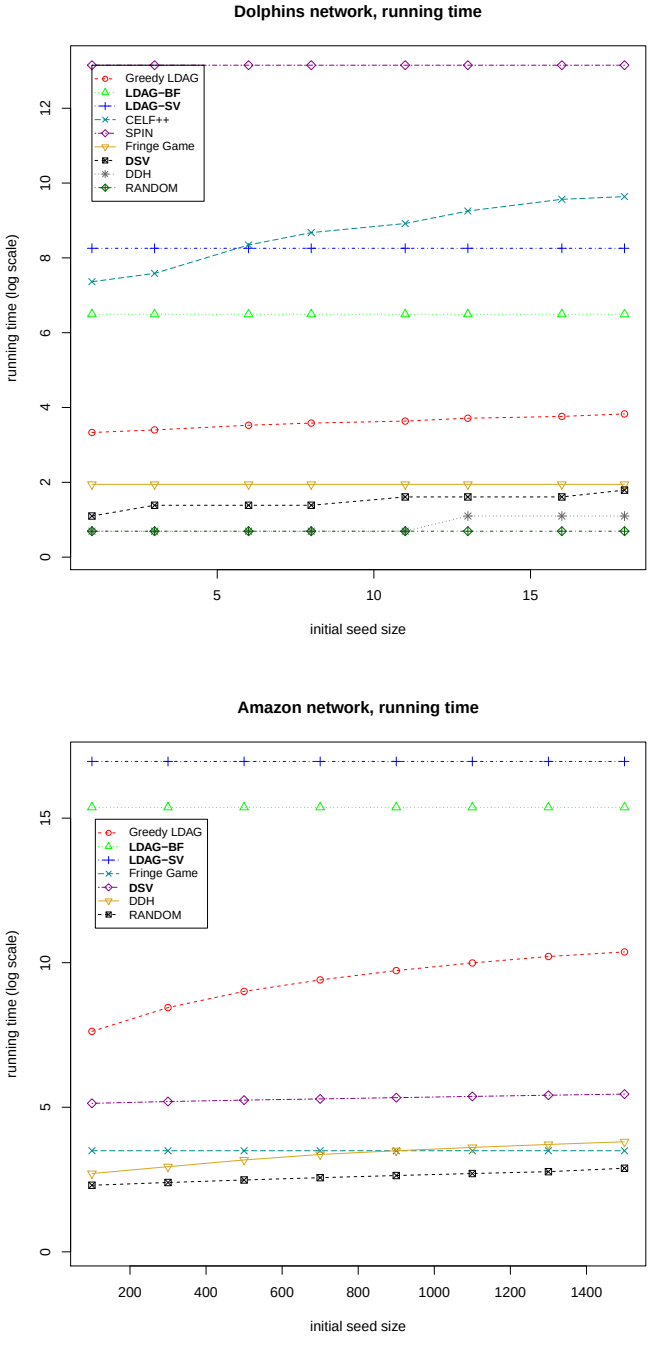


Fig. 4. Running times on Dolphins and Amazon data

Algorithm 3: The Shapley Value for the fringe characteristic function (9)

Data: Diffusion network
Result: Shapley values of all nodes
for $i \leftarrow 1$ **to** n **do**
  foreach $v_j \in outneighbor(v_i)$ **do**
    $shapley[i] += \frac{1}{1+\deg_{in}(v_j)}$;
  end
  $shapley[i] += \frac{1}{1+\deg_{in}(v_i)}$;
end

APPENDIX B: Main notation Used in the Article

	Cooperative games
N	Set of players.
n	Cardinality of set N.
a, a_i	Player in N.
C	Coalition $C \subset N$.
$\nu(C)$	Characteristic function of a coalition C.
$\mu(a, C)$	Marginal contribution of player a to coalition C.
$\mathrm{SV}(a)$	Shapley value of player a.
$\mathrm{BI}(a)$	Banzhaf index of player a.
	Graph theory
V	Set of nodes.
E	Set of edges, $E \subset (V \times V)$.
$G(V, E)$	Graph with vertices V and edges E
v, u	Node is a graph.
$\deg(u)$	Degree of node u.
$w(u, v)$	Weight of an edge from node u to node v.
	Influence
S	Seed set.
M	Influence model, either LT (Linear threshold) or IC (Independent Cascade).
$\sigma_M(S)$	Spread from set S under model M.
$\mathrm{ap}_M^G(v, S)$	Activation probability of node v when S is the seed set under model M in graph G.
$X_v^G(S)$	Probability that v is infected in G when S is the initial seed.
$\mathrm{ldag}(v)$	Local Directed Acyclic Graph (LDAG) for node v.
$\mathrm{ldag}_{+u}(v)$	Ancestors of node u in LDAG for node v.
$\mathrm{ldag}_{-u}(v)$	Predecessors of node u in LDAG for node v.

References

[1] Aadithya, K.V., Ravindran, B., Michalak, T.P., Jennings, N.R.: Efficient Computation of the Shapley Value for Centrality in Network. 6th Int. Work. Internet Netw. Econ. 6484, 1–13 (2010)

[2] Adamczewski, K., Matejczyk, S., Michalak, T.P.: How good is the shapley value-based approach to the influence maximization problem? CoRR abs/1409.7830 (2014), http://arxiv.org/abs/1409.7830

[3] Bachrach, Y., Markakis, E., Resnick, E., Procaccia, A.D., Rosenschein, J.S., Saberi, A.: Approximating power indices: Theoretical and empirical analysis. Auton. Agent. Multi. Agent. Syst. 20(2), 105–122 (2010)

[4] Banzhaf, J.F.: Weighted Voting Doesn't Work: A Mathematical Analysis. Rutgers Law Rev. 19, 317–343 (1965)

[5] Castro, J., Gómez, D., Tejada, J.: Polynomial calculation of the Shapley value based on sampling. Comput. Oper. Res. 36(5), 1726–1730 (2009)

[6] Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational Aspects of Cooperative Game Theory. Synth. Lect. Artif. Intell. Mach. Learn. 5(6), 1–168 (2011)

[7] Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '10. p. 1029. ACM (2010), http://dl.acm.org/citation.cfm?doid=1835804.1835934

[8] Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proc. - IEEE Int. Conf. Data Mining, ICDM. pp. 88–97. IEEE (2010)

[9] Chen, Y.C., Peng, W.C., Lee, S.Y.: Efficient algorithms for influence maximization in social networks. In: Knowl. Inf. Syst. pp. 577–601. No. 3, ACM (2012), http://portal.acm.org/citation.cfm?doid=1557019.1557047

[10] Cohen, S., Ruppin, E., Dror, G.: Feature Selection Based on the Shapley Value. In other words 1, 665–670 (2005), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.4694\&rep=rep1\&type=pdf

[11] Colman, A.M.: Game theory. CUP (2008), http://hdl.handle.net/2381/3973

[12] Conitzer, V., Conitzer, V., Sandholm, T., Sandholm, T.: Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In: Proc. Natl. Conf. Artif. Intell. vol. 4, pp. 219–225 (2004)

[13] Dean, J., Ghemawat, S.: MapReduce. Commun. ACM 51(1), 107 (2008), http://doi.acm.org/10.1145/1327452.1327492

[14] Deng, X., Papadimitriou, C.H.: On the Complexity of Cooperative Solution Concepts. Math. Oper. Res. 19(2), 257–266 (1994)

[15] Domingos, P., Richardson, M.: Mining the Network Value of Customers. In: Proc. Seventh {ACM} {SIGKDD} Int. Conf. Knowl. Discov. Data Min. pp. 57–66 (2001), http://doi.acm.org/10.1145/502512.502525

[16] Everett, M.G., Borgatti, S.P.: The centrality of groups and classes. J. Math. Sociol. 23(3), 181–201 (1999)
[17] Fatima, S., Wooldridge, M., Jennings, N.: A Linear Approximation Method for the Shapley Value. Artif. Intell. 172(14), 1673–1699 (2008), http://eprints.soton.ac.uk/265802/
[18] Fatima, S., Wooldridge, M., Jennings, N.R.: A randomised method for the shapely value for the voting game. In: Proc. 6th Int. Jt. Conf. Auton. agents multiagent Syst. p. 157. ACM (2007), http://eprints.soton.ac.uk/264220/
[19] Freeman, L.C.: Centrality in social networks conceptual clarification. Soc. Networks 1(3), 215–239 (1978)
[20] Goldenberg, J., Libai, B., Muller, E.: Using Complex Systems Analysis to Advance Marketing Theory Development. Acad. Mark. Sci. Rev. 2001(3), 1–18 (2001), http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Using+Complex+Systems+Analysis+to+Advance+Marketing+Theory+Development+:+Modelling+Heterogeneity+Effects+on+New+Product+Growth+through+Stochastic+Cellular+Automata\#0
[21] Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring Networks of Diffusion and Influence. In: Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. pp. 1019–1028. ACM (2010), http://arxiv.org/abs/1006.0234
[22] Goyal, A., Lu, W., Lakshmanan, L.V.S.: SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. In: Icdm. pp. 211–220. IEEE (2011)
[23] Goyal, A., Lu, W., Lakshmanan, L.V.: Celf++. In: Proc. 20th Int. Conf. companion World wide web - WWW '11. p. 47. ACM (2011), http://portal.acm.org/citation.cfm?doid=1963192.1963217
[24] Granovetter, M.S.: Threshold Models of Collective Behavior. Am. J. Sociol. 83(6), 1420–1443 (2007)
[25] Grofman, B., Owen, G.: A game theoretic approach to measuring degree of centrality in social networks. Soc. Networks 4(3), 213–224 (1982)
[26] Jackson, M.O.: Social and Economic Networks 1. No. March, Princeton University Press Princeton (2008)
[27] Jackson, M.O., Watts, A.: The Evolution of Social and Economic Networks. J. Econ. Theory 106(2), 265–295 (2002), http://www.sciencedirect.com/science/article/pii/S0022053101929035$\backslash$npapers2://publication/uuid/97AA7660-4F32-4C92-BD5C-28A9D9F6991A
[28] Joyce, K.E., Laurienti, P.J., Burdette, J.H., Hayasaka, S.: A new measure of centrality for brain networks. PLoS One 5(8), e12200 (2010)
[29] Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proc. ninth ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '03. p. 137. ACM (2003), http://portal.acm.org/citation.cfm?doid=956750.956769
[30] Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM 46(5), 604–632 (1999)
[31] Krebs, V.: Books about US politics (1988)

[32] Krebs, V.E.: Mapping Networks of Terrorist Cells. Connections 24(3), 43–52 (2002), http://www.insna.org/pubs/connections/v24.html
[33] Lehrer, E.: An axiomatization of the Banzhaf value. Int. J. Game Theory 17(2), 89–99 (1988)
[34] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective Outbreak Detection in Networks. In: Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. pp. 420–429. ACM (2007), http://eprints.pascal-network.org/archive/00005342/
[35] Littlechild, S.C., Owen, G.: A Simple Expression for the Shapley Value in a Special Case. Manage. Sci. 20(3), 370–372 (1973)
[36] Ma, R.T.B., Chiu, D.M., Lui, J.C.S., Misra, V., Rubenstein, D.: Internet economics: The use of shapley value for isp settlement. IEEE/ACM Trans. Netw. 18(3), 775–787 (2010)
[37] Mahajan, V., Muller, E., Bass, F.M.: New Product Diffusion Models in Marketing: A Review and Directions for Research. J. Mark. 54(1), 1–26 (1990)
[38] Maniquet, F.: A characterization of the Shapley value in queueing problems. J. Econ. Theory 109(1), 90–103 (2003)
[39] Narayanam, R., Narahari, Y.: Determining the top-k nodes in social networks using the shapley value. In: Proc. 7th Int. Jt. .... pp. 2–5. International Foundation for Autonomous Agents and Multiagent Systems (2008), http://dl.acm.org/citation.cfm?id=1402911
[40] Narayanam, R., Narahari, Y.: A shapley value-based approach to discover influential nodes in social networks. IEEE Trans. Autom. Sci. Eng. 8(1), 130–147 (2011)
[41] Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. (1999)
[42] Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proc. eighth ACM SIGKDD Int. Conf. Knowl. Discov. data Min. KDD 02. p. 61. No. 3 (2002), http://portal.acm.org/citation.cfm?doid=775047.775057
[43] Shapley, L.S.: A Value for n-person Games. Ann. Math. Stud. 28, 307–317 (1953)
[44] Tarkowski, M., Michalak, T., Rahwan, T., Elkind, E., Wooldridge, M.: Game-theoretic Network Centrality Measures—A Critical Survey (2014)
[45] Valente, T.W.: Network models of the diffusion of innovations. 1381-298X 1572-9346 (Online), Issue 2, Number 2 / June 2(2), 163–164 (1996)