On Manipulating Weight Predictions in Signed Weighted Networks

Tomasz Lizurej[1,2], **Tomasz P. Michalak**[1,2], **Stefan Dziembowski**[1,2]
[1] University of Warsaw
[2] IDEAS NCBR
tomasz.lizurej@crypto.edu.pl, tpm@mimuw.edu.pl, stefan.dziembowski@crypto.edu.pl

Abstract

Adversarial social network analysis studies how graphs can be rewired or otherwise manipulated to evade social network analysis tools. While there is ample literature on manipulating simple networks, more sophisticated network types are much less understood in this respect. In this paper, we focus on evading Fairness-Goodness Algorithm which is an edge weight prediction method for signed weighted networks developed by Kumar et al. in 2016. Among others, this method can be used for trust prediction in reputation systems. We study the theoretical underpinnings of this algorithm and its computational properties in terms of manipulability. Our positive finding is that, unlike many other tools, this measure is not only difficult to manipulate optimally, but also it can be difficult to manipulate in practice.

Introduction

Adversarial social network analysis studies how networks can be rewired or otherwise manipulated to falsify network examination. In particular, many works in this body of research studied how to manipulate classic tools of social network analysis such as centrality measures (Crescenzi et al. 2016; Bergamini et al. 2018; Was et al. 2020), and community detection algorithms (Waniek et al. 2018a; Fionda and Pirro 2017; Chen et al. 2019). Also, a rapidly growing body of works studies adversarial learning on graphs using deep learning (Chen et al. 2020).

While most of the above literature focused on simple networks, in this paper, we consider a more complex model of weighted signed networks. In this class of networks, links are labeled with real-valued weights representing positive or negative relations between the nodes (Leskovec, Huttenlocher, and Kleinberg 2010a,b; Tang et al. 2016). An important application of signed weighted networks is the modelling of trust networks/reputation systems, the goal of which is to avoid transaction risk by providing feedback data about the trustworthiness of a potential business partner (Resnick et al. 2000). As an example, let us consider the cryptocurrency trading platform Bitcoin OTC (Kumar et al. 2016). In this platform, users are allowed to rate their business partners on the scale $\{-10, -9, \dots, 10\}$, and the ratings are publicly available in the form of a who-trusts-whom network. A 6-node sample fragment is presented in Figure 1.

A user who thinks of doing a transaction with another user for the first time can use the information from such a who-trust-whom network to predict the potential risk. Technically, given a trust network modeled as a weighted signed network, predicting trust amounts to predicting the weights of potential new edges. A well-known edge weight prediction method, called FGA, was proposed by (Kumar et al. 2016). FGA is based on two measures of node behavior: the **goodness** that evaluates how much other nodes trust a given node, and the **fairness** that captures how fair this node is in rating other nodes. Both concepts have a mutually recursive definition that converges to a unique solution. Most importantly, Kumar et al. showed that FGA is effective in predicting edge weights, i.e., the level of trust between unlinked nodes. For example, in Figure 1, the trust of node 1031 towards node 715 is predicted by FGA to be 2.26.

While FGA seems to be an interesting tool to apply in practice, little is known about its resilience to malicious behaviour. In this paper, we present the first study of manipulating the FGA function by a *rating fraud* (Cai and Zhu 2016; Mayzlin, Dover, and Chevalier 2014). It involves fraudulent raters to strategically underrate or overrate other users for their own benefit. To magnify the strength of the manipulation, the attacker may create and act via multiple fake user identities. Such so called *Sybil attacks* are especially tempting in environments such as cryptocurrency trading platforms where creating a new identity is affordable. Rating fraud attacks may be *direct*—when targeted nodes are rated directly by the attackers—and *indirect*—when the

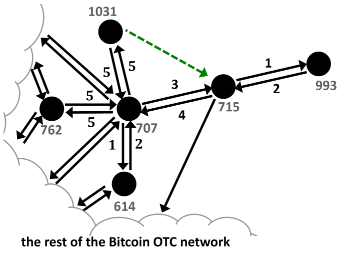


Figure 1: A fragment of the Bitcoin OTC network composed of nodes 993, 715, 707, 614, 1031, 762.

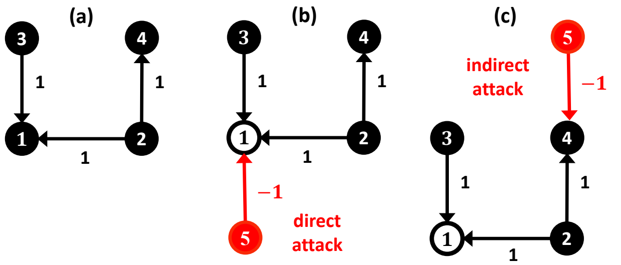
attackers try to manipulate the neighbourhood of the target nodes rather than the target nodes themselves (see Figure 2). It is important to distinguish between direct and indirect manipulations, as in some situations, only indirect ones will be practical. This may be the case on e-commerce platforms such as e-Bay, where nodes rate each other only after completing a transaction. When a retailer of expensive products is the target, the cost of a direct attack can be prohibitive. Hence, an indirect attack becomes an attractive alternative—it may be much cheaper to attack through the clients or business partners of such an expensive retailer (see the next section for an example).

Our contributions can be summarised as follows:

- To analyze the theoretical underpinnings of the FGA measure, we propose the system of basic axioms for both fairness and goodness. We prove that together they uniquely determine the FGA measure;
- Next, we formulate the issue of manipulating the FGA measures of some target group of nodes as a set of computational problems. We then prove that all these problems are NP-hard and $W[2]$-hard, i.e., FGA is, in general, hard to manipulate.
- Given the hardness of attacking a group of nodes, we then focus our analysis on targeting a single node - directly or indirectly. We first prove that direct attacks on a single node are easy, i.e., it is easy for an attacker to directly rate the target node to change the sign of her *goodness* value. As for an indirect attack, we show analitycally that for some class of networks (which we call *minimum-k-neighbour graphs*, since we require that every node in this network has $indegree$ and $outdegree$ at least k), we can bound the strength of indirect attacks. Our positive finding is that, in this case, FGA measure turns out to be rather difficult to manipulate.
- In our experimental analysis, we first evaluate two benchmarks: (a) the strength of the aforementioned direct attack, and (b) the strength of an indirect attack based on a simple greedy approach. The latter one turns out to be very ineffective. Next, we analyse an improved greedy approach by attacking at a larger scale in every step. This approach, although costly, proves to be often effective.

Preliminaries

A Weighted Signed Network (WSN) is a directed, weighted graph $G = (V, E, W)$, where V is a set of users, $E \subseteq V \times V$ is a set of (directed) edges, and $\omega : E \rightarrow [1, +1]$ is a weight function that to each $(u, v) \in E$ assigns a value between -1 and $+1$ that represents how u rates v. For any directed edge $(u, v) \in E$, let us denote by $\overline{(u, v)}$ the edge in the opposite direction, i.e., $\overline{(u, v)} = (v, u)$. For any set of directed edges E, denote by $\overline{E} = \{\overline{e} : e \in E\}$. Furthermore, let P be a set of pairs of nodes of cardinality n, i.e, $P = \{\{u_1, v_1\}, ..., \{u_n, v_n\}\}$. The domain of P is the set of nodes that make the pairs in P, i.e. $dom(P) = \{u : u \in \{u, v\} \in P\}$. Finally, we write $Pred(v)$ (resp. $Succ(v)$) to denote the set of *predecessors* (resp. *successors*) of v (resp. u) defined as follows: $Pred(v) = \{u : u \in (u, v) \in E\}$ (resp. $Succ(u) = \{v : v \in (u, v) \in E\}$).



Network	(a)		(b)		(c)	
node v	$g(v)$	$f(v)$	$g(v)$	$f(v)$	$g(v)$	$f(v)$
1	1	1	**0.4**	1	**0.83**	1
2	1	1	1	0.8	1	0.75
3	1	1	1	0.7	1	0.92
4	1	1	0.8	1	0.17	1
5			1	0.3	1	0.42

Figure 2: Sample networks and two types of attacks.

For a square matrix $M^{m \times m}$, we define $||M||_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{m} m_{ij}$, $||M||_1 = \max_{1 \leq j \leq m} \sum_{i=1}^{m} m_{ij}$. It is also known that $||M \times M||_\infty \leq ||M||_\infty \cdot ||M||_1$ and $||M \times M||_\infty \leq ||M||_1 \cdot ||M||_1$ (see https://en.wikipedia.org/wiki/Matrix_norm).

Kumar et al. (2016) define a recursive function, FGA, that assigns to each vertex of a weighted directed graph two values: *fairness* and *goodness*, $(f(v), g(v))$. The first one, $f(v)$, assigns a real value from range $[0, 1]$ to v that indicates how *fair* this node is in rating other nodes. The second one, $g(v)$, assigns a value from range $[-r, r]$ to v indicating how much *trusted* this node is by other nodes (for simplicity we assume that $r = 1$ in this paper). Finally we define an in-degree ($indeg(u)$) and out-degree ($outdeg(u)$) of a node $u \in V$. $indeg(u) = |\{(v, u) : (v, u) \in E\}|$ and $outdeg(u) = |\{(u, v) : (u, v) \in E\}|$. Kumar et al.'s recursive formula for $(f(v), g(v))$ is as follows:

$$g(v) = \frac{1}{indeg(v)} \sum_{u \in Pred(v)} f(u) \times \omega(u, v) \qquad (1)$$

$$f(u) = 1 - \frac{1}{outdeg(u)} \sum_{v \in Succ(u)} \frac{|\omega(u, v) - g(v)|}{2}, \qquad (2)$$

where $g(v) = 1$ for $v \in V$ with $indeg(v) = 0$, and $f(v) = 1$ for $v \in V$ with $outdeg(v) = 0$.

Kumar et al. (2016) showed that this function can be computed iteratively starting from $f^{(0)}(u) = g^{(0)}(u) = 1$. Theorem 1 from the aforementioned work states that at each step, t, the estimated values $f^{(t)}(u)$, $g^{(t)}(u)$ get closer to their limits $f^{(\infty)}(u), g^{(\infty)}(u)$, i.e. we have $|f^{(\infty)}(u) - f^{(t)}(u)| < \frac{1}{2^t}$ and $|g^{(\infty)}(u) - g^{(t)}(u)| < \frac{1}{2^{t-1}}$. The FGA function can be used for predicting the weight of some not-yet existing (or unknown) edge $(u, v) \in V \times V \setminus E$ by computing the product: $\omega(u, v) = f(u) \times g(v)$.

As an example of the FGA function and how it could be attacked, let us consider Figure 2. Network (a) is a benchmark, where every node rates others with the highest possible value.In network (b), a new node 5 is used to perform a **direct attack** by rating node 1 with the worst possible value of -1. This decreases the goodness of node 1 to 0.4. However, as argued in the introduction such a direct attack can be prohibitively costly. Nevertheless, given the definition of the FGA function, node 5 can also perform an indirect attack on node 1. This can be done, for instance, by directly attacking node 4. As node 4 has already been rated positively by node 2, an opposite rating introduced by 5 will decrease the fairness of 2. In particular, comparing network (c) to (a) in Figure 2, the fairness of 2 decreased from 1 to 0.75. This lower fairness means that node's 2 ratings are less meaningful in network (c) than in network (a). Hence, the goodness of node 1 decreases to 0.83.

Axiomatization

Our first result is an axiom system that completely characterizes the FGA. Below we present a comprehensive summary, while the details will be available in the full version of the paper.

We begin with the characterization of the goodness part of the FGA function. Recall that the idea behind the goodness of v is that it should reflect how this node is rated by its predecessors. Moreover, the ratings of the fairer predecessors should count more. We translate these high-level requirements into the following axioms:

- SMOOTH GOODNESS—let all predecessors of a particular node, $v \in V$, be unanimous in how they rate v and let their fairness be the same. Now, let us assume that their fairness increases equally, i.e., intuitively, the nodes that rate v become more trustworthy. Then, we require that this will result in an increase of the goodness of v, and that this increase is proportional to the increase of the fairness of v's predecessors;
- INCREASE WEIGHT—let the predecessors of v be all equally fair and unanimous in how they rate v. Now, let them increase their rating of v equally. Then, we require that the goodness of v increases and that this increase is proportional to the increase in how v is rated;
- MONOTONICITY FOR GOODNESS—the predecessors with higher fairness should have a bigger impact on the goodness of v. Similarly, higher weights should also have a bigger impact;
- GROUPS FOR GOODNESS—let v be rated by k groups of the predecessors and let the nodes in each group be homogeneous and unanimous w.r.t. v. What is then the relationship between the impact these groups have on the goodness of v? In line with the previous axioms,we require that the goodness of v should be equal to the *weighted average* of the ratings achieved when these groups separately rate v;
- MAXIMAL TRUST—this basic condition requires that any if all the predecessors of v have the highest possible fairness and their ratings are the highest possible, then the goodness of v should be the highest possible;
- BASELINE FOR GOODNESS—a non-rated node has the goodness of 1.

Our first result is that the above axioms uniquely define the goodness part of the FGA function.

Let us now characterise the fairness part of the FGA function. Recall that the idea behind the fairness of v is that it should reflect how the ratings given by this node agree with the ratings given by other nodes, i.e. how erroneous v is. In this respect, we have the following axioms:

- SMOOTH FAIRNESS—this axiom stipulates that the fairness of a node making an average error is an average of the fairness values of nodes making extreme errors;
- MONOTONICITY FOR FAIRNESS—our first axiom stipulates that the fairness of a node that rates more accurately than before should rise;
- GROUPS FOR FAIRNESS—if the nodes rated by v can be divided into k groups such that each node in a particular group is rated by v in the same way, then the fairness of v should be equal to the *weighted average* of v's fairness in a setting where v rates these groups separately;
- OBVIOUS FAIRNESS METRIC—here, we stipulate that when a node makes maximal errors when rating all of its neighbors, then its fairness should be 0, and when there is no error, then the fairness is 1;
- BASELINE FOR FAIRNESS—the fairness of a node that rates noone is 1.

The above axioms uniquely define the fairness part of the FGA function. In summary, all the above axioms uniquely define the FGA function.

Theorem 1. *The SMOOTH GOODNESS, INCREASE WEIGHT, MONOTONICITY FOR GOODNESS, MAXIMAL TRUST, GROUPS FOR GOODNESS, BASELINE FOR GOODNESS axioms and the SMOOTH FAIRNESS, MONOTONICITY FOR FAIRNESS , OBVIOUS FAIRNESS METRIC, GROUPS FOR FAIRNESS, and BASELINE FOR FAIRNESS axioms uniquely define the FGA function.*

Complexity of Attack

Let us now study the complexity of manipulating FGA.

Attack models Given $G = (V, E, \omega(E))$, let $A \subseteq V$ be a set of attackers. We define two types of the A's objectives:

- **targeting potential links** — here, the target set TP is composed of disconnected pairs of nodes from $V \setminus A$:
  $$TP \subseteq \{\{u, v\} : u, v \in V \setminus A \wedge (u, v), (v, u) \notin E\}\,. \quad (3)$$
  Intuitively, the aim is to change the predicted weight of the potential links between the pairs from TP.
- **targeting nodes** — here, the target set is $T \subseteq V \setminus A$. Intuitively, the goal is to alter the targets' reputation.

The attackers can make the following types of moves:

- **edge addition** — the attackers can add an edge (u, v) to G, where $u \in A$, $v \in T$, $(u, v) \notin E$, and with the weight $\omega(u, v) \in [-1, 1]$. This corresponds to the attacker $u \in A$ rating node $v \in T$ for the first time.
- **weight update** — an attacker $u \in A$ can update the weight of an existing edge $(u, v) \in G$ to some value $\omega(u, v) \in [-1, 1]$. This corresponds to a modification of the existing rating by the attacker.

Algorithm 1: Direct attack

```
Data: $A, T = \{t\}, G$
for $a \in A$ do
    add an edge $(a, t)$, $\omega(a, t) = -1$ to the graph $G$
end
```

All the attackers are allowed to make no more than k such moves in total. We will refer to k as a budget.[1]

We will now formalize our computational problems. In the first one, the attackers aim at modifying the predicted weights between the pairs of nodes in TP to decrease them below (increase above) a certain threshold. This attack corresponds to breaking potential business connections.

Problem 1 (DECREASE (INCREASE) MUTUAL TRUST, DMT (IMT))**.** *Given a weighted signed network $G = (V, E, \omega)$, a set of attacking nodes $A \subseteq V$, a target set of disconnected pairs of nodes TP as defined in eq. 3, an intermediary set $I \subseteq V$, the budget k, and a threshold $t \in [-1, 1]$, decide for all $\{u, v\} \in TP$ whether it is possible to decrease (increase) the value of either predicted weight $f(u) \times g(v)$ or $f(v) \times g(u)$ to or below (above) the threshold t by making no more than k edge additions or weight updates with the restriction that the attackers $u \in A$ are rating only the nodes from the intermediary set I.*

In the second problem, the attackers aim at altering the goodness value of the nodes from a target set T. This attack corresponds to spoiling the reputation of the target nodes.

Problem 2 (DECREASE (INCREASE) NODES RATING, DNR (INR))**.** *Given WSN $G = (V, E, \omega)$, a set of attackers $A \subseteq V$, a target set $T \subseteq V \setminus A$, an intermediary set $I \subseteq V$, the number of possible moves k, and threshold $t \in [-1, 1]$, decide whether it is possible, for all $v \in T$, to decrease (increase) the goodness of each vertex v to or below (above) threshold t by making no more than k edge additions or weight updates with the restriction that the attackers $u \in A$ are rating only the nodes from the intermediary set I.*

Hardness Results We first consider DMT (IMT).

Theorem 2. *Solving the $DMT(IMT) = (G = (V, E, \omega)$, A, $TP, I, t, k)$ problem is NP-hard.*

Theorem 3. *Solving the $DNR(INR) = (G = (V, E, \omega)$, $A, T, I, t, k)$ problem is NP-hard.*

Proof of the above theorems can be found in the full version of the paper.

Parametrized complexity The following results, in terms of the W-hierarchy for the parameterized algorithms (Cygan et al. 2015), hold:

Theorem 4. *$DNR(INR)$ parameterized by k is $W[2]$-hard.*

Theorem 5. *$DMT(IMT)$ parameterized by k is $W[2]$-hard.*

Proof of the above theorems can be found in the full version of the paper.

Algorithm 2: Indirect attack

```
Data: $A, T = \{t\}, G$
sort nodes in $A$ by their fairness score
for $a \in sorted(A)$ do
    $N_1 \leftarrow Pred(t)$
    find a neighbor $n_2 \in Succ(n_1) \setminus \{t\}$ of a neighbor
    $n_1 \in N_1$ that minimizes the goodness value of $t$, when
    adding an edge $(a, n_2)$ with weight $\omega(a, n_2) = 1$ or
    $\omega(a, n_2) = -1$
    add the edge to the graph $G$
end
```

Manipulating a Node Directly

Let us now focus on attacks on attacking a single node. First, we report a result on the scale of manipulability of the FGA function by a direct attack. In particular, the following theorem says that it does not take many edges to change the sign of a single node in the DNR problem when the attacker is able to rate the target directly.

Theorem 6. *Let us consider an instance of the DNR problem, where, for an arbitrary $G = (V, E, \omega)$, a single node u_T is attacked with $0 < g(u_T) \leq 1$ (thus $T = \{u_T\}$), and the set of attacking nodes $A \subseteq V$ is relatively trusted, $f(v) \geq \frac{1}{2}$ for $v \in A$ and $|A| > \lceil 2 \times g(u_T) \times indeg(u_T) \rceil$. Then, it is trivial to change the sign of the goodness value of u_T (i.e. to achieve the threshold $t = 0$) if the attackers can attack directly (i.e. $T \subseteq I$ in the DNR problem).*

The proof can be found in the full version of the paper.

Bounding the Strength of Indirect Attacks

In this section, we give bounds on the strength of an indirect *Sybil attacks*, i.e., the attack in which the attacker creates a new node when adding a new edge.[2] Our results hold for a family of relatively dense networks, $G = (V, E, \omega)$, in which every node has a lower bound on its indegree and outdegree, i.e., $\forall_{v \in V} indeg(v) \geq k \ \ \& \ \ outdeg(v) \geq k$, and the intermediary nodes, $j \in V$, are relatively weakly rated, i.e. $\sum_{v \in Pred(j)} |\omega_{vj}| \leq k$. We call such networks *minimum-k-neighbour networks*.

Theorem 7 (Indirect Sybil attack)**.** *Assume a WSN $G = (V, E, \omega)$, where a new Sybil node s_i is added that rates some intermediary node $i \neq t$. Whenever $\forall_{v \in V} indeg(v) \geq k \ \ \& \ \ outdeg(v) \geq k$, and $\forall_{j \in V} \sum_{v \in Pred(j)} |\omega_{vj}| \leq k$, then $|\Delta g(t)| \leq \frac{2}{(indeg(i)+1) \times k}$.*

Proof. Let us define set V' as follows. We begin with $V' = \{t\}$. Next, we iteratively add to V' other nodes $v \in V$ which are indirectly connected to at least one node in V' (i.e., $\exists v' \in V' : \exists (l, v'), (l, v) \in E$). It is easy to see that the intermediary node i has to belong to V' in order to make the indirect attack successful.

[1] We place no constraints on how the attackers distribute this budget among themselves. In an extreme case, a single attacker can do all k actions.

[2] Our analysis also provides some additional bound for a direct Sybil attack.

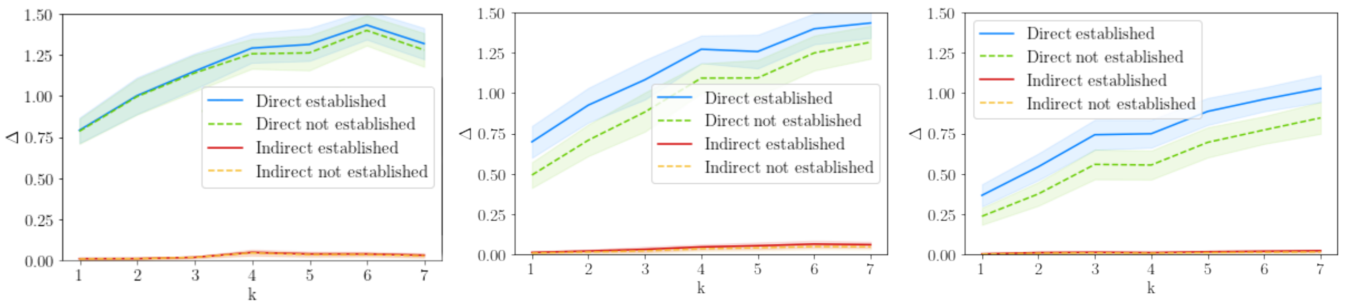


Figure 3: The comparison of direct/indirect, established/non-established attacks for Bitcoin OTC, Bitcoin Alpha and RFA Net networks.

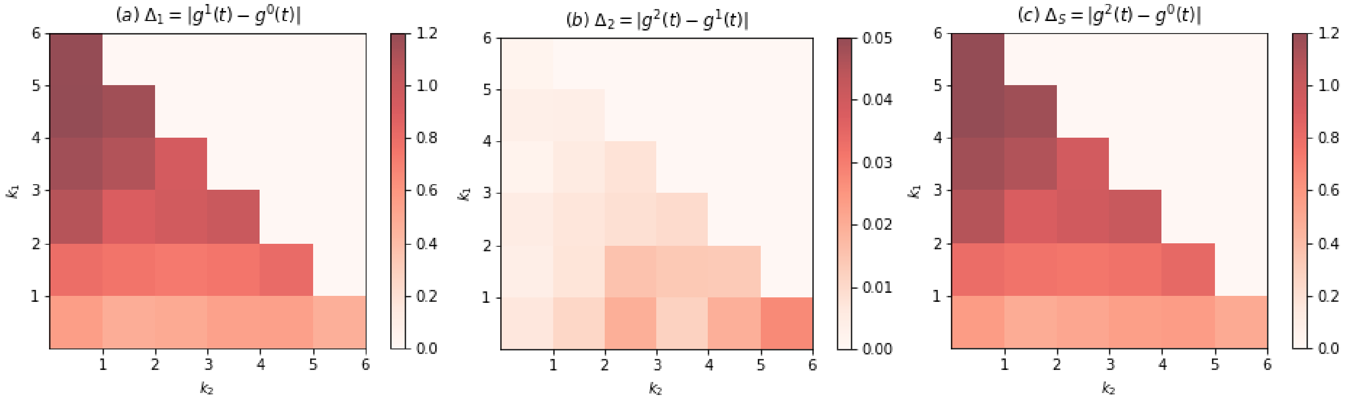


Figure 4: The results for the mixed settings in Bitcoin OTC. The average strength of an indirect attack is small and significantly smaller that the average strength of a direct attack. Δ_1 shows the influence of the attack with k_1 direct edges, Δ_2 shows the influence of the attack with k_2 indirect edges, Δ_s shows the influence of the attack with k_1 direct edges and k_2 indirect edges.

We denote by $g(l)/f(l)$ the goodness/fairness value of the node l before the Sybil attack, and by $g'(l)/f'(l)$ the goodness/fairness value after the attack. Let $\Delta g(l) = g'(l) - g(l)$ and $\Delta f(l) = f'(l) - f(l)$.

For the target node, $t \in V'$, we can calculate how its $g(t)$ changes w.r.t. the changes introduced to the goodness of all other nodes. Here, we assume that the Sybil attack is indirect, i.e., the Sybil edge is not added to t.

$$g(t) = \frac{1}{indeg(t)} \sum_{u\in Pred(t)} f(u) \times \omega(u,t) = \frac{1}{indeg(t)} \times \sum_{u\in Pred(t)} \omega(u,t) \times \left[1 - \frac{1}{outdeg(u)} \sum_{v\in Succ(u)} \frac{|\omega(u,v) - g(v)|}{2}\right]$$

Thus, from the triangle inequality:

$$|\Delta g(t)| \le \frac{1}{2 \times indeg(t)} \sum_{u\in Pred(t)} \sum_{v\in Succ(u)} \frac{1}{outdeg(u)} \times |\omega(u,t)| \times |\Delta g(v)| \le \frac{1}{2 \times indeg(t)} \sum_{v:\exists(t,u)\in E \ \&\ (u,v)\in E} \sum_{u\in Succ(v)} \frac{|\Delta g(v)|}{outdeg(v)}$$

And because $indeg(t) \ge k$, then:

$$|\Delta g(t)| \le \frac{\sum_{v:\exists(t,u)\in E \ \&\ (u,v)\in E} |\Delta g(v)|}{2 \times k}. \tag{4}$$

Let us calculate $\Delta g(l)$ for all $l \in V'$. We can see that whenever we introduce a new node, s_i, that aims at node i in the network, then:

$$|\Delta g(i)| = \left| \frac{-1 + \sum_{u\in Pred(i)} f'(u) \times \omega(u,i)}{indeg(i)+1} - \frac{\sum_{u\in Pred(i)} f(u) \times \omega(u,i)}{indeg(i)} \right| \le \left| \frac{\sum_{u\in Pred(i)} \Delta f(u) \times \omega(u,i)}{indeg(i)} - \frac{1}{indeg(i+1)} - \sum_{u\in Pred(i)} \frac{f'(u) \times \omega(u,i)}{indeg(i)(indeg(i)+1)} \right| \le \left| \frac{\sum_{u\in Pred(i)} \Delta f(u) \times \omega(u,i)}{indeg(i)} \right| + \frac{2}{indeg(i)+1}$$

For the other nodes, $l \in V'$, that are not targeted by s_i:

$$|\Delta g(l)| = \left| \frac{\sum_{u\in Pred(l)} \Delta f(u) \times \omega(u,l)}{indeg(l)} \right|$$

For all $l \in V'$, we can write:

$$\Big|\sum_{u\in Pred(l)} \frac{\Delta f(u)\times \omega(u,l)}{indeg(l)}\Big| \leq$$

$$\frac{1}{2\times indeg(l)} \sum_{v\in Pred(l), u\in Succ(v)\setminus\{l\}} \frac{|\omega_{vl}|\times|\Delta g(v)|}{outdeg(v)} =$$

$$\frac{1}{2\times indeg(l)} \sum_{i\in V}\sum_{(v,l),(v,i)\in E} \frac{|\omega_{vi}|}{outdeg(v)}|\Delta g(i)|$$

In the matrix form, we thus have:

$$Q \leq M\times Q + \begin{bmatrix}0 & \frac{2}{indeg(i)+1} & 0 & 0 & 0\end{bmatrix}^T,$$

where Q is a vector of length $|V'|$ which on the l'th position has $\Delta g(l)$ for $l \in V'$. And M is a matrix of size $|V'|\times|V'|$, and its coefficients are filled according to Equation 5. Note that:

$$\frac{1}{2\times indeg(l)} \sum_{v\in Pred(l), u\in Succ(v)\setminus\{l\}} \frac{|\omega_{vl}|}{outdeg(v)} \leq \frac{1}{2} \tag{5}$$

This implies that $||M||_\infty \leq \frac{1}{2}$. On the other hand, for a given column j in the matrix M:

$$\sum_{l\in V}\frac{1}{2\times indeg(l)} \sum_{(v,l),(v,j)\in E} \frac{|\omega_{vj}|}{outdeg(v)} \leq$$

$$\frac{1}{2k}\sum_{l\in V}\sum_{(v,l),(v,j)\in E}\frac{|\omega_{vj}|}{outdeg(v)} \leq \frac{1}{2k}\sum_{v\in Pred(j)}|\omega_{vj}| \leq \frac{1}{2}$$

whenever $\sum_{v\in Pred(j)}|\omega_{vj}| \leq k$, which implies that $||M||_1 \leq \frac{1}{2}$.

The values of $\Delta g(l)$ achieve maximum when:

$$Q = M\times Q + \begin{bmatrix}0 & \frac{2}{indeg(i)+1} & 0 & 0 & 0\end{bmatrix}^T$$

But in this case, we can solve the equation system with:

$$Q = \frac{1}{I-M}\times\begin{bmatrix}0 & \frac{2}{indeg(i)+1} & 0 & 0 & 0\end{bmatrix}^T$$

Matrix M is indeed invertible due to appropriately selected nodes $l \in V'$. What is more, since $||M||_\infty \leq \frac{1}{2}$, then we can write $\frac{1}{I-M} = I+M+M^2+\ldots$ (Turnbull 1930). Finally, the above quality and $||M||_1 \leq \frac{1}{2}$ imply that $|\sum_{l\in V}\Delta g(l)| \leq \frac{4}{indeg(i)+1}$.

Now, because Equation 4 holds for the target node, t, and $|\sum_{l\in V}\Delta g(l)| \leq \frac{4}{indeg(i)+1}$, then:

$$|\Delta g(t)| \leq \frac{2}{(indeg(i)+1)\times k}.$$

□

The above theorem shows that in a minimum-k-neighbour network, the indirect attack is at least k times weaker than the direct attack. That is, when we modify the goodness value of some node i by Δ, then the value of the target node t is modified by at most $\frac{\Delta}{k}$.

Building upon the above reasoning, we can show that the following result also holds (the proof in the full version of the paper).

Theorem 8 (Direct Sybil attack)**.** *Assuming in a WSN $G = (V, E, \omega)$ where one adds a new Sybil node rating directly some target node t, then the goodness value of the target node t decreases by at most $|\Delta g(t)| \leq \frac{2}{indeg(t)}$*

Simulations

We conduct a series of simulations on the Bitcoin OTC, Bitcoin Alpha, and RFA Net datasets studied by Kumar et al. (2016). They consist of weighted signed networks with $|V| =\geq 3,700$, $|E| \geq 24,000$ each, where the proportion of positively weighted edges is $\geq 84\%$. A vast majority of the nodes in each network, i.e., more than 76%, have an indegree up to 10. Furthermore, most of the users in the networks are evaluated as fair by the FGA function—$f(v) \geq 0.7$ for 100% of the nodes (with the mean $f(v)$ equal to 0.94). As for goodness, only less than 4% of users have a strongly positive score of more than 0.5, and in the Bitcoin OTC network 8% of users have negative score of less than -0.3, whereas in Bitcoin Alpha $3,8\%$ have goodness below -0.3.

We focus on the attacks that lower the *goodness* of the nodes, as in the *DNR* problem. In particular, each experiment was conducted on the set of attacking nodes A of size $k = \{1,\ldots,7\}$ and the target set $T = \{t\}$ of size 1. The target, $t \in T$, was chosen randomly from those nodes that have relatively high goodness ($g(t) \geq 0.50$) and a low indegree ($0 < indeg(t) < 10$). We study two types of the attackers:

- **not-established** attackers — chosen from relatively newly created nodes with $0 < indeg < 10$ and $outdeg = 0$). This allows for studying Sybil-style attacks; and
- **established** attackers — chosen from the nodes with $outdeg(v) > 5$ (and iteratively choosing nodes with fairness $f(v) > 0.7$). This allows for studying attacks by the nodes whose standing in the network has been built for some time.

We simulate three types of attacks:

- **direct attacks** — a set of attackers A of size k rates directly the target node $t \in T$. The pseudocode is presented in Algorithm 1. Each attacker rates t using weight -1;
- **indirect attacks** — the attackers set A of size k rates the neighbors of the neighbors of the target node, to minimize the goodness part of the FGA of the target node by manipulating *fairness* of the targets' neighbors. The pseudocode of the attack is presented in the Alorithm 2. More precisely, the algorithm implements a greedy approach, where each new edge is used to minimize the *goodness* of the target node t by minimizing (or maximizing) the *fairness* of one of the targets' neighbors by directly rating the successor of the target's neighbor with an edge of weight 1 or -1. The algorithm performs calculations iteratively on the attackers sorted by the value of their fairness value.
- **mixed attack** — k_1 attacking nodes perform a direct attack and k_2 perform an indirect one, where $k_1 + k_2 = k$.

The results in Figure 3 are presented with a 95% confidence interval (marked with the opaque region around the solid/dashed lines). They show how a direct/indirect attack by established/not established nodes influences the goodness

Algorithm 3: Modified indirect attack

```
**Data:** $A, T = \{t\}, G$
sort nodes in $A$ by their fairness score
**while** $i < len(sorted(A))$ **do**
    $a \leftarrow sorted(A)[i]$
    $N_1 \leftarrow Pred(t)$
    find a neighbor $n_2 \in Succ(n_1) \setminus \{t\}$ of a neighbor $n_1 \in N_1$ that minimizes the goodness value of $t$, when adding an edge $(a, n_2)$ with weight $\omega(a, n_2) = 1$ or $\omega(a, n_2) = -1$
    $edges\_len \leftarrow min(SCALE * len(indegree(n_2), MAX, len(A) - i)$
    add $edges\_len$ edges to the graph $G$
    $i \leftarrow i + edges\_len$
**end**
```

	B. OTC	B. Alpha	RFA Net
max *indeg*	10	13	10
min *goodness*	0.8	0.5	0.5
num of samples	20	30	27
num of edges	20	20	20

Table 1: The parameters used to search weak target nodes in the test sets. “B.” stands for “Bitcoin”.

of the target node (Δ). For Bitcoin OTC and Bitcoin Alpha, and RFA Net in both cases (direct and indirect attacks), there is no significant difference between the established and not established results (solid and dashed lines).

In Figure 4 (see the full version for a bigger picture), we present results for a mixed setting. The individual cells of the heatmaps show: (a) Δ_1 — the absolute change of the goodness of the target node introduced by the k_1 direct edges; (b) Δ_2 — the absolute change of the goodness of the target node introduced by the k_2 indirect edges; and (c) Δ_S — the total change, i.e., $\Delta_S = \Delta_1 + \Delta_2$. The results show that the average strength of a direct attack varies between 0.2 and 1.2 for different k, and the average strength of the indirect attack is lower than 0.05, i.e., significantly smaller than the average strength of a direct attack. Theorem 6 proved in the full version of the paper provides some intuition why the direct attacks have such a strong impact, whereas in the full version we give another intuition why the indirect attack is weaker than the direct attack.

Better Heuristic for Indirect Attacks

We conduct additional tests to analyze the strength of the indirect attacks. In Algorithm 3, instead of adding only a single edge in each iteration, as in Algorithm 2, we add a series of new edges. In more detail, we add $SCALE = 5$ times more Sybil edges than the indegree of the target node (but at most some predefined maximum $MAX = 10$). We take this approach to scale up the effect of manipulating the goodness value of the target nodes.

We attack only nodes with bounded $indegree$, and the goodness value bigger than some threshold. We believe these nodes are more easily manipulable than an average node. See Table 1 for the details.

	B. OTC	B. Alpha	RFA Net
average change	0.081	0.085	0.030
standard deviation	0.089	0.085	0.028
min change	0.010	0.008	0.009
max change	0.300	0.298	0.131
median	0.053	0.042	0.021
0.75-quantile	0.111	0.121	0.041

Table 2: The results of Algorithm 3 on different datasets.

The analysis of the data in Table 2 shows that the attack using the Algorithm 3 may (but rarely does) achieve relatively strong results in some cases. To be more precise, the maximum change of the goodness value of the target node introduced by the indirect attack in the Bitcoin OTC and Bitcoin Alpha detasets reached the barrier of 0.3. This shows that in general networks (unlike the minimum-k-neighbour ones described in Theorem 7) do not have strong resistance property against indirect attacks. In most cases however the attack gives rather weak results (75%-quantile on all datasets is at most 0.12 with low median of at most 0.05). The minimum strength of the attack in all datasets achieves 0.01.

Conclusions

In this paper, we axiomatized the FGA measure with respect to, among others, the properties of homogeneously and unanimously rated nodes and with respect to the properties of the rating nodes that achieve constant rating error. Furthermore, we presented the hardness results on the manipulability problems. We also derived analytical results concerning the strength of the direct attacks and weakness of the indirect attacks in the networks in which each node has minimum k neighbours (in and out). Finally, we visualised experimentally the strength of direct attacks and analysed two different greedy algorithms for indirect attacks. This showed that FGA might be manipulated indirectly in non-minimum-k-neighbour networks. Overall, a higher-level insight from our analysis is that FGA is generally more difficult to manipulate compared to other social network analysis tools (e.g., centrality measures). In particular, while worst-case hardness results are common in the literature, various other tools turned out to be easily manipulable in practice by well-crafted heuristics (Bergamini et al. 2018; Waniek et al. 2018b, 2019, e.g.). The FGA measure turns out to be more resilient, which provides a good argument for using it in practice.

As a next step, we want to analyse the manipulability of other measures in the literature. We also encourage studies on the axiomatization of the ranking functions, which would result in a better understanding of their properties.

Acknowledgments

This research was supported by NCN Grant 2019/35/B/ST6/04138 and ERC Grant 885666.

References

Bergamini, E.; Crescenzi, P.; D'angelo, G.; Meyerhenke, H.; Severini, L.; and Velaj, Y. 2018. Improving the betweenness centrality of a node by adding links. *Journal of Experimental Algorithmics (JEA)*, 23: 1–5.

Cai, Y.; and Zhu, D. 2016. Fraud detections for online businesses: a perspective from blockchain technology. *Financial Innovation*, 2(1): 1–10.

Chen, J.; Chen, L.; Chen, Y.; Zhao, M.; Yu, S.; Xuan, Q.; and Yang, X. 2019. GA-based Q-attack on community detection. *IEEE Transactions on Computational Social Systems*, 6(3): 491–503.

Chen, L.; Li, J.; Peng, J.; Xie, T.; Cao, Z.; Xu, K.; He, X.; and Zheng, Z. 2020. A Survey of Adversarial Learning on Graphs. *arXiv*, arXiv–2003.

Crescenzi, P.; D'angelo, G.; Severini, L.; and Velaj, Y. 2016. Greedily improving our own closeness centrality in a network. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(1): 9.

Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edition. ISBN 3319212745.

Fionda, V.; and Pirro, G. 2017. Community deception or: How to stop fearing community detection algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 30(4): 660–673.

Kumar, S.; Spezzano, F.; Subrahmanian, V.; and Faloutsos, C. 2016. Edge weight prediction in weighted signed networks. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 221–230. IEEE.

Leskovec, J.; Huttenlocher, D.; and Kleinberg, J. 2010a. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, 641–650.

Leskovec, J.; Huttenlocher, D.; and Kleinberg, J. 2010b. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 1361–1370.

Mayzlin, D.; Dover, Y.; and Chevalier, J. 2014. Promotional reviews: An empirical investigation of online review manipulation. *American Economic Review*, 104(8): 2421–55.

Resnick, P.; Kuwabara, K.; Zeckhauser, R.; and Friedman, E. 2000. Reputation systems. *Communications of the ACM*, 43(12): 45–48.

Tang, J.; Chang, Y.; Aggarwal, C.; and Liu, H. 2016. A survey of signed network mining in social media. *ACM Computing Surveys (CSUR)*, 49(3): 1–37.

Turnbull, H. W. 1930. A Matrix Form of Taylor's Theorem. *Proceedings of the Edinburgh Mathematical Society*, 2(1): 33–54.

Waniek, M.; Michalak, T. P.; Wooldridge, M. J.; and Rahwan, T. 2018a. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2(2): 139.

Waniek, M.; Michalak, T. P.; Wooldridge, M. J.; and Rahwan, T. 2018b. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2(2): 139–147.

Waniek, M.; Zhou, K.; Vorobeychik, Y.; Moro, E.; Michalak, T. P.; and Rahwan, T. 2019. How to Hide one's Relationships from Link prediction Algorithms. *Scientific reports*, 9(1): 1–10.

Was, T.; Waniek, M.; Rahwan, T.; and Michalak, T. 2020. The Manipulability of Centrality Measures-An Axiomatic Approach. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 1467–1475. Auckland, New Zealand: AAMAS.