From acquaintance to best friend forever: robust and fine-grained inference of social-tie strengths

Florian Adriaens
Dept. of Electronics and Information Systems, IDLab, Ghent University
florian.adriaens@ugent.be

Tijl De Bie
Dept. of Electronics and Information Systems, IDLab, Ghent University
tijl.debie@ugent.be

Aristides Gionis
Dept. of Computer Science
Aalto University
aristides.gionis@aalto.fi

Jefrey Lijffijt
Dept. of Electronics and Information Systems, IDLab, Ghent University
jefrey.lijffijt@ugent.be

Polina Rozenshtein
Dept. of Computer Science
Aalto University
polina.rozenshtein@aalto.fi

ABSTRACT

Social networks often provide only a binary perspective on social ties: two individuals are either connected or not. While sometimes external information can be used to infer the *strength* of social ties, access to such information may be restricted or impractical. Sintos and Tsaparas (KDD 2014) first suggested to infer the strength of social ties from the topology of the network alone, by leveraging the *Strong Triadic Closure (STC)* property. The STC property states that if person A has strong social ties with persons B and C, B and C must be connected to each other as well (whether with a weak or strong tie). Sintos and Tsaparas exploited this property to formulate the inference of the strength of social ties as an **NP**-hard optimization problem, and proposed two approximation algorithms. We refine and improve this line of work, by developing a sequence of linear relaxations of the problem, which can be solved exactly in polynomial time. Usefully, these relaxations infer more fine-grained levels of tie strength (beyond strong and weak), which also allows to avoid making arbitrary strong/weak strength assignments when the network topology provides inconclusive evidence. One of the relaxations simultaneously infers the presence of a limited number of STC violations. An extensive theoretical analysis leads to two efficient algorithmic approaches. Finally, our experimental results elucidate the strengths of the proposed approach, and sheds new light on the validity of the STC property in practice.

KEYWORDS

Strong Triadic Closure, strength of social ties, Linear Programming, convex relaxations, half-integrality

1 INTRODUCTION

Online social networks, such as Facebook, provide unique insights into the social fabric of our society. They form an unprecedented resource to study social-science questions, such as how information propagates on a social network, how friendships come and go, how echo chambers work, how conflicts arise, and much more.

Yet, many social networks provide a black-and-white perspective on friendship: they are modeled by unweighted graphs, with an edge connecting two nodes representing that two people are friends. Surely though, some friendships are stronger than others, and clearly, in studying social phenomena understanding the strength of social ties can be critical.

Although in some cases detailed data are available and can be used for inferring the strength of social ties, e.g., communication frequency between users, or explicit declaration of relationship types, such information may not always be available.

The question of whether the strength of social ties can be inferred *from the structure of the social network alone*, the subject of the current paper, is therefore an important one.

Background. An important line of research attempting to address the inference of the strength of social ties is based on the *strong triadic closure* (STC) property from sociology, introduced by Georg Simmel in 1908 [13]. To understand the STC property, consider an undirected network $G = (V, E)$, with $E \subseteq \binom{V}{2}$. Consider additionally a *strength function* $w : E \rightarrow \{\texttt{weak}, \texttt{strong}\}$ assigning a binary strength value to each edge. A triple of connected nodes $i, j, k \in V$ is said to satisfy the STC property, with respect to the strength function w, if $w(\{i, j\}) = w(\{i, k\}) = \texttt{strong}$ implies $\{j, k\} \in E$. In other words, two adjacent strong edges always need to be closed by an edge (whether weak or strong). We refer to a strength function for which all connected triples satisfy the STC property as *STC-compliant*:

Definition 1.1 (STC-compliant strength function on a network). A *strength function* $w : E \rightarrow \{\texttt{weak}, \texttt{strong}\}$ is STC-compliant on an undirected network $G = (V, E)$ if and only if

$$\text{for all } i, j, k \in V, \{i, j\}, \{i, k\} \in E :$$
$$w(\{i, j\}) = w(\{i, k\}) = \texttt{strong} \text{ implies } \{j, k\} \in E.$$

A consequence of this definition is that for an STC-compliant strength function, any *wedge*—defined as a triple of nodes $i, j, k \in V$ for which $\{i, j\}, \{i, k\} \in E$ but $\{j, k\} \notin E$—can include only one strong edge. We will denote such a wedge by the pair $(i, \{j, k\})$, where i is the root and $\{j, k\}$ are the end-points of the wedge, and denote the set of wedges in a given network by $\mathcal{W}$.

On the other hand, for a *triangle*—defined as a triple of nodes $i, j, k \in V$ for which $\{i, j\}, \{i, k\}, \{j, k\} \in E$—no constraints are implied on the strengths of the three involved edges. We will denote a triangle simply by the (unordered) set of its three nodes $\{i, j, k\}$, and the set of all triangles in a given network as $\mathcal{T}$.

Relying on the STC property, Sintos and Tsaparas [14] propose an approach to infer the strength of social ties. They observe that a strength function that labels all edges as weak is always STC-compliant. However, as a large number of strong ties is expected to be found in a social network, they suggest searching for a strength function that maximizes the number of strong edges, or (equivalently) minimizes the number of weak edges.

To write this formally, we introduce a variable w_{ij} for each edge $\{i, j\} \in E$, defined as $w_{ij} = 0$ if $w(\{i, j\}) = \texttt{weak}$ and $w_{ij} = 1$ if $w(\{i, j\}) = \texttt{strong}$. Then, the original STC problem, maximizing the number of strong edges, can be formulated as:

$$\max_{w_{ij}:\{i,j\}\in E} \sum_{\{i,j\}\in E} w_{ij}, \tag{STCmax}$$

$$\text{such that } w_{ij} + w_{ik} \leq 1, \qquad \text{for all } (i, \{j, k\}) \in \mathcal{W}, \tag{1}$$

$$w_{ij} \in \{0, 1\}, \qquad \text{for all } \{i, j\} \in E. \tag{2}$$

Equivalently, one could instead minimize $\sum_{\{i,j\}\in E}(1 - w_{ij})$ subject to the same constraints, or with transformed variables $v_{ij} = 1 - w_{ij}$ equal to 1 for weak edges and 0 for strong edges:

$$\min_{v_{ij}:\{i,j\}\in E} \sum_{\{i,j\}\in E} v_{ij}, \tag{STCmin}$$

$$\text{such that } v_{ij} + v_{ik} \geq 1, \qquad \text{for all } (i, \{j, k\}) \in \mathcal{W}, \tag{3}$$

$$v_{ij} \in \{0, 1\}, \qquad \text{for all } \{i, j\} \in E. \tag{4}$$

When we do not wish to distinguish between the two formulations, we will refer to them jointly as STCbinary.

Sintos and Tsaparas [14] observe that STCmin is equivalent to Vertex Cover on the so-called *wedge graph* $G_E = (E, F)$, whose nodes are the edges of the original input graph G, and whose edges are $F = \{(\{i, j\}, \{i, k\}) \mid (i, \{j, k\}) \in \mathcal{W}\}$, i.e., two nodes of G_E are connected by an edge if the edges they represented in G form a wedge. While Vertex Cover is **NP**-hard, a simple factor-2 approximation algorithm can be adopted for STCmin. On the other hand, STCmax is equivalent to finding the *maximum independent set* on the wedge graph G_E, or equivalently the *maximum clique* on the *complement* of the wedge graph. It is known that there cannot be a polynomial-time algorithm that for every real number $\varepsilon > 0$ approximates the maximum clique to within a factor better than $O(n^{1-\varepsilon})$ [6]. In other words, while a polynomial-time approximation algorithm exists for minimizing the number of weak edges (with approximation factor two), no such polynomial-time approximation algorithm exists for maximizing the number of strong edges.

Despite its novelty and elegance, STCbinary suffers from a number of weaknesses, which we address in this paper.

First, STCbinary is an **NP**-hard problem. Thus, one has to either resort to approximation algorithms, which are applicable only for certain problem variants—see the discussion on STCmin vs. STCmax above—or rely on exponential algorithms and hope for good behavior in practice. Second, the problem returns *only binary edge strengths*, weak vs. strong. In contrast, real-world social networks contain tie strengths of many different levels. A third limitation is that, on real-life networks, STCbinary tends to have many optimal solutions. Thus, any such optimal solution makes *arbitrary strength assignments* for the edges where different optimal solutions differ from each other.[1] Last but not least, STCbinary *assumes that the STC property holds for all wedges*. Yet, real-world social networks tend to be noisy, with spurious connections as well as missing edges.

Contributions. In this paper we propose a series of linear programming relaxations that address all of the above limitations of STCbinary. In particular, our LP relaxations provide the following advantages.

- The first relaxation replaces the integrality constraints $w_{ij} \in \{0, 1\}$ with fractional counterparts $0 \leq w_{ij} \leq 1$. It can be shown that this relaxed LP is *half-integral*, i.e., the edge strengths in the optimal solution take values $w_{ij} \in \{0, \frac{1}{2}, 1\}$. Thus, not only the problem becomes polynomial, but the formulation introduces meaningful *three-level* social strengths.
- Next we relax the upper-bound constraint, requiring only $w_{ij} \geq 0$, while generalizing the STC property to deal with higher gradations of edge strengths. We show that the optimal edge strengths still take values in a small discrete set. Thus, our approach can yield multi-level edge strengths, from a small set of discrete values, while ensuring a polynomial algorithm.
- We show how the previous relaxations can be solved by advanced and highly efficient combinatorial algorithms, so that one need not rely on generic LP solvers.
- As our relaxations allow intermediate strength levels, arbitrary choices between weak and strong values can be avoided by assigning an intermediate strength.
- Our final relaxation simultaneously edits the network while optimizing the edge strengths, making it robust against noise in the network. Also this variant has no integrality constraints, and thus, it can again be solved in polynomial time.

Outline. We start by proposing the successive relaxations in Sec. 2. In Sec. 3 we analyse these relaxations and derive properties of their optima, highlighting the benefits of these relaxations with respect to STCbinary. The theory developed in Sec. 3 leads to efficient algorithms, discussed in Sec. 4. Empirical performance is evaluated in Sec. 5 and related work is reviewed in Sec. 6, before drawing conclusions in Sec. 7.

2 LP RELAXATIONS

Here we will derive a sequence of increasingly loose relaxations of Problem STCmax. Their detailed analysis is deferred to Sec. 3.[2]

2.1 Elementary relaxations

In this subsection we simply enlarge the feasible set of strengths w_{ij}, for all edges $\{i, j\} \in E$. This is done in two steps.

2.1.1 Relaxing the integrality constraint. The first relaxation relaxes the constraint $w_{ij} \in \{0, 1\}$ to $0 \leq w_{ij} \leq 1$. Denoting the set of edge strengths with $\mathbf{w} = \{w_{ij} \mid \{i, j\} \in E\}$, this yields:

$$\max_{\mathbf{w}} \sum_{\{i,j\}\in E} w_{ij}, \tag{LP1}$$

$$\text{such that } w_{ij} + w_{ik} \leq 1, \qquad \text{for all } (i, \{j, k\}) \in \mathcal{W}, \tag{5}$$

$$w_{ij} \geq 0, \qquad \text{for all } \{i, j\} \in E, \tag{6}$$

$$w_{ij} \leq 1, \qquad \text{for all } \{i, j\} \in E. \tag{7}$$

Clearly, this relaxation will lead to solutions that are not necessarily binary. However, as will be explained in Sec. 3, Problem LP1 is

[1] A case in point is a star graph, where the optimal solution contains one strong edge (arbitrarily selected), while all others are weak.

[2] Our relaxations can also be applied to Problem STCmin, however, for brevity, hereinafter we omit discussion on this minimization problem.

half-integral, meaning that there always exists an optimal solution with values $w_{ij} \in \{0, \frac{1}{2}, 1\}$ for all $\{i,j\} \in E$.

2.1.2 Relaxing the upper bound constraints to triangle constraints. We now further relax Problem LP1, so as to allow for edge strengths larger than 1. The motivation to do so is to allow for higher gradations in the inference of edge strengths.

Simply dropping the upper-bound constraint (7) would yield uninformative unbounded solutions, as edges that are not part of any wedge would be unconstrained. Thus, the upper-bound constraints cannot simply be deleted; they must be replaced by looser constraints that bound the values of edge strengths in triangles in the same spirit as the STC constraint does for edges in wedges.

To do so, we propose to generalize the wedge STC constraints (5) to STC-like constraints on triangles, as follows: *in every triangle, the combined strength of two adjacent edges should be bounded by an increasing function of the strength of the closing edge.* In social-network terms: the stronger a person's friendship with two other people, the stronger the friendship between these two people must be. Encoding this intuition as a linear constraint yields:

$$w_{ij} + w_{ik} \le c + d \cdot w_{jk},$$

for some $c, d \in \mathbb{R}^+$. This is the most general linear constraint that imposes a bound on $w_{ij} + w_{ik}$ that is increasing with w_{jk}, as desired. We will refer to such constraints as *triangle constraints*.

In sum, we relax Problem LP1 by first adding the triangle constraints for all triangles, and subsequently dropping the upper-bound constraints (7). For the resulting optimization problem to be a *relaxation* of Problem LP1, the triangle constraints must be satisfied throughout the original feasible region. This is the case as long as $c \ge 2$: indeed, then the box constraints $0 \le w_{ij} \le 1$ ensure that the triangle constraint is always satisfied. The tightest possible relaxation is thus achieved with $c = 2$, yielding the following relaxation:

$$\max_{\mathbf{w}} \sum_{\{i,j\} \in E} w_{ij}, \tag{LP2}$$

$$\begin{aligned}
\text{such that } & w_{ij} + w_{ik} \le 1, && \text{for all } (i, \{j,k\}) \in \mathcal{W}, \\
& w_{ij} + w_{ik} \le 2 + d \cdot w_{jk}, && \text{for all } \{i,j,k\} \in \mathcal{T}, \quad (8) \\
& w_{ij} \ge 0, && \text{for all } \{i,j\} \in E.
\end{aligned}$$

Remark 1 (The wedge constraint is a special case of the triangle constraint). *Considering an absent edge as an edge with negative strength $-1/d$, the wedge constraint can in fact be regarded as a special case of the triangle constraint.*

2.2 Enhancing robustness by allowing edge additions and deletions

As noted earlier, although the STC property is theoretically motivated, real-world social networks are noisy and may contain many exceptions to this rule. In this subsection we propose two further relaxations of Problem LP2 that gracefully deal with exceptions of two kinds: wedges where the sum of edges strengths exceeds 1, and edges with a negative edge strength, indicating that the STC property would be satisfied should the edge not be present.

These relaxations thus solve the STC problem while allowing a small number of edges to be added or removed from the network.

2.2.1 Allowing violated wedge STC constraints. In order to allow for violated wedge STC constraints, we can simply add positive *slack variables* ϵ_{jk} for all $(i, \{j,k\}) \in \mathcal{W}$:

$$w_{ij} + w_{ik} \le 1 + \epsilon_{jk}, \quad \epsilon_{jk} \ge 0. \tag{9}$$

Elegantly, the slacks ϵ_{jk} can be interpreted as quantifying the strength of the (absent) edge between j and k. To show this, let $\bar{E}$ denote the set of pairs of end-points of all the wedges in the graph, i.e., $\bar{E} = \{\{j,k\} \mid \text{there exists } i \in V : (i, \{j,k\}) \in \mathcal{W}\}$. We also extend our notation to introduce strength values for those pairs, i.e., $\mathbf{w} = \left\{w_{ij} \mid \{i,j\} \in E \text{ or } \{i,j\} \in \bar{E}\right\}$, and define $w_{jk} \triangleq \frac{\epsilon_{jk}-1}{d}$ for $\{j,k\} \in \bar{E}$. The relaxed wedge constraints (9) are then formally identical to the triangle STC constraints (8). Meanwhile, the lower bound $\epsilon_{jk} \ge 0$ from (9) implies $w_{jk} \ge -\frac{1}{d}$, i.e., allowing the strength of these absent edges to be negative.

In order to bias the solution towards few violated wedge constraints a term $-C \sum_{\{j,k\} \in \bar{E}} w_{jk}$ is added to the objective function. The larger the parameter C, the more a violation of a wedge constraint will be penalized. The resulting problem is:

$$\max_{\mathbf{w}} \sum_{\{i,j\} \in E} w_{ij} - C \sum_{\{j,k\} \in \bar{E}} w_{jk}, \tag{LP3}$$

$$\begin{aligned}
\text{such that } & w_{ij} + w_{ik} \le 2 + d \cdot w_{jk}, && \text{for all } (i, \{j,k\}) \in \mathcal{W}, \\
& w_{ij} + w_{ik} \le 2 + d \cdot w_{jk}, && \text{for all } \{i,j,k\} \in \mathcal{T}, \\
& w_{ij} \ge 0, && \text{for all } \{i,j\} \in E. \\
& w_{jk} \ge -\frac{1}{d}, && \text{for all } \{j,k\} \in \bar{E}.
\end{aligned}$$

Note that in Remark 1, $-\frac{1}{d}$ was argued to correspond to the strength of an absent edge. Thus, the lower-bound constraint on w_{jk} requires these weights to be at least as large as the weight that signifies an absent edge. If it is strictly larger, this may suggest that the edge is in fact missing, as adding it increases the sum of strengths in the objective more than the penalty paid for adding it.

2.2.2 Allowing negative edge strengths. The final relaxation is obtained by allowing edges to have negative strength, with lower bound equal to the strength signifying an absent edge:

$$\max_{\mathbf{w}} \sum_{\{i,j\} \in E} w_{ij} - C \sum_{\{j,k\} \in \bar{E}} w_{jk}, \tag{LP4}$$

$$\begin{aligned}
\text{such that } & w_{ij} + w_{ik} \le 2 + d \cdot w_{jk}, && \text{for all } (i, \{j,k\}) \in \mathcal{W}, \\
& w_{ij} + w_{ik} \le 2 + d \cdot w_{jk}, && \text{for all } \{i,j,k\} \in \mathcal{T}, \\
& w_{ij} \ge -\frac{1}{d}, && \text{for all } \{i,j\} \in E. \\
& w_{jk} \ge -\frac{1}{d}, && \text{for all } \{j,k\} \in \bar{E}.
\end{aligned}$$

This formulation allows the optimization problem to strategically delete some edges from the graph, if doing so allows it to increase the sum of all edge strengths.

3 THEORETICAL ANALYSIS OF THE OPTIMA

The general form of relaxation LP1 is a well-studied problem, and it is known that there always exists a half-integral solution—a solution where all $w_{ij} \in \{0, \frac{1}{2}, 1\}$ [10]. In this section we demonstrate and exploit the existence of symmetries in the optima to show an

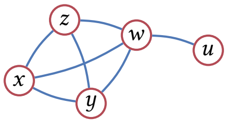


Figure 1: A toy graph illustrating the different type of edges defined in Section 3.1.

analogous result for Problem LP2. Furthermore, the described symmetries also exist for Problems LP3 and LP4, although they do not imply an analogue of the half-integrality result for these problems.

We also discuss how the described symmetries are useful in reducing the arbitrariness of the optima, as compared to Problems STCmax and STCmin, where structurally-indistinguishable edges might be assigned different strengths at the optima. Furthermore, in Sec. 4 we will show how the symmetries can be exploited for algorithmic performance gains, as well.

We start by giving some useful definitions and lemmas. Due to space limitations, the proofs of all results in this section are referred to an extended technical report [1].

3.1 Auxiliary definitions and results

It is useful to distinguish two types of edges:

Definition 3.1 (Triangle edge and wedge edge). A *triangle edge* is an edge that is part of at least one triangle, but that is part of no wedge. A *wedge edge* is an edge that is part of at least one wedge.

These definitions are illustrated in a toy graph in Figure 1, where edges (x, y), (y, z), and (x, z) are triangle edges, while edges (w, x), (w, y), (w, z), and (w, u) are wedge edges.

It is clear that in this toy example the set of triangle edges forms a clique. This is in fact a general property of triangle edges:

Lemma 3.2 (Subgraph induced by triangle edges). *Each connected component in the edge-induced subgraph, induced by all triangle edges, is a clique.*

Thus, we can introduce the notion of a *triangle clique.*

Definition 3.3 (Triangle cliques). The connected components in the edge-induced subgraph induced by all triangle edges are called *triangle cliques.*

The nodes $\{x, y, z\}$ in Figure 1 form a triangle clique. Note that not every clique in a graph is a triangle clique. E.g., nodes $\{x, y, z, w\}$ form a clique but not a triangle clique.

A node k is a *neighbor* of a triangle clique C if k is connected to at least one node of C. It turns out that a neighbor of a triangle clique is connected to all the nodes of that triangle clique.

Lemma 3.4 (Neighbors of a triangle clique). *Consider a triangle clique $C \subseteq V$, and a node $k \in V \setminus C$. Then, either $\{k, i\} \notin E$ for all $i \in C$, or $\{k, i\} \in E$ for all $i \in C$.*

In other words, a neighbor of one node in the triangle clique must be a neighbor of them all, in which case we can call it a *neighbor of the triangle clique.* This lemma allows us to define the concepts *bundle* and *ray*:

Definition 3.5 (Bundle and ray). Consider a triangle clique $C \subseteq V$ and one of its neighbors $k \in V \setminus C$. The set of edges $\{k, i\}$ connecting k with $i \in C$ is called a *bundle* of the triangle clique. Each edge $\{k, i\}$ in a bundle is called a *ray* of the triangle clique.

In Figure 1 the edges (w, x), (w, y), and (w, z) form a bundle of the triangle clique with nodes x, y, and z.

A technical condition to ensure finiteness of the optimal solution. Without loss of generality, we will further assume that no connected component of the graph is a clique — such connected components can be easily detected and handled separately. This ensures that a finite optimal solution exists, as we show in Propositions 3.6 and 3.7.

Proposition 3.6 (Finite feasible region without slacks). *A graph in which no connected component is a clique has a finite feasible region for Problems LP1 and LP2.*

Thus, also the optimal solution is finite. For Problems LP3 and LP4 the following weaker result holds:

Proposition 3.7 (Finite optimal solution with slacks). *A graph in which no connected component is a clique has a finite optimal solution for Problems LP3 and LP4 for sufficiently large C.*

Note that for these problems the feasible region is unbounded.

3.2 Symmetry in the optimal solutions

We now proceed to show that certain symmetries exist in *all* optimal solutions, while for other symmetries we show that there always *exists* an optimal solution that exhibits it.

3.2.1 There always exists an optimal solution that exhibits symmetry. We first state a general result, before stating a more practical corollary. The theorem pertains to automorphisms $\alpha : V \longrightarrow V$ of the graph G, defined as node permutations that leave the edges of the graph unaltered: for α to be a graph automorphism, it must hold that $\{i, j\} \in E$ if and only if $\{\alpha(i), \alpha(j)\} \in E$. Graph automorphisms form a permutation group defined over the nodes of the graph.

Theorem 3.8 (Invariance under graph automorphisms). *For any subgroup $\mathcal{A}$ of the graph automorphism group of G, there exists an optimal solution for the Problems LP1, LP2, LP3 and LP4 that is invariant under all automorphisms $\alpha \in \mathcal{A}$. In other words, there exists an optimal solution $\mathbf{w}$ such that $w_{ij} = w_{\alpha(i)\alpha(j)}$ for each automorphism $\alpha \in \mathcal{A}$.*

Enumerating all automorphisms of a graph is computationally at least as hard as solving the graph-isomorphism problem. The graph-isomorphism problem is known to belong to **NP**, but it is not known whether it belongs to **P**. However, the set of permutations in the following proposition is easy to find.

Proposition 3.9. *The set Π of permutations $\alpha : V \longrightarrow V$ for which $i \in C$ if and only if $\alpha(i) \in C$ for all triangle cliques C in G forms a subgroup of the automorphism group of G.*

Thus the set Π contains permutations of the nodes that map any node in a triangle clique onto another node in the same triangle clique.

We can now state the more practical Corollary of Theorem 3.8:

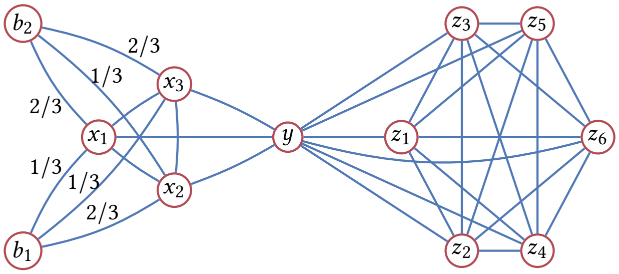


Figure 2: This graph is an example where an optimal solution of Problem LP2 (with $d = 2$) exists that is not constant within a bundle. To see this, note that y is the root of a bundle to both triangle cliques (the one with nodes x_i and the one with nodes z_i). Its rays to both bundles constrain each other in wedge constraints. As the z triangle clique is large, the optimal solution has the largest possible value for edges to those nodes. This is achieved by assigning strengths of 1 to y's rays to z_i, and 0 to y's rays to x_i. Then the triangle edges in the z triangle clique can have strength 3, and the strengths between the x nodes is 2. There are two other bundles to the x triangle clique: from b_1 and b_2. These constrain each other in wedges $(x_i, \{b_1, b_2\})$, such that edges from b_1 and b_2 to the same x_i must sum to 1 at the optimum. Furthermore, triangles $\{b_i, x_j, x_k\}$ impose a constraint on the strength of those edges as: $w_{b_i x_j} + w_{x_i x_k} \leq 2 + d \cdot x_{b_i x_k}$. For $d = 2$ and $w_{x_j x_k} = 2$, this gives: $w_{b_i x_j} \leq 2 \cdot x_{b_i x_k}$. No other constraints apply. Thus, the (unequal) strengths for the edges in the bundles from b_1 and b_2 shown in the figure are feasible. Moreover, this particular optimal solution is a vertex point of the feasible polytope. $1/2$ for each of those edges is also feasible.

Corollary 3.10 (Invariance under permutations within triangle cliques). *Let Π be the set of permutations $\alpha : V \longrightarrow V$ for which $i \in C$ if and only if $\alpha(i) \in C$ for all triangle cliques C. There exists an optimal solution $\mathbf{w}$ for problems LP1, LP2, LP3 and LP4 for which $w_{ij} = w_{\alpha(i)\alpha(j)}$ for each permutation $\alpha \in \Pi$.*

Thus there always exists an optimal solution for which edges in the same triangle clique (i.e., adjacent triangle edges) have equal strength, and for which rays in the same bundle have equal strength.

3.2.2 In each optimum, connected triangle-edges have equal strength. Only some of the symmetries discussed above are present in *all* optimal solutions, as formalized by the following theorem:

Theorem 3.11 (Optimal strengths of adjacent triangle edges are equal). *In any optimal solution of Problems LP1, LP2, LP3 and LP4, the strengths of adjacent triangle edges are equal.*

Note that there do exist graphs for which not all optimal solutions have equal strengths within a bundle. An example is shown in Fig. 2.

3.3 An equivalent formulation for finding symmetric optima of Problem LP2

Solutions that lack the symmetry properties specified in Corollary 3.10 essentially make arbitrary strength assignments. Thus, it makes sense to constrain the search space to just those optimal solutions

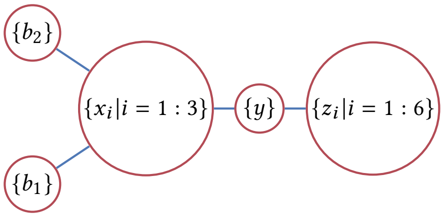


Figure 3: The contracted graph corresponding to the graph shown in Fig. 2.

that exhibit these symmetries.[3] In addition, exploiting symmetry leads to fewer variables, and thus, computational-efficiency gains.

In this section, we will refer to strength assignments that are invariant with respect to permutations within triangle cliques as *symmetric*, for short. The results here apply only to Problem LP2.

The set of free variables consists of one variable per triangle clique, one variable per bundle, and one variable per edge that is neither a triangle edge nor a ray in a bundle. To reformulate Problem LP2 in terms of this reduced set of variables, it is convenient to introduce the *contracted graph*, defined as the graph obtained by edge-contracting all triangle edges in G. More formally:

Definition 3.12 (Contracted graph). Let $\sim$ denote the equivalence relation between nodes defined as $i \sim j$ if and only if i and j are connected by a triangle edge. Then, the contracted graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ with $\widetilde{E} \subseteq \binom{\widetilde{V}}{2}$ is defined as the graph for which $\widetilde{V} = V/\!\sim$ (the quotient set of $\sim$ on V), and for any $A, B \in \widetilde{V}$, it holds that $\{A, B\} \in \widetilde{E}$ if and only if for all $i \in A$ and $j \in B$ it holds that $\{i, j\} \in E$.

Figure 3 illustrates these definitions for the graph from Fig. 2.

We now introduce a vector $\mathbf{w}^t$ indexed by sets $A \subseteq V$, with $|A| \geq 2$, with w^t_A denoting the strength of the edges in the triangle clique $A \subseteq V$. We also introduce a vector $\mathbf{w}^w$ indexed by unordered pairs $\{A, B\} \in \widetilde{E}$, with w^w_{AB} denoting the strength of the wedge edges between nodes in $A \subseteq V$ and $B \subseteq V$. Note that if $|A| \geq 2$ or $|B| \geq 2$, these edges are rays in a bundle.

With this notation, we can state the symmetrized problem as:

$$\max_{\mathbf{w}^t, \mathbf{w}^w} \sum_{A \in \widetilde{V} : |A| \geq 2} \frac{|A|(|A|-1)}{2} w^t_A + \sum_{\{A,B\} \in \widetilde{E}} |A||B| w^w_{AB}, \tag{LP2sym}$$

$$\begin{aligned}
\text{s.t.} \quad & w^w_{AB} + w^w_{AC} \leq 1, && \text{for all } (A, \{B, C\}) \in \widetilde{\mathcal{W}}, && (10)\\
& w^t_A \leq 2 + (d-1) \cdot w^w_{AB}, && \text{for all } \{A, B\} \in \widetilde{E}, |A| \geq 2, && (11)\\
& w^t_A \leq \frac{2}{2-d} \ (\text{if } d < 1), && \text{for all } A \in \widetilde{V}, |A| \geq 3, && (12)\\
& w^t_A \geq 0, && \text{for all } A \in \widetilde{V}, |A| \geq 2, && (13)\\
& w^w_{AB} \geq 0, && \text{for all } \{A, B\} \in \widetilde{E}. && (14)
\end{aligned}$$

[3] It would be desirable to search only for solutions that exhibit all symmetries guaranteed by Theorem 3.8, but given the algorithmic difficulty of enumerating all automorphisms, this is hard to achieve directly. Also, realistic graphs probably contain few automorphisms other than the permutations within triangle cliques. The extended report [1] does however describe an indirect but still polynomial-time approach for finding fully symmetric solutions.

The following theorem shows that there is a one-to-one mapping between optimal solutions to this problem and *symmetric* optimal solutions to Problem LP2, setting $w_{ij} = w^t_A$ if and only if $i, j \in A$, and $w_{ij} = w^w_{AB}$ if and only if $i \in A, j \in B$.

Theorem 3.13 (Problem LP2sym finds symmetric solutions of Problem LP2). *The set of optimal symmetric optimal solutions of Problem LP2 is equivalent to the set of all optimal solutions of Problem LP2sym.*

3.4 The vertex points of the feasible polytope of Problem LP2

The following theorem generalizes the well-known half-integrality result for Problem LP1 [10] to Problem LP2sym.

Theorem 3.14 (Vertices of the optimal face of the feasible polytope). *On the vertices of the optimal face of the feasible polytope of Problem LP2sym, the strengths of the wedge edges take values* $w^w_{AB} \in \left\{0, \frac{1}{2}, 1\right\}$*, and the strengths of the triangle edge take values* $w^t_A \in \left\{2, \frac{d+3}{2}, d+1\right\}$ *if* $d \geq 1$*, or* $w^t_A \in \left\{\frac{2}{2-d}, d+1, \frac{d+3}{2}, 2\right\}$ *if* $d < 1$*. Moreover, for* $d < 1$*, triangle edge strengths for* $|A| \geq 3$ *are all equal to* $w^t_A = \frac{2}{2-d}$ *throughout the optimal face of the feasible polytope.*

This means that there always exists an optimal solution to Problem LP2sym where the edge strengths belong to these small sets of possible values. Note that the symmetric optima of Problem LP2 coincide with those of Problem LP2sym, such that this result obviously also applies to the symmetric optima of LP2.

4 ALGORITHMS

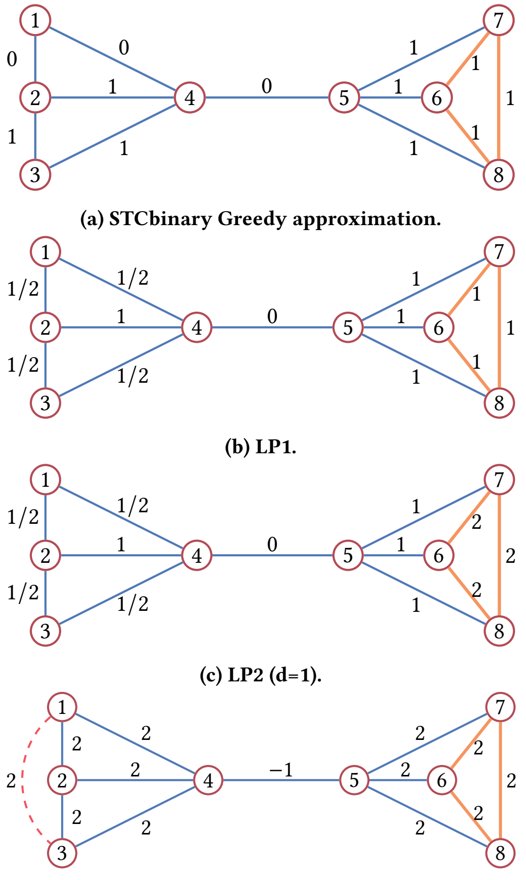
In this section we discuss algorithms for solving the edge-strength inference problems LP1, LP2, LP3, and LP4.

First, all proposed formulations are linear programs (LP), and thus, standard LP solvers can be used. In our experimental evaluation we used CVX [5] from within Matlab, and MOSEK [2] as the solver that implements an interior-point method.

Interior-point algorithms for LP run in polynomial time, namely in $O(n^3L)$ operations, where n is the number of variables, and L is the number of digits in the problem specification [9]. For our problem formulations, L is proportional to the number of constraints. In particular, problem LP1 has $|E|$ variables and $|\mathcal{W}|$ constraints, problem LP2 has $|E|$ variables and $|\mathcal{W}| + |\mathcal{T}|$ constraints, and problems LP3 and LP4 have $|E| + |\bar{E}|$ variables and $|\mathcal{W}| + |\mathcal{T}|$ constraints.

Today, the development of primal-dual methods and practical improvements ensure convergence that is often much faster than this worst-case complexity. Alternatively, one can use the Simplex algorithm, which has worst-case exponential running time, but is known to yield excellent performance in practice [15].

For rational d, we can exploit the special structure of Problems LP1 and LP2 and solve them using more efficient combinatorial algorithms. In particular, the algorithm of Hochbaum and Naor [7] is designed for a family of integer problems named 2VAR problems. It turns out that Problem LP2sym can be formulated as a 2VAR problem for rational d, such that this algorithm is applicable. More details are provided in an extended report [1].



(a) STCbinary Greedy approximation.

(b) LP1.

(c) LP2 (d=1).

(d) LP4 (d=1, C=1).

Figure 4: Toy example with 8 nodes to show the different outcomes of the proposed algorithms. The triangle edges are shown in orange.

5 EMPIRICAL RESULTS

This section contains the main empirical findings. Further details are available in the extended report [1]. The code used in the experiments is available at https://bitbucket.org/ghentdatascience/stc-code-public.

5.1 Qualitative analysis

To gain some insight in our methods, we start by discussing a simple toy example. Figure 4 shows a network of 8 nodes, modelling a scenario of 2 communities being connected by a bridge, i.e., the edge $\{4, 5\}$. The nodes $\{1, 2, 3, 4\}$ form a near-clique—the edge $\{1, 3\}$ is missing—while the nodes $\{5, 6, 7, 8\}$ form a 4-clique. This 4-clique contains a triangle clique: the subgraph induced by the nodes $\{6, 7, 8\}$. Triangle edges are colored orange in the figure.

Fig. 4a contains a solution to STCbinary. Fig. 4b shows a half-integral optimal solution to Problem LP1. We observe that for STCbinary we could swap nodes 1 and 3 and obtain a different yet equally good solution, hence the strength assignment is arbitrary with respect to several edges, while for LP1 the is not the case. Indeed,

Table 1: Network statistics.

Network	Vertices	Edges	Edge weight meaning
Les Mis.	77	254	co-appearence of characters in same chapter
KDD	2 738	11 073	co-authorship between 2 authors
Facebook [16]	3 228	4 585	number of posts on each other's wall
Twitter	4 185	5 680	mentions of each other
Authors	9 150	34 614	unknown
BitCoin OTC	5 875	21 489	Who-trust-whom score in Bitcoin OTC
BitCoin Alpha	3 775	14 120	Who-trust-whom score in Bitcoin Alpha

there is no evidence to prefer a strong label for edges $\{2, 3\}$ and $\{3, 4\}$ over the edges $\{1, 2\}$ and $\{1, 4\}$.

Figure 4c shows a symmetric optimal solution to Problem LP2, allowing for multi-level edge strengths. It labels the triangle edges as stronger than all other (wedge) edges, in accordance with Theorem 3.11 and Theorem 3.14.

Finally, Figure 4d shows the outcome of LP4 for $d = 1$ and $C = 1$, allowing for edge additions and deletions. For $C = 0$, the problem becomes unbounded: the edge $\{4, 5\}$ is only part of wedges, and since wedge violations are unpenalized, $w_{45} = +\infty$ is the best solution (see Section 2.2.2). Since this edge is part of 6 wedges, the problem becomes bounded for $C > 1/6$. For $C = 1$, the algorithm produces a value of 2 for the absent edge $\{1, 3\}$. This suggests the addition of an edge $\{1, 3\}$ with strength 2 to the network, in order to increase the objective function. Edge $\{4, 5\}$, on the other hand, is given a value of -1. As discussed in Section 2.2.2, this corresponds to the strength of an absent edge (when $d = 1$), suggesting the removal of the bridge in the network in order to increase the objective.

A further illustration on a more realistic network is given in Fig 5, which shows the edge strengths assigned by STCbinary (1st), LP1 (2nd), LP2 with $d = 1$ (3rd), and LP4 with $d = 1$ and $C = 1$ (4th). Also here, we see that STCbinary is forced to make arbitrary choices, while LP1, and LP2 avoids this by making use of an intermediate level. Densely-connected parts of the graph tend to contain edges marked as strong, with an extra level of strength for LP2 assigned to the triangle edges. In comparison with LP2, LP4 suggests to remove a lot of weak edges (weight 0 in LP2) that act as bridges between the communities, in order to allow a stronger labeling in the densely-connected regions. Besides edge removal, it also suggests the addition of edges in a near-cliques to form full cliques.

5.2 Objective performance analysis

We evaluate our approaches in a similar manner as Sintos and Tsaparas [14]. In particular, we investigate whether the optimal strength assignments correlate to externally provided ground truth measures of tie strength, on a number of networks for which such information is available. Table 1 shows a summary of the dataset statistics and edge weight interpretations.

We compare the algorithms STCbinary Greedy (which Sintos and Tsaparas found to perform best), LP1 and LP2. For each dataset, the first row in Table 2 displays the number of edges that are assigned in that category. The second row shows the mean ground truth weight over the labeling assigned by the respective algorithm.

Les Miserables is a network where STCbinary Greedy is known to perform well [14]. For this dataset, we can clearly see that our methods provide a correct multi-level strength labeling, enabling more refined notions of tie strength.

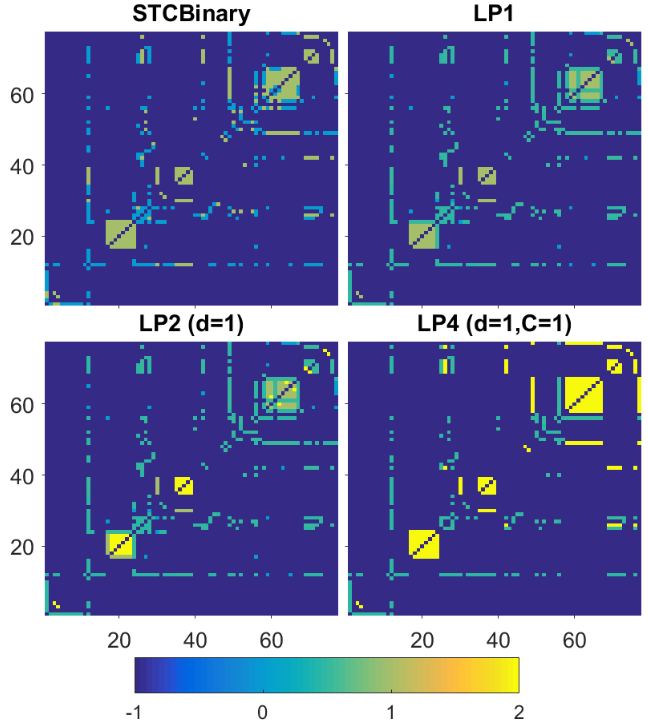


Figure 5: Heatmap of the edge strengths on Les Miserables with methods STCbinary Greedy (1st), LP1 (2nd), LP2 with $d = 1$ (3rd) and LP4 with $d = 1$ and $C = 1$ (4th). A strength of -1 indicates there is no edge.

A second observation is that in for the networks KDD, Facebook, Twitter, and Authors, neither the existing nor the newly-proposed methods perform well. This raises the question of whether the STC assumption is valid in these networks with the provided ground truths.[4] That said, it is reassuring to see that our methods work in a robust and fail-safe way: in such cases, as indicated by the high number of 1/2 strength assignments.

For trust networks in particular, however, it has been described that the STC property is likely to develop due to the transitive property [3]. Indeed, if a user A trusts user B and user B trusts user C, then user A has a basis for trusting user C. The two BitCoin networks are examples of such trust networks. Our methods perform well in identifying some clearly strong and some clearly weak edges, although it again takes a cautious approach in assigning an intermediate strength to many edges. Remarkably though, STCbinary Greedy performs poorly on this network, incorrectly labeling many strong edges as weak and vice versa.

[4] For example, in a co-authorship network, junior researchers having published their first paper with several co-authors could well have all their first edges marked as strong, as their co-authors are connected through the same publication. Yet, they have not yet had the time to form strong connections according to the ground truth. Also, although the Facebook and Twitter networks are social networks, and hence have a natural tendency to satisfy the STC property [3], these sampled networks are too sparse to accommodate a meaningful number of strong edges without any STC violations.

Table 2: Mean ground-truth weight analysis comparison of different STC methods. For each dataset, the first row is the number of edges of an assigned label. The 2nd row indicates the mean groundtruth weight over that respective set of edges. The ground-truth strength ranges are indicated by the numbers between brackets.

Network	STCbinary Greedy		LP1			LP2 (d=1)			
	1	0	1	1/2	0	2	1	1/2	0
Les Mis.	131	123	60	180	14	30	30	180	14
[1–31]	3.6	2.8	4.5	2.9	1.5	3.3	5.7	2.9	1.5
KDD	3 085	7 988	545	10 390	138	290	252	10 396	135
[0.04–47.3]	1.14	0.85	0.89	0.93	0.61	0.77	1.03	0.94	0.61
Facebook	1 451	3 134	28	4 547	10	11	17	4 547	10
[1–30]	1.9	1.94	2.29	1.92	1.5	2.46	2.18	1.92	1.5
Twitter	282	5 398	0	5 680	0	0	0	5 680	0
[1–139]	1.29	2	-	1.97	-	-	-	1.97	-
Authors	16 647	17 967	9 599	22 994	2 021	5 590	4 009	22 994	2 021
[1–52]	1.19	1.4	1.1	1.41	1.16	1.09	1.1	1.41	1.16
BitCoin OTC	1 794	19 695	37	21 446	6	26	11	21 446	6
[−10–10]	0.89	0.62	2.37	0.64	-2.33	2.5	2.1	0.64	-2.33
BitCoin Alpha	1 178	12 942	6	14 113	1	4	2	14 113	1
[−10–10]	1.21	1.43	5	1.4	-10	6	3	1.4	-10

Concerning the running times of the CVX/MOSEK and Hochbaum-Naor implementations, we refer to the extended report [1]. It demonstrates the superior performance of the latter. Remarkably, the Hochbaum-Naor algorithm performs very comparably to the greedy approximation algorithm for STCbinary.

6 RELATED WORK

The work by Sintos and Tsaparas [14] is part of a broader line of active recent research aiming to infer the strength of the links in a social network. E.g., Jones et al. [8] uses frequency of online interaction to predict of strength ties with high accuracy. Gilbert et al. [4] characterize social ties based on similarity and interaction information. Similarly, Xiang et al. [17] estimate relationship strength from homophily principle and interaction patterns and extend the approach to heterogeneous types of relationships. Pham et al. [11] incorporate spatio-temporal features of social iterations to increase accuracy of inferred tie strength. Most of these works, however, make use of various meta-data and characteristics of social interactions in the networks. In contrast, like Sintos and Tsaparas' work, our aim is to infer strength of ties solely based of graph structure, and in particular on the STC assumption.

Another recent extension of the work of Sintos and Tsaparas [14] is followed by Rozenshtein et al. [12]. However, their direction is different: they consider binary strong and weak labeling with additional community connectivity constrains and allow STC violations to satisfy those constraints.

7 CONCLUSIONS AND FURTHER WORK

We have proposed a sequence of linear programming relaxations of the STCbinary problem introduced by Sintos and Tsaparas [14]. These formulations have a number of advantages, most notably their computational complexity, the fact that they refrain from making arbitrary strength assignments in the presence of uncertainty, and as a result, enhanced robustness. Extensive theoretical analysis of the second relaxation (LP2) has not only provided insight into the solution and the arbitrariness the solution from STCbinary may exhibit, it also yielded a highly efficient algorithm for finding a symmetric (non-arbitrary) optimal strenght assignment.

The empirical results confirm these findings. At the same time, they raise doubts about the validity of the STC property in real-life networks, with trust networks appearing to be a notable exception.

Our results suggest a number of possible avenues for further research, discussed in detail in [1]. Perhaps the most important question is whether STC property could be modified so as to become more widely applicable across real-life social networks.

Acknowledgements. This work was supported by the ERC under the EU's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 615517, FWO (project no. G091017N, G0F9816N), the EU's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501, three Academy of Finland projects (286211, 313927, and 317085), and the EC H2020 RIA project "SoBigData" (654024).

REFERENCES

[1] Florian Adriaens, Tijl De Bie, Aristides Gionis, Jefrey Lijffijt, and Polina Rozenshtein. 2018. *From acquaintance to best friend forever: robust and fine-grained inference of social-tie strengths.* Technical Report submitted to Arxiv, identifier not yet known.

[2] MOSEK ApS. 2015. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).* http://docs.mosek.com/7.1/toolbox/index.html

[3] David Easley and Jon Kleinberg. 2010. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, New York, NY, USA.

[4] Eric Gilbert and Karrie Karahalios. 2009. Predicting tie strength with social media. In *Proceedings of the SIGCHI conference on human factors in computing systems.* ACM, 211–220.

[5] Michael Grant and Stephen Boyd. 2014. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. http://cvxr.com/cvx. (March 2014).

[6] Johan Håstad. 1999. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica* 182, 1 (1999), 105–142.

[7] Dorit S Hochbaum and Joseph Naor. 1994. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM J. Comput.* 23, 6 (1994), 1179–1192.

[8] Jason J Jones, Jaime E Settle, Robert M Bond, Christopher J Fariss, Cameron Marlow, and James H Fowler. 2013. Inferring tie strength from online directed behavior. *PloS one* 8, 1 (2013), e52168.

[9] Sanjay Mehrotra and Yinyu Ye. 1993. Finding an interior point in the optimal face of linear programs. *Mathematical Programming* 62, 1 (1993), 497–515.

[10] George L Nemhauser and Leslie Earl Trotter. 1975. Vertex packings: structural properties and algorithms. *Mathematical Programming* 8, 1 (1975), 232–248.

[11] Huy Pham, Cyrus Shahabi, and Yan Liu. 2016. Inferring social strength from spatiotemporal data. *ACM Transactions on Database Systems (TODS)* 41, 1 (2016), 7.

[12] Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. 2017. Inferring the Strength of Social Ties: A Community-Driven Approach. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 1017–1025.

[13] Georg Simmel. 1908. *Soziologie Untersuchungen Ãijber die Formen der Vergesellschaftung.*

[14] Stavros Sintos and Panayiotis Tsaparas. 2014. Using strong triadic closure to characterize ties in social networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 1466–1475.

[15] Daniel A Spielman and Shang-Hua Teng. 2004. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)* 51, 3 (2004), 385–463.

[16] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. 2009. On the Evolution of User Interaction in Facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09).*

[17] Rongjing Xiang, Jennifer Neville, and Monica Rogati. 2010. Modeling relationship strength in online social networks. In *Proceedings of the 19th international conference on World wide web.* ACM, 981–990.