Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-020-00673-0

Relaxing the strong triadic closure problem for edge strength inference

Florian Adriaens[1] · Tijl De Bie[1] · Aristides Gionis[2] · Jefrey Lijffijt[1] · Antonis Matakos[2] · Polina Rozenshtein[2]

Received: 29 June 2018 / Accepted: 3 January 2020

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 to obtain. 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). They exploited this property to formulate the inference of the strength of social ties as a NP-hard maximization 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 one to avoid making arbitrary strong/weak strength assignments when the network topology provides inconclusive evidence. Moreover, these relaxations allow us to easily change the objective function to more sensible alternatives, instead of simply maximizing the number of strong edges. An extensive theoretical analysis leads to two efficient algorithmic approaches. Finally, our experimental results elucidate the strengths of the proposed approach, while at the same time questioning the validity of leveraging the STC property for edge strength inference in practice.

Keywords Strong triadic closure · Strength of social ties · Linear programming · Convex relaxations · Half-integrality

Responsible editor: Tina Eliassi-Rad and Johannes Fïrnkranz.

✉ Florian Adriaens
Florian.Adriaens@ugent.be

Extended author information available on the last page of the article

Published online: 17 January 2020

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. Facebook friendships are formed because of reciprocity, human error or peer-pressure, but also simply because of the existence of real-life friendship. Understanding the strength of social ties can be critical. It is used by sociologists trying to understand user-to-user interactions in complex network dynamics (Granovetter 1977). It has applications in viral marketing (De Meo et al. 2014) and friend recommender systems (Lu and Zhou 2010).

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. Edge strength inference has been extensively studied before, and we refer to Sect. 7 for an overview of existing methods and motivation.

However, we emphasize that the aim of this paper is not to improve upon the state-of-the-art methods for edge strength inference. Instead, the main topic of this paper is improving upon earlier work that leverages the *Strong Triadic Closure* (STC) principle for edge strength inference, and investigating the practical usefulness of methods based on this principle.

Before we discuss our specific contributions, however, let us provide some essential background on prior work on this topic.

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 Simmel (1908) and later popularized by Granovetter (1977). 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 (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 and k are the end-points of the wedge, and we will 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 $\mathcal{T}$. Without loss of generality, in this paper we will assume that the graph does not have any isolated edge: indeed, isolated edges can be assigned to a predefined strength level.

Relying on the STC property, Sintos and Tsaparas (2014) 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, but this is not a meaningful assignment. Instead, they postulate that a large number of strong ties is expected to be found[1] in a social network, and hence 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 (2014) 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.,

[1] However, our empirical findings in Sect. 6 indicate that this is *not* true for most available datasets. This is confirmed by earlier work of De Meo et al. (2014) on the Facebook network.

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 2-approximation algorithm can be adopted for STCmin (f.e., by finding a maximal matching in the wedge graph). 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 $\mathcal{O}(n^{1-\varepsilon})$ (Håstad 1999). 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 a **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* w.r.t. the graph's topology, for edges having different assignments in different optimal solutions. [2] Last but not least, the objective of STCbinary is to maximize the total number of strong edges. This is motivated by the assumption that social networks contain a lot of strong relationships. However, our empirical findings indicate that real-life networks often do *not* have a large amount of strong edges. Instead of maximizing the strong edges, there are other meaningful objective functions that perform better in practice, such as maximizing the number of strong triangles.

Contributions In this paper we propose a series of Linear Programming (LP) 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 a range constraint $0 \leq w_{ij} \leq 1$. It can be shown that this relaxed LP is *half-integral*, i.e., there is an optimal solution for which all edge strengths take values $w_{ij} \in \{0, \frac{1}{2}, 1\}$. Thus, not only the problem becomes polynomial, but the formulation also introduces meaningful three-level edge strengths.
- Next we relax the upper-bound constraint, requiring only $w_{ij} \geq 0$, while generalizing the STC property in triangles to deal with higher gradations of edge strengths. Furthermore, we propose additional relaxations that allow for some STC violations and a change of the objective function.
- We analyze these relaxations and derive properties of their optima, highlighting the benefits of these relaxations with respect to STCbinary.

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

- We show that the relaxations can be solved by the efficient combinatorial Hochbaum–Naor algorithm, and propose a number of methods for reducing aribitrariness w.r.t. the graph's topology.
- Finally, we conduct an extensive experimental evaluation, and discuss the usefulness of leveraging the STC property in practical situations.

Outline We start by proposing the successive relaxations in Sects. 2 and 3. In Sect. 4 we analyze these relaxations and the properties of their optima in more detail. The theory developed in Sect. 4 leads to efficient algorithms, discussed in Sect. 5. Empirical performance is evaluated in Sect. 6, and related work is reviewed in Sect. 7, before drawing conclusions in Sect. 8.

2 Main LP relaxations

Here we will derive two increasingly loose relaxations of Problem STCmax.[3] We simply enlarge the feasible set of strengths w_{ij}, for all edges $\{i, j\} \in E$. The detailed analysis of both problems is deferred to Sect. 4.

2.1 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:

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

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

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

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

Equivalently in Problem STCmin, one can relax constraint (4) to $0 \leq v_{ij} \leq 1$. Recall that Problems STCmax and STCmin are equivalent respectively with the Independent Set and Vertex Cover problems on the wedge graph. For those problems, this particular linear relaxation is well-known, and for Vertex Cover it can be used to achieve a 2-approximation (Hochbaum 1982, 1983).

Clearly, this relaxation will lead to solutions that are not necessarily binary. However, as will be explained in Sect. 4, Problem LP1 is *half-integral*, meaning that there always exists an optimal solution with $w_{ij} \in \{0, \frac{1}{2}, 1\}$ for all $\{i, j\} \in E$.

2.2 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 is to allow for higher gradations in the inference of edge strengths.

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

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 enable such a relaxation, 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:

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

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 $c \geq 2$: indeed, then the box constraints $0 \leq w_{ij} \leq 1$ ensure that the triangle constraint is always satisfied. The tightest possible relaxation is thus achieved with $c = 2$, yielding the following relaxation:

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

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

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.

3 Additional LP formulations

Although the STC property is theoretically motivated, real-world social networks are noisy and may contain many exceptions to this rule. In Sects. 3.1 and 3.2 we propose two further relaxations of Problem LP2 that gracefully deal with exceptions of two kinds: wedges where the sum of edge 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 methods thus solve the STC relaxation in polynomial time, while allowing a small number of edges to be added or removed from the network. We note that Sintos and Tsaparas (2014) also suggest a variant of STCbinary that allows the introduction of new edges. However, the resulting problem is again **NP**-hard, and the provided algorithm provides an $\mathcal{O}(\log(|E|))$-approximation, rather than a constant-factor approximation.

Finally, in Sect. 3.3 we propose an alternative objective function to be maximized. Instead of maximizing the number of strong edges, we propose to maximize the number of *strong triangles* in the graph.

3.1 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 $(i, \{j, k\}) \in \mathcal{W}$:

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

The slack variables ϵ_{jk} can be interpreted as quantifying the strength of the (absent) edge between j and k. In more detail, 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 $w_{jk} = \frac{\epsilon_{jk}-1}{d}$ for $\{j, k\} \in \bar{E}$. The relaxed wedge constraints (9) are then formally equal to the triangle STC constraints (8). Indeed, the upper bound in the relaxed wedge constraint (9) is then equal to $1+\epsilon_{jk} = 2+d\cdot w_{jk}$, which is of the same form as the upper bound in the triangle constraint (8). Meanwhile, the lower bound $\epsilon_{jk} \geq 0$ from (9) is equivalent with $w_{jk} \geq -\frac{1}{d}$, 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:

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

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

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.

3.2 Allowing negative edge strengths

A further relaxation is obtained by allowing edges to have negative strength, with lower bound equal to the strength signifying an absent edge:

$$\begin{aligned} &\max_{\mathbf{w}} \sum_{\{i,j\}\in E} w_{ij} - C \sum_{\{j,k\}\in \bar{E}} w_{jk}, && \text{(LP4)} \\ \text{such that } & w_{ij} + w_{ik} \leq 2 + d \cdot w_{jk}, && \text{for all } (i, \{j,k\}) \in \mathcal{W}, \\ & w_{ij} + w_{ik} \leq 2 + d \cdot w_{jk}, && \text{for all } \{i,j,k\} \in \mathcal{T}, \\ & w_{ij} \geq -\frac{1}{d}, && \text{for all } \{i,j\} \in E. \\ & w_{jk} \geq -\frac{1}{d}, && \text{for all } \{j,k\} \in \bar{E}. \end{aligned}$$

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.

3.3 Maximizing the number of strong edges in all triangles

Sintos and Tsaparas (2014) proposed to maximize the number of strong ties in a network, arguing that “people build social networks with the goal to create strong ties with other people”. However, our emperical findings in Sect. 6 show that most real datasets do *not* have a large number of strong edges. On the contrary, most networks have heavily right-skewed distributions of their empirical edge weights: there are numerously many more weak edges than strong edges. Moreover, the STC property does not provide grounding to label edges that are not part of any triangle as strong.

Hence, alternative objective functions could be more appropriate. For example, in their book “Networks, Crowds and Markets” (Chapter 3.3), Easley and Kleinberg (2010) show that empirical tie strength linearly scales with the neighborhood overlap of an edge in a whotalks-to-whom network maintained by a cell-phone provider that covered roughly 20% of the US population.

Instead of maximizing the total edge weights sum, we propose to maximize the *sum of the weights of all triangles*. Let Δ_{ij} denote the number of triangles that an edge $\{i, j\}$ is part of. Since every edge can be part of multiple triangles, maximizing the sum of the weights of all triangles is equivalent to

$$\max_{\mathbf{w}} \sum_{\{i,j\}\in E} w_{ij} \cdot \Delta_{ij} \qquad \text{(LP}\Delta\text{)}$$

This affine objective function favors edges that are part of many triangles to be labeled as strong, and can be applied to any of the previously discussed LP’s 1–4. Edges that are not part of any triangle can simply be put to their weakest assignment possible, since they do not contribute anything to the objective. Section 4 discusses theoretical results for LP’s 1–4. Without much effort, one can show that the theoretical results from

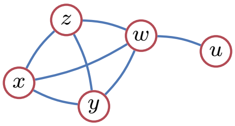


Fig. 1 A toy graph illustrating the different type of edges defined in Sect. 4.1

Sect. 4 still apply to any of the LP△ formulations. More specifically, half-integrality results are still valid.

4 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 $w_{ij} \in \{0, \frac{1}{2}, 1\}$ (Nemhauser and Trotter 1975). In this section we demonstrate how to exploit the symmetries in the optima, and show an 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 even structurally-indistinguishable edges might be assigned different strengths at the optima. Furthermore, in Sect. 5 we will show how the symmetries can be exploited for algorithmic performance gains, as well.

We start by giving some useful definitions and lemmas.

4.1 Auxiliary definitions and results

It is useful to distinguish two types of edges:

Definition 2 (*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 Fig. 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 1 (Subgraph induced by triangle edges) *Each connected component in the edge-induced subgraph, induced by all triangle edges, is a clique.*

Proof See “Appendix”. □

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

Definition 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 Fig. 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 2 (Neighbors of a triangle clique) *Consider a triangle clique $C \subseteq V$, and a node $k \in V \backslash C$. Then, either $\{k, i\} \notin E$ for all $i \in C$, or $\{k, i\} \in E$ for all $i \in C$.*

Proof See "Appendix". □

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*. Lemma 2 allows us to define the concepts *bundle* and *ray*:

Definition 4 (*Bundle and ray*) Consider a triangle clique $C \subseteq V$ and one of its neighbors $k \in V \backslash C$. 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.

Note that rays of a triangle clique are always wedge edges. Indeed, otherwise they would have to be part of the triangle clique.

In Fig. 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 1 and 2. These propositions rest on the following lemma:

Lemma 3 (Each triangle edge is adjacent to a wedge edge) *Each triangle edge in a graph without cliques as connected components is immediately adjacent to a wedge edge.*

Proof See "Appendix". □

Proposition 1 (Finite feasible region in Problems LP1 and LP2) *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.

Proof See "Appendix". □

For Problems LP3 and LP4 the following weaker result holds:

Proposition 2 (Finite optimal solution in Problems LP3 and LP4) *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.

Proof See "Appendix". □

4.2 Symmetry in the optimal solutions

We now proceed to show that certain symmetries exist in *all* optimal solutions (Sect. 4.2.2), while for other symmetries we show that there always *exists* an optimal solution that exhibits it (Sect. 4.2.1).

4.2.1 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 $\alpha : V \rightarrow V$ 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 1 (Invariance under graph automorphisms) *For any subgroup $\mathcal{A}$ of the graph automorphism group of G, there exists an optimal solution for 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 $\alpha \in \mathcal{A}$.*

Proof See "Appendix". □

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 *The set Π of permutations $\alpha : V \rightarrow V$ 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.

Proof See "Appendix". □

We can now state the more practical Corollary of Theorem 1:

Corollary 1 (Invariance under permutations within triangle cliques) *Let Π be the set of permutations $\alpha : V \rightarrow V$ 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.

Such a symmetric optimal solution can be constructed from any other optimal solution, by setting the strength of a triangle edge equal to the average of strengths within the triangle clique it is part of, and setting the strength of each ray equal to the average of the strengths within the bundle it is part of. Indeed, this averaged solution is equal to the average of all permutations of the optimal solution, which, from convexity of the problem, is also feasible and optimal.

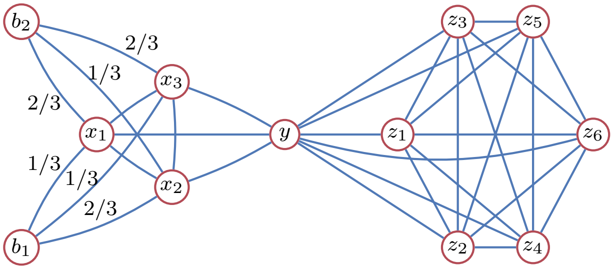


Fig. 2 This graph is an example where an optimal solution of Problem LP2 (with $d = 2$) exists that is not constant within a bundle. To see this, note that y is the root of a bundle to both triangle cliques (the one with nodes x_i and the one with nodes z_i). Its rays to both bundles constrain each other 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 (proof not given). Note that strengths equal to 1/2 for each of those edges is also feasible

4.2.2 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 2 (Strengths of adjacent triangle edges in the optimum) *In any optimal solution of Problems LP1, LP2, LP3 and LP4, the strengths of adjacent triangle edges are equal.*

Proof See "Appendix". □

In other words, all triangle edges within the same triangle clique have the same strength in any optimal solution of these problems. However, there do exist graphs for which not all optimal solutions have equal strengths for the rays within a bundle. An example is shown in Fig. 2.

4.3 An equivalent formulation for finding symmetric optima of problem LP2

Corollary 1 asserts that symmetries in the structure of the graph imply that there are optimal solutions that exhibit symmetries with respect to the strength of the corresponding edges. This is an intuitive property: solutions that lack these symmetry properties essentially make arbitrary strength assignments. Thus, it makes sense to constrain the search space to just those optimal solutions that exhibit these symmetries.[4]

[4] It would be desirable to search only for solutions that exhibit all symmetries guaranteed by Theorem 1, but given the algorithmic difficulty of enumerating all automorphisms, this is hard to achieve directly. Also,

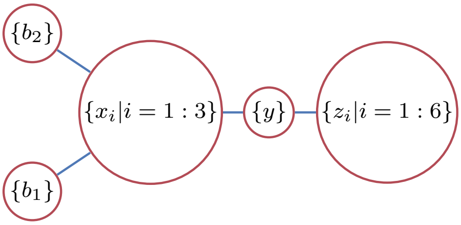


Fig. 3 The contracted graph corresponding to the graph shown in Fig. 2

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 5 (*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.

The contracted graph will of course contain wedges, the set of which will be denoted as $\widetilde{\mathcal{W}}$. 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:

$$\max_{\mathbf{w}^t, \mathbf{w}^w} \sum_{A \in \widetilde{V}: |A| \geq 2} \frac{|A|(|A|-1)}{2} w^t_A + \sum_{\{A,B\} \in \widetilde{E}} |A||B| w^w_{AB}, \tag{LP2SYM}$$

$$\text{such that} \quad w^w_{AB} + w^w_{AC} \leq 1, \qquad \text{for all } (A, \{B, C\}) \in \widetilde{\mathcal{W}}, \tag{10}$$

$$w^t_A \leq 2 + (d-1) \cdot w^w_{AB}, \qquad \text{for all } \{A, B\} \in \widetilde{E}, |A| \geq 2, \tag{11}$$

$$w^t_A \leq \frac{2}{2-d} \ (\text{if } d < 1), \qquad \text{for all } A \in \widetilde{V}, |A| \geq 3, \tag{12}$$

realistic graphs probably contain few automorphisms other than the permutations within triangle cliques. Section 5.3 does however describe an indirect but still polynomial-time approach for finding fully symmetric solutions.

$$w^t_A \geq 0, \qquad \text{for all } A \in \widetilde{V}, |A| \geq 2, \tag{13}$$

$$w^w_{AB} \geq 0, \qquad \text{for all } \{A, B\} \in \widetilde{E}. \tag{14}$$

The following theorem shows that there is a one-to-one mapping between the optimal solutions of Problem LP2SYM and the *symmetric* optimal solutions of Problem LP2. In particular the mapping is given by 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 (Problem LP2SYM finds symmetric solutions of Problem LP2) *The set of symmetric optimal solutions of Problem LP2 is equivalent to the set of all optimal solutions of Problem LP2SYM.*

Proof See “Appendix”. □

4.4 The vertex points of the feasible polytope of problem LP2

The following theorem generalizes the well-known half-integrality result for Problem LP1 (Nemhauser and Trotter 1975) to Problem LP2SYM.

Theorem 4 (Vertices of the feasible polytope) *On the vertex points 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 edges take values $w^t_A \in \left\{0, 2, \frac{d+3}{2}, d+1\right\}$ for $d \geq 1$, or $w^t_A \in \left\{0, \frac{2}{2-d}, d+1, \frac{d+3}{2}, 2\right\}$ for $d < 1$.*

Proof See “Appendix”. □

Corollary 2 *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.*

Proof See “Appendix”. □

Corollary 2 asserts 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.

5 Algorithms

In this section we discuss algorithms for solving the edge-strength inference problems LP1, LP2, LP3, LP4, and LP2SYM. The final Sect. 5.3 also discusses a number of ways to further reduce the arbitrariness of the optimal solutions.

5.1 Using generic LP solvers

First, all proposed formulations are linear programs (LP), and thus, standard LP solvers can be used. In our experimental evaluation we used CVX (Grant and Boyd 2014) from within Matlab, and MOSEK (ApS 2015) as the solver that implements an interior-point method.

Interior-point algorithms for LP run in polynomial time, namely in $\mathcal{O}(n^3L)$ operations, where n is the number of variables, and L is the number of digits in the problem specification (Mehrotra and Ye 1993). 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. Here $|E|$ is the number of edges in the input graph, $|\mathcal{W}|$ the number of wedges, and $|\mathcal{T}|$ the number of triangles.

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 (Spielman and Teng 2004).

5.2 Using the Hochbaum–Naor algorithm

For rational d, we can also exploit the special structure of Problems LP1 and LP2SYM and solve them using more efficient combinatorial algorithms. In particular, the algorithm of Hochbaum and Naor (1994) is designed for a family of integer problems named 2VAR problems. 2VAR problems are integer programs (IP) with 2 variables per constraint of the form $a_k x_{i_k} - b_k x_{j_k} \geq c_k$ with rational a_k, b_k, and c_k, in addition to integer lower and upper bounds on the variables. A 2VAR problem is called *monotone* if the coefficients a_k and b_k have the same sign. Otherwise the IP is called *non-monotone*. The algorithm of Hochbaum and Naor (1994) gives an *optimal integral* solution for monotone IPs and an *optimal half-integral* solution for non-monotone IPs. The running time of the algorithm is pseudopolynomial, that is, polynomial in the range (difference between lower bound and upper bound) of the variables. More formally, the running time is $\mathcal{O}(n\Delta^2(n + r))$, where n is the number of variables, r is the number of constraints, and Δ is maximum range size. For completeness, we briefly discuss the problem and algorithm for solving it below.

The monotone case We first consider an IP with monotone inequalities:

$$\max \sum_{i=1}^{n} d_i x_i, \qquad \text{(monotone IP)}$$

$$\text{such that } a_k x_{i_k} - b_k x_{j_k} \geq c_k \qquad \text{for } k = 1, \ldots, r, \qquad (15)$$

$$\ell_i \leq x_i \leq u_i, \quad x_i \in \mathbb{Z}, \qquad \text{for } i = 1, \ldots, n, \qquad (16)$$

where a_k, b_k, c_k, and d_i are rational, while ℓ_i and u_i are integral. The coefficients a_k and b_k have the same sign, and d_i can be negative.

The algorithm is based on constructing a weighted directed graph $G' = (V', E')$ and finding a minimum $s - t$ cut on G'.

For the construction of the graph G', for each variable x_i in the IP we create a set of $(u_i - \ell_i + 1)$ nodes $\{v_{ip}\}$, one for each integer p in the range $[\ell_i, u_i]$. An auxiliary source node s and a sink node t are added. All nodes that correspond to positive integers are denoted by V^+, and all nodes that correspond to non-positive integers are denoted by V^-.

The edges of G' are created as follows: First, we connect the source s to all nodes $v_{ip} \in V^+$, with $\ell_i + 1 \leq p \leq u_i$. We also connect all nodes $v_{ip} \in V^-$, with $\ell_i + 1 \leq p \leq u_i$, to the sink node t. All these edges have weight $|d_i|$. The rest of the edges described below have infinite weight.

For the rest of the graph, we add edges from s to all nodes v_{ip} with $p = \ell_i$—both in V^+ and V^-. For all $\ell_i + 1 \leq p \leq u_i$, the node v_{ip} is connected to $v_{i(p-1)}$ by a directed edge. Let $q_k(p) = \lceil \frac{c_k + b_k p}{a_k} \rceil$. For each inequality k we connect node $v_{j_k p}$, corresponding to x_{j_k} with $\ell_{j_k} \leq p \leq u_{j_k}$, to the node $v_{i_k q}$, corresponding to x_{i_k} where $q = q_k(p)$. If $q_k(p)$ is below the feasible range $[\ell_{i_k}, u_{i_k}]$, then the edge is not needed. If $q_k(p)$ is above this range, then node $v_{j_k p}$ must be connected to the sink t.

Hochbaum and Naor (1994) show that the optimal solution of (monotone IP) can be derived from the source set S of minimum s-t cuts on the graph G' by setting $x_i = \max\{p \mid v_{ip} \in S\}$. The complexity of this algorithm is dominated by solving the minimum s-t cut problem, which is $\mathcal{O}(|V'||E'|) = \mathcal{O}(n\Delta^2(n + r))$, where n is the number of variables in the (monotone IP) problem, r is the number of constraints, and Δ is maximum range size $\Delta = max_{i=[1,n]}(u_i - \ell_i + 1)$. Note also that in practice the graph is sparse and finding the cut is faster than this theoretical complexity analysis might suggest.

Monotonization and half-integrality A non-monotone IP with two variables per constraint is **NP**-hard. Edelsbrunner et al. (1989) showed that a non-monotone IP with two variables per constraint has half-integral solutions, which can be obtained by the following monotonization procedure. Consider a non-monotone IP:

$$\max \sum_{i=1}^{n} d_i x_i, \qquad \text{(non-monotone IP)}$$

$$\text{such that } a_k x_{i_k} + b_k x_{j_k} \geq c_k \qquad \text{for all } k = 1, \dots, m, \tag{17}$$

$$\ell_i \leq x_i \leq u_i, \qquad x_i \in \mathbb{Z}, \qquad \text{for all } i = 1, \dots, n, \tag{18}$$

with no constraints on the signs of a_k and b_k.

For monotonization we replace each variable x_i by $x_i = \frac{x_i^+ - x_i^-}{2}$, where $\ell_i \leq x_i^+ \leq u_i$ and $-u_i \leq x_i^- \leq -\ell_i$. Each non-monotone inequality (a_k and b_k having the same sign) $a_k x_{i_k} + b_k x_{j_k} \geq c_k$ is replaced by a pair:

$$a_k x_{i_k}^+ - b_k x_{i_k}^- \geq c_k \tag{19}$$

$$-a_k x_{i_k}^- + b_k x_{i_k}^+ \geq c_k \tag{20}$$

Each monotone inequality $\bar{a}_k x_{i_k} - \bar{b}_k x_{j_k} \geq c_k$ is replaced by:

$$\bar{a}_k x^+_{i_k} - \bar{b}_k x^+_{j_k} \geq c_k \tag{21}$$

$$-\bar{a}_k x^-_{i_k} + \bar{b}_k x^-_{j_k} \geq c_k \tag{22}$$

The objective function is replaced by $\sum_{i=1}^{n} \frac{1}{2} d_i x_i^+ - \frac{1}{2} d_i x_i^-$.

By construction, the resulting monotone IP is a half-integral relaxation of the Problem (non-monotone IP).

LP1 is a (non-monotone) 2VAR system, so that it can directly be solved by the algorithm of Hochbaum and Naor. Problem LP2, however, is not a 2VAR problem, such that the Hochbaum and Naor algorithm is not directly applicable. Yet for integer $d \geq 1$, Problem LP2SYM *is* a 2VAR problem. The lower bound on each of the variables is 0, and the upper bound is equal to 1 for the wedge edges and $d + 1$ for the triangle edges—i.e., both lower and upper bound are integers. For rational d, the upper bound is $\max\{2, d + 1\}$, which may be rational, but reformulating the problem in terms of $a \cdot w_{ij}$ for a the smallest integer for which $a \cdot d$ is integer turns it into a 2VAR problem again. Thus, for d rational, finding one of the *symmetric* solutions of Problem LP2 *can* be done using Hochbaum and Naor's algorithm. Moreover, this symmetric solution will immediately be one of the half-integral solutions we know exist from Corollary 2. Note that the monotonization and using the Hochbaum–Naor algorithm still works when changing the objective to LPΔ.

5.3 Approaches for further reducing arbitrariness

As pointed out in Sect. 4.3, Problem LP2SYM does not impose symmetry with respect to *all* graph automorphisms, as it would be impractical to enumerate them. However, in Sect. 5.3.1 below we discuss an efficient (polynomial-time) algorithm that is able to find a solution that satisfies all such symmetries, without the need to explicitly enumerate all graph automorphisms.

Furthermore, in Sects. 5.3.2 and 5.3.2, we discuss strategies for reducing arbitrariness that is not based on finding a fully symmetric solution. These algorithms attempt to meaningfully, and in polynomial time, describe the entire optimal face of the feasible polytope, rather than selecting a single optimal (symmetric) solution from it.

Several algorithms discussed below exploit the following characterization of the optimal face. As an example, and with o^* the value of the objective at the optimum, for Problem LP2SYM this characterization is:

$$\begin{aligned}
\mathcal{P}^* = \Big\{\mathbf{w} \mid \sum_{\{i,j\}\in E} w_{ij} = o^*, & \\
w_{ij} + w_{ik} \leq 1, & \qquad \text{for all } (i, \{j,k\}) \in \mathcal{W}, \\
w_{ij} + w_{ik} \leq 2 + d \cdot w_{jk}, & \qquad \text{for all } \{i,j,k\} \in \mathcal{T}, \\
w_{ij} \geq 0, & \qquad \text{for all } \{i,j\} \in E. \Big\}
\end{aligned}$$

It is trivial to extend this to the optimal faces of the other problems.

5.3.1 Invariance with respect to all graph automorphisms

Here we discuss an efficient algorithm to find a fully symmetric solution, without explicitly having to enumerate all graph automorphisms.

Given the optimal value of the objective function of (for example) Problem LP2, consider the following problem which finds a point in the optimal face of the feasible polytope that minimizes the sum of squares of all edge strengths:

$$\begin{aligned} \min_{\mathbf{w}} \sum_{\{i,j\}\in E} (w_{ij})^2, \\ \text{such that } \mathbf{w} \in \mathcal{P}^*. \end{aligned} \tag{LP2FULLSYM}$$

As $\mathcal{P}^*$ is a polytope, this is a Linearly Constrained Quadratic Program (LCQP), which again can be solved efficiently using interior point methods.

Theorem 5 (Problem LP2FULLSYM finds a solution symmetric with respect to all graph automorphisms) *The edge strength assignments that minimize Problem LP2FULLSYM are an optimal solution to Problem LP2 that is symmetric with respect to all graph automorphisms.*

Proof See "Appendix". □

For simplicity of notation, we explained this strategy for Problem LP2, but of course it is computationally more attractive to seek a solution within the optimal face of the feasible polytope for Problem LP2SYM. Note that Theorem 5 still holds for LP1, LP3 and LP4, and also when changing the objective to LPΔ.

5.3.2 Characterizing the entire optimal face of the feasible polytope

Here, we discuss an alternative strategy for reducing arbitrariness, which is to characterize the entire optimal face of the feasible polytope of the proposed problem formulations, rather than to select a single (possibly arbitrary) optimal solution from it. Specifically, we propose three algorithmic implementations of this strategy.

The first algorithmic implementation of this strategy exactly characterizes the range of the strength of each edge amongst the optimal solutions. This range can be found by solving, for edge strength w_{ij} for each $\{i, j\} \in E$, two optimization problems:

$$\begin{aligned} \max_{\mathbf{w}} w_{ij} \quad \text{and} \quad \min_{\mathbf{w}} w_{ij}, \\ \text{such that } \mathbf{w} \in \mathcal{P}^*. \end{aligned}$$

These are again LP's, and thus require polynomial time. Yet, it is clear that this approach is impractical, as the number of such optimization problems to be solved is twice the number of variables in the original problem.

The second algorithmic implementation of this strategy is computationally much more attractive, but quantifies the range of each edge strength only partially. It exploits the fact that the strengths at the vertex points of the optimal face belong to a finite set of values. Thus, given any optimal solution, we can be sure that for each edge, there exists an optimal solution for which any given edge's strength is equal to the smallest value within that set equal to or exceeding the value in that optimal solution, Moreover, it is equal to the largest value within that set equal to or smaller than the value in that optimal solution. To ensure this range is as large as possible, it is beneficial to avoid finding vertex points of the feasible polytope, and more generally points that do not lie within the *relative interior* of the optimal face. This can be done in the same polynomial time complexity as solving the LP itself, namely $\mathcal{O}(n^3 L)$ where L is the input length of the LP (Mehrotra and Ye 1993). This could be repeated several times with different random restarts to yield wider intervals for each edge strength.

The third implementation is to uniformly sample points (i.e., optimal solutions) from the optimal face $\mathcal{P}^*$. A recent paper (Chen et al. 2017) details an MCMC algorithm with polynomial mixing time for achieving this.

5.3.3 Approximating the centroid of the optimal face by using the Chebychev center

The centroid of the optimal face is the least arbitrary solution out of all optimal solutions. Like LP2FULLSYM, it is easy to see that the centroid is also symmetric with respect to all graph automorphisms. One problem in practice with LP2FULLSYM, is that it will often assign a large number of 1/2-weighted edges (because of the quadratic minimization), whereas the centroid will not use more 1/2-weighted edges than needed. However, finding the centroid of a polytope is **#P**-hard, even when the polytope is given as an intersection of halfspaces (Rademacher 2007). We will use the Chebychev center for a good approximation of the centroid of the optimal face.

The Chebychev center y_c of a polytope $\mathcal{P} = \{y \in \mathbb{R}^n | a_i^T y \leq b_i\}$ is defined as the center of the largest inscribed ball, and can be determined by solving the following LP:

$$\max r \geq 0, \qquad \text{(LPCHEBY)}$$
$$\text{such that } \forall i : a_i^T y_c + r\|a_i\|_2 \leq b_i.$$

In our case, we want the Chebychev center of the *relative interior* of the optimal face. This can be easily achieved as follows: After finding an initial solution in the relative interior of the optimal face, solve (LPCHEBY) with ball constraints only applied to the *inactive* constraints $a_i^T y < b_i$, while keeping the *active* constraints $a_i^T y = b_i$ in their original form.

Let us look at the following example to see the difference between LP2FULLSYM and LPCHEBY. Figure 4 shows 3 optimal solutions to Problem LP2 on a small graph. Since all edges are part of atleast one wedge, the triangle constraints are redundant and not active throughout the feasible polytope. Observe that the wedge $(x_4, \{x_3, x_5\})$ has an inactive constraint at the optimal face, because there exists an optimal solution where $w_{x_3x_4} + w_{x_4x_5} = 0$ (see Fig. 4a). Similarly for the wedge $(x_4, \{x_1, x_5\})$. The wedges

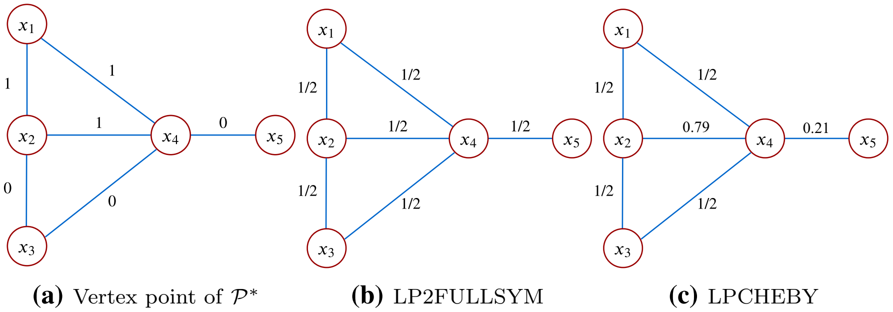


Fig. 4 Toy example to show the difference between LP2FULLSYM and LPCHEBY ($d > 0$)

$(x_4, \{x_2, x_5\})$, $(x_4, \{x_1, x_3\})$ and $(x_2, \{x_1, x_3\})$ always have active constraints. Hence, the LPCHEBY solution is formed by giving an equal amount of slack to the two inactive wedge constraints, see Fig. 4c. After simplyfing and ignoring redundant equations, we find that LPCHEBY reduces to

$$\begin{aligned} &\max r \geq 0, \\ \text{such that } & w_{x_4 x_5} + r\sqrt{2} \leq 1/2, \\ & w_{x_2 x_4} + w_{x_4 x_5} = 1, \\ & w_{x_2 x_4} \geq r \quad \text{and} \quad w_{x_4 x_5} \geq r. \end{aligned}$$

The radius r is maximized by setting $w_{x_4 x_5} = r$, leading to a unique solution where $r = \frac{1}{2(1+\sqrt{2})} \approx 0.21$, see Fig. 4c. Conversely, LP2FULLSYM simply assigns all edges to a $1/2$, minimizing the quadratic form discussed in Sect. 5.3.1. Observe that Fig. 4a is also a solution to STCbinary, showing the arbitrariness (asymmetry) in the proposed problem setting.

On this example, LPCHEBY has better explainability than LP2FULLSYM: the edge (x_2, x_4) is part of more triangles and less wedges than the edge (x_4, x_5), so intuitively it should get a stronger assignment. Note that LPCHEBY is not compatible with the efficient Hochbaum–Naor algorithm, discussed in Sect. 5.2, since LPCHEBY needs a point in the relative interior of the optimal face as an input. Hence in general, this approach is not very scalable.

Remark 2 For general polytopes, the Chebychev center is not unique and is not necessarily a good approximation of the centroid. This occurs in cases where the polytope is "long and thin" (Boyd and Vandenberghe 2004). However, the vertex points of the feasible polytopes of the LP's discussed in this paper are all half-integral, and thus avoiding these cases, motivating our use for the Chebychev center. Alternatively, the centroid can be approximated by uniformly sampling the optimal face $\mathcal{P}^*$ (Chen et al. 2017), but this is not tested in this paper.

6 Empirical results

This section contains the main empirical findings. The code used in the experiments is publicly available[5]. All experiments were run on a large server with 48 Intel (R) Xeon Gold 6136 CPU cores @ 3.00GHz and a total of 256 GB RAM. We will denote STCbinary [GREEDY] as the greedy algorithm, and STCbinary [MM] as the maximal matching algorithm proposed by Sintos and Tsaparas (2014). STCbinary [GREEDY] assigns the edge that is part of most wedges as weak, and then ignores those wedges in future countings. STCbinary [MM] finds a maximal matching in the wedge graph G_E, and then assigns those nodes in the maximal matching (corresponding to edges in the original graph) as weak. Both algorithms have an important tiebreaker: they break ties by assigning the edge with the *least* amount of common neighbors as weak.

We start by thoroughly discussing a toy example (Sect. 6.1). After that, we evaluate performance on 20 real datasets (Sects. 6.2, 6.3). In Sect. 6.4 we visualize two small networks and discuss the outcomes of the methods in more detail.

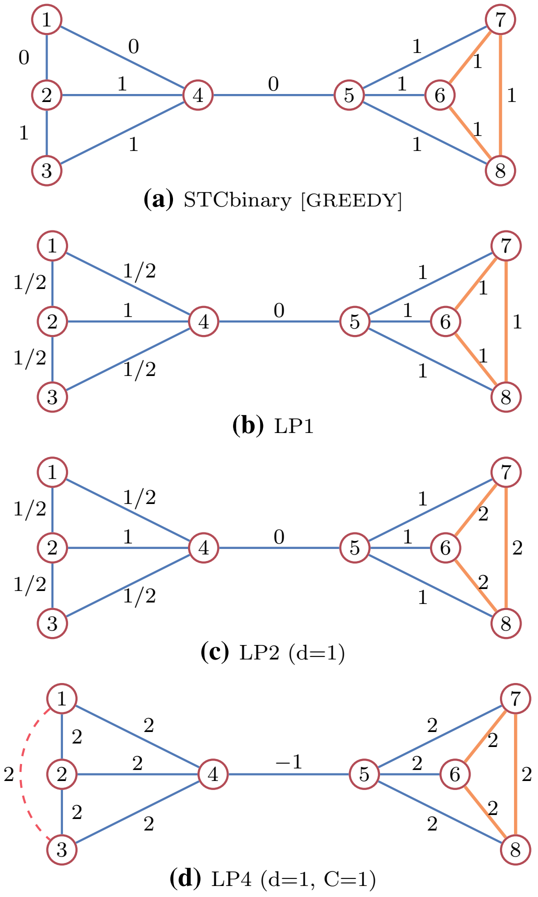
6.1 Discussion of a toy example in detail

To gain some insight in our methods, we start by discussing a simple toy example. Figure 5 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 $\{6, 7, 8\}$.

Figure 5a shows the solution by STCbinary [GREEDY], which is optimal for the STCbinary problem. Figure 5b shows a half-integral optimal solution to Problem LP1. We observe that for STCbinary [GREEDY] 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, 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 5c 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 2 and Corollary 2. Figure 5d 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 Sect. 3.2). 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. The addition of this edge decreases the objective function by 2, but enables the sum of the edges $\{1, 2\}$ and $\{2, 3\}$ to increase from 1 to 4, leading to a net gain of $3 - 2 = 1$ in the objective function. This is the only edge being suggested for addition by the algorithm. Edge $\{4, 5\}$, on the other hand, is given a value of -1. As discussed in Sect. 3.2, 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

[5] https://bitbucket.org/ghentdatascience/stc-code-public.

Fig. 5 Toy example with 8 nodes to show the different outcomes of the proposed algorithms. The triangle clique is shown in orange (Color figure online)



(a) STCbinary [GREEDY]

(b) LP1

(c) LP2 (d=1)

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

objective. For large C there will be no more edge additions being suggested, as can be seen by setting $C = \infty$ in LP4 (reducing it to LP3). The cost of a violation of a wedge constraint will always be higher than the possible benefits. However, regardless of the value of C, the edge $\{4, 5\}$ is always being suggested for edge deletion.

For this example, there is no difference in the outcomes of the LPFULLSYM and LPCHEBY algorithms, nor does changing the objective to LPΔ have an effect.

6.2 Performance analysis on real datasets

We evaluate our approach on the 20 datasets shown in Table 1. Most of these datasets have been previously used in edge strength prediction problems (Gupte and Eliassi-Rad 2011; Rozenshtein et al. 2017; Sintos and Tsaparas 2014). For all these datasets, empirical tie strength is given and their interpretation is shown in the last column. Table 1 shows basic statistics of the datasets. The clustering coefficient is shown in the 4th column. To measure if STC in present in a dataset, we look if “strong” connected triples are closed more often than “normal” connected triples. In order to do so, we define strong by looking at percentiles. The 5th column shows the clustering coefficient, when computed on the induced subgraph defined by keeping only the 10% strongest edges.[6] In all datasets except Actors, strong connected triples are more closed than normal connected triples, providing evidence that STC is atleast partially present in these datasets.

To measure the correlation of a method's proposed edge strength and the empirical edge strength, we use the Kendall τ_b rank correlation coefficient (Kendall and Gibbons 1990). We compare our methods with both the STCbinary [GREEDY] and STCbinary [MM] algorithms, as well as some simple but powerful baselines: Common Neighbors (number of triangles an edge is part of) and Preferential Attachment (the product of the degrees of the two nodes of an edge). We test both the main relaxations LP1 and LP2 ($d = 1$), as well as the LPΔ formulations (Sect. 3.3). Note that we will use LP2SYM to solve LP2, but will just denote it as LP2 for brevity. To reduce arbitrariness w.r.t. the graph structure, we compute the Chebychev center (Sect. 5.3.3) as the final assignment. The performance of LP4 was similar to method LP2, and is not reported here due to the overhead of tuning the parameter C.

Table 2 shows the result on the 20 different datasets. Our methods were the best performing on only 5 out of 20 datasets: Southern Woman, Freeman 2, Les Miserables, Twitter (tied with Common Neighbors) and Actors. The first 3 of them are all small datasets with a relatively high clustering coefficient. Moreover, the empirical edge strength distribution for these networks were not that heavily right-skewed as the other small datasets (like Freeman 1, Terrorists, Beach, Kangaroo and Students), indicating that the objective of maximizing strong edges (or triangles) makes more sense here. We refer to Sects. 6.4 and 8 for a more thorough discussion.

Note the increase in performance for both LP1 and LP2 when changing the objective to LPΔ. Also, it is remarkable that the main relaxations LP1 and LP2 do not outperform the methods for STCbinary (Even though they are better, by definition of the relaxations, at maximizing the objective function). Although not tested over all datasets, this could be an effect of the tiebreaker of both STCbinary [GREEDY] and STCbinary [MM]: ties are breaked by selecting the edge with the *least* amount of common neighbors as the next weak edge. Averaged over all datasets, Common Neighbors was the best performing, a powerful baseline that is known to work well for link prediction.

Another observation is that for the very sparse networks with low clustering coefficient (Students, Twitter), the LP methods work in a more robust way than the STCbinary algorithms. Indeed, for the Students dataset, STCbinary [GREEDY] assigned 12% of the edges as strong. Most of these assignments were false positives. Instead, LP1 assigned 99% of the edges a 1/2 score (intermediate), while only 0.5% edges got a score of 0.21 (weak), and the remaining 0.5% edges got a score of 0.79 (strong). Interestingly, the values of 0.21 and 0.79 coincide with the assignments of the Chebychev center on the toy graph in Fig. 4. When the objective is changed to LPΔ, the labelling changed quite drastically: 66,5% of the edges were now assigned a 0 (weak), 32,7% of the edges got a 1/2 (intermediate), and 0.8% of the edges were labeled a 1 (strong). This clearly shows that the LPΔ formulation takes the absence of triangles in this network

[6] There are other meaningful ways of defining strong edges. Moreover, another perecentile than 10% could be chosen. However, the conclusion that STC is present in the datasets remains the same.

Table 1 Network statistics

Network	Vertices	Edges	C.C.	C.C. (10%)	Edge weight interpretation
Kangaroo[a]	17	91	0.84	1	Interactions between kangaroos
Southern Women[b]	18	139	0.93	1	Co-attendance to social events
Terrorists[a]	64	243	0.56	0.92	Relationships between terrorists
Beach[a]	43	336	0.56	0.95	Interactions between windsurfers
Freeman 1[b]	32	266	0.67	0.94	Messages between researchers
Freeman 2[b]	34	415	0.82	0.92	Relationship strength between researchers
Cross Parker Consulting[b]	46	550	0.72	0.98	Advice requests between employees
Les Miserables[a]	77	254	0.49	0.81	Co-appearence of characters
Students[d]	865	2 254	0.03	0.08	Messages between students
Facebook-like Social[b]	1893	13,835	0.06	0.1	Private messages between students
KDD[e]	2738	11,073	0.16	0.33	Co-authorship between 2 authors
ICDM[e]	2795	10,280	0.17	0.37	Co-authorship between 2 authors
Facebook[c]	26,561	47,400	0.06	0.14	Number of posts on each other's wall
Twitter[d]	40,993	60,992	0	0.02	Mentions of each other

Table 1 continued

Network	Vertices	Edges	C.C.	C.C. (10%)	Edge weight interpretation
Authors[d]	9150	34,614	0.34	0.36	Unknown
BitCoin Alpha[f]	3775	14,120	0.08	0.17	Who-trust-whom score in trading platform
BitCoin OTC[f]	5875	21,489	0.06	0.14	Who-trust-whom score in trading platform
Actors[d]	1986	103,121	0.17	0.17	Unknown
Facebook-like Forum[b]	897	71,380	0.51	0.7	Common messages in a forum topic
Newman's collab. network[b]	13,861	44,619	0.35	0.63	Co-authored papers

The 5th column denotes the clustering coefficient, computed on the induced subgraph only containing the 10% strongest edges. Note the increase when compared to the global clustering coefficient (4th column), providing light evidence that STC is present in these datasets

[a] http://konect.uni-koblenz.de/networks/

[b] https://toreopsahl.com/datasets/

[c] Viswanath et al. (2009)

[d] https://bitbucket.org/ghentdatascience/stc-code-public

[e] http://dblp.uni-trier.de

[f] Leskovec and Krevl (2014)

Table 2 Kendall τ_b rank correlation coefficients between the predicted and empirical edge strengths on the 20 datasets listed in Table 1

Network	STCb [MM]	STCb [GREEDY]	LP1	LP1 (Δ)	LP2	LP2 (Δ)	Common Neighb.	Pref. Attach.
Kangaroo	0.26	− 0.08	− 0.06	0.17	0	0.21	0.48	**0.5**
Southern Women	0.36	0.39	0.39	0.39	0.45	**0.45**	0.4	0.35
Terrorists	0.25	0.21	0.02	0.09	0	0.06	**0.32**	0.26
Beach	0.27	0.31	0.07	0.07	0.07	0.07	**0.38**	0.24
Freeman 1	0.23	0.21	0.17	0.1	0.17	0.10	0.27	**0.31**
Freeman 2	0.12	0.11	0.07	0.22	0.07	**0.22**	0.15	0.14
Cross Parker Consulting	0.16	**0.21**	− 0.03	0.19	− 0.03	0.19	0.11	− 0.05
Les Miserables	0.36	0.34	0.34	**0.46**	0.32	0.43	0.36	0.20
Students	− 0.02	− 0.06	0.02	0.15	0.02	0.14	0.15	**0.16**
Facebook-like Social	− 0.04	− 0.05	0	0.15	0	0.15	**0.16**	0.16
KDD	**0.08**	0.07	0.02	0.01	0.02	0.01	0.01	− 0.05
ICDM	0.14	**0.15**	0.04	0.05	0.04	0.05	0.01	− 0.11
Facebook	− 0.02	− 0.03	− 0.02	0.08	− 0.02	0.08	**0.09**	0.03
Twitter	− 0.05	− 0.07	0	**0.09**	0	0.09	0.09	0.02
Authors	− 0.11	− 0.12	− 0.11	− 0.09	− 0.11	− 0.09	0.07	**0.14**
BitCoin Alpha	0	− 0.04	0.02	0.05	0.02	0.06	**0.08**	0.05
BitCoin OTC	0.01	0.01	0.03	0.04	0.03	0.04	**0.05**	0.02
Actors	0.11	0.12	0.05	0.26	0.06	**0.27**	0.23	0.04
Facebook-like Forum	0.03	0.01	− 0.07	0.16	0.01	0.18	0.32	**0.33**
Newman's collab. network	− 0.02	− 0.02	− 0.01	0.02	− 0.01	0.02	0.09	**0.10**
Average	0.10	0.08	0.05	0.13	0.06	0.14	**0.19**	0.14

The best scoring method for each dataset is highlighted in bold

Table 3 Running time (s) for the different methods

Network	STCb [GREEDY]	Hochbaum Naor	LP1	LP2
Kangaroo	0.01	0.43	0.56	0.34
Southern Women	0.01	0.01	0.39	0.37
Terrorists	0.01	0.07	0.4	0.37
Beach	0.02	0.07	0.48	0.49
Freeman 1	0.01	0.06	0.38	0.4
Freeman 2	0.01	0.03	0.44	0.6
Cross Parker Consulting	0.01	0.05	0.5	0.7
Les Miserables	0.02	0.03	0.39	0.36
Students	0.4	0.83	3.36	3.01
Facebook-like Social	5.14	11.02	282	306
KDD	1.99	4.8	8.8	7.76
ICDM	1.71	4.17	8.07	8.7
Facebook	26	105	46	39
Twitter	68	198	3 924	3 861
Authors	9.59	24.5	19	23.6
BitCoin Alpha	6.44	14.5	37	45
BitCoin OTC	14.45	35	93	98
Actors	111	227	4 323	4 401
Facebook-like Forum	51	110	2 450	2 377
Newman collab. network	14	43	32	31

Total times include problem construction for the LP's, graph construction for Hochbaum–Naor, and wedge-graph construction for STCbinary [GREEDY]

into account, while still being cautious with the assignment of strong edges, leading to improved performance.

6.3 Runtime analysis

Tabel 3 shows the running time of methods STCbinary [GREEDY], LP1, LP2 ($d = 1$) and the combinatorial algorithm of Hochbaum–Naor ($d = 1$). It demonstrates the superior performance of the latter when compared to the LP solvers, especially on the more challenging networks like Actors and Facebook-like Forum. The running time of STCbinary [MM] is comparable to STCbinary [GREEDY] and is not reported here. Remarkably, the Hochbaum–Naor algorithm performs very comparably to STCbinary [GREEDY].

6.4 Discussion of the Southern Woman and the Terrorists datasets

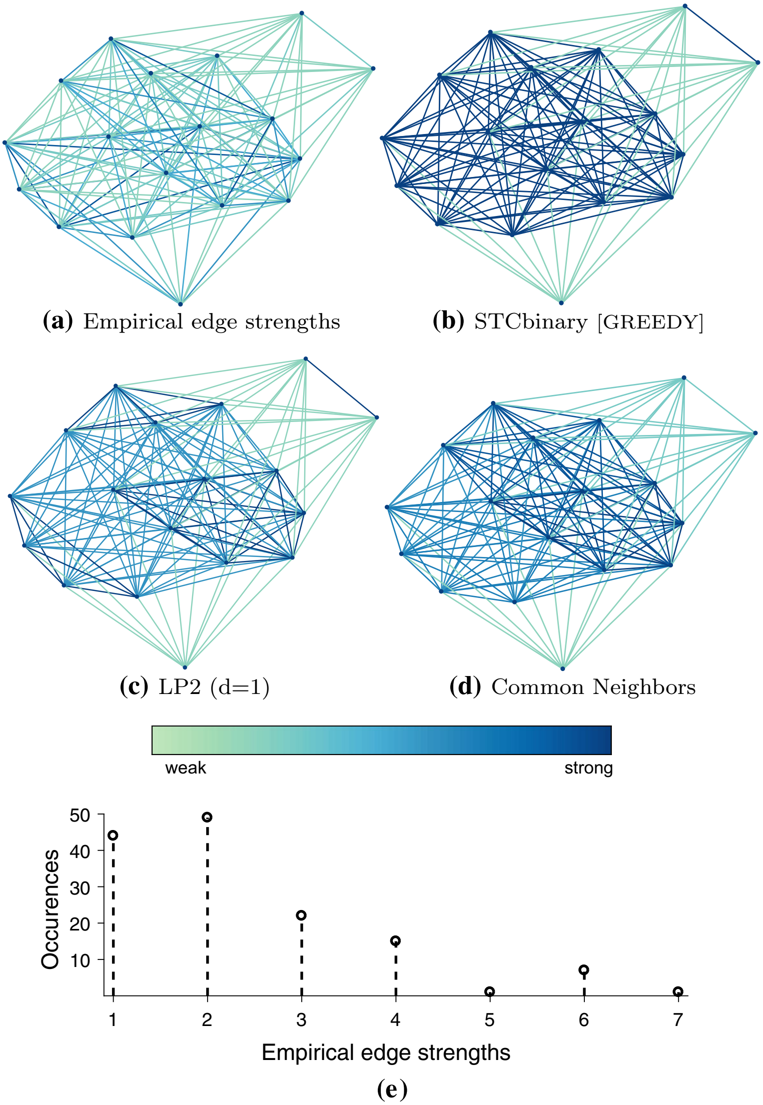
For a better understanding, we visualize two datasets, one where our methods perform well and one where our methods perform poorly. Figure 6 shows a visualization of the methods STCbinary [GREEDY], LP2 and Common Neighbors on the Southern Women dataset. It shows the edge strengths as inferred by each of the methods. The dataset recorded the attendance at 14 social events of 18 women is the 1930s. The empirical edge strengths denote the number of co-attendances between pairs of women, recorded over a 9 month period. Figure 6e shows the distribution of the empirical edge strengths. This dataset has a natural tendency for STC: if woman A has numerous co-attendences with both woman B and C, then there is a high chance that also woman B and C have atleast one co-attendence, especially when the number of social events is small. Method LP2 has the highest Kendall τ_b score (0.45) of the different methods. On this example, LP2 can be seen as a refinement of method STCbinary [GREEDY]: only a small set of edges in the central cluster are labeled as strong (the trianglecliques), while the rest of the clusteredges have an intermediate score. All but one edge (i.e. the edge connecting them) coming from the three outliers are labeled as weak.

On the other hand, Fig. 7 shows the methods' results on the Terrorists dataset. The network contains the suspected terrorists involved in the train bombing of Madrid on March 11, 2004 as reconstructed from newspapers. The edge weights denote how strong a connection was between the terrorists (friendship, co-participating in previous attacks, etc.). As with the Southern Women data, also this network seems to have an intuitive tendency for STC. This is confirmed by Table 1. However, our methods perform quite badly on this dataset. Figure 7e shows that there are very few strong edges in the network. Method LP1(Δ) fails to recognize most of these edges as strong, it assigns them an intermediate 1/2- score most of the time. Method STCbinary [MM] also incorrectly labels a lot of weak edges as strong. However, most of the strong edges are indeed labeled as strong, leading to better performance than the LP methods. As discussed in Sect. 3.3, one can argue that for this dataset the objective is not well-suited: it might not be justified to maximize the number of strong edges (or triangles), given that there are very few strong edges (or triangles) present in the empirical data, see Fig. 7e. Common Neighbors doesn't overscore too much, and was the best performing method, with a Kendall τ_b score of 0.32.

7 Related work

This work builds on the STC principle for edge strength inference in (social) networks. The STC principle was first suggested by sociologist Simmel (1908), and later popularized by Granovetter (1977). The book of Easley and Kleinberg (2010) discusses the STC property in more detail, studying the effect of the property on certain structural properties, such as the clustering coefficient. They claim that the STC property is too extreme to hold true for large, complex networks, but suggested that it can be a useful simplification of reality that can be used to understand and predict networks.

Sintos and Tsaparas (2014) were the first to cast this into a **NP**-hard optimization problem (STCbinary) for edge strength inference: they propose a {`weak`, `strong`} labeling that maximizes the number of strong edges, without violating the STC property. Our work proposes a number of LP relaxations of STCbinary. The main advantages are that our methods are solvable in polynomial time, solutions are less arbitrary w.r.t. the graph structure, and they allow for more fine-grained levels of predicted tie strengths.


(a) Empirical edge strengths **(b)** STCbinary [GREEDY]

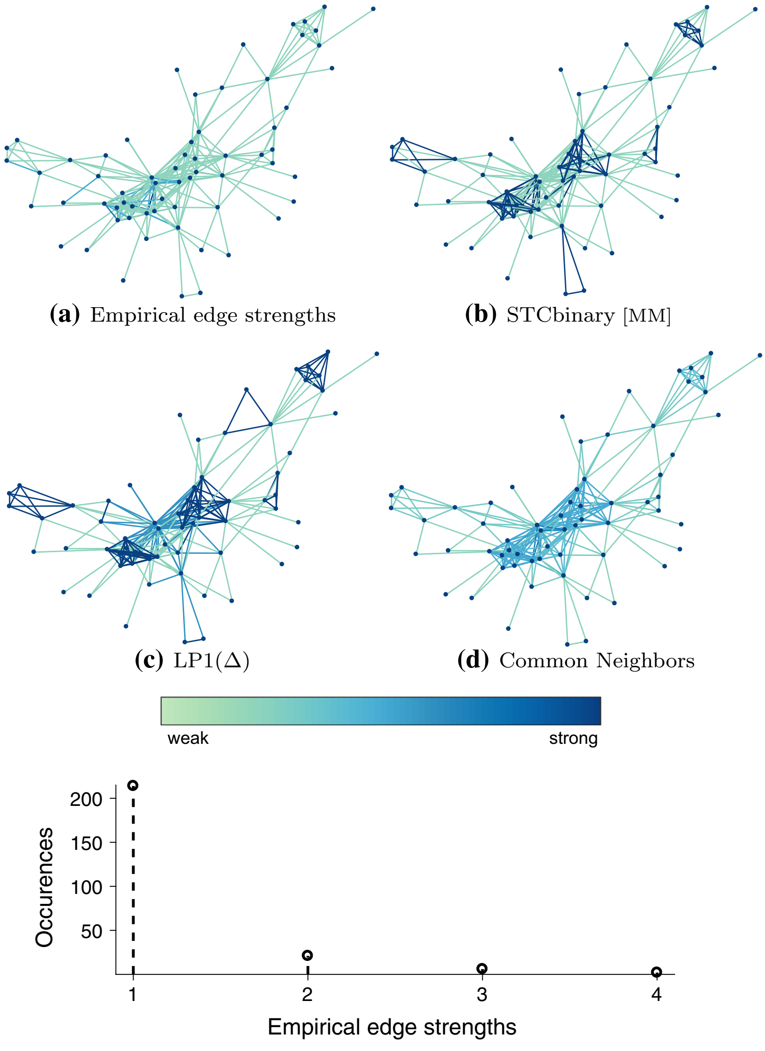
(c) LP2 (d=1) **(d)** Common Neighbors

(e)

Fig. 6 A case where our methods perform well. Southern Women dataset

Another recent extension of the work of Sintos and Tsaparas (2014) is is the work of Rozenshtein et al. (2017). They consider a {weak, strong} labeling with additional community connectivity constraints, and allowing for a small number of STC violations to satisfy those constraints.

All of these methods can also be seen as part of a broader line of active research aiming to infer the strength of ties in (social) networks. Excellent surveys of the existing methods for this general problem are given by Hasan and Zaki (2011); Rossi et al. (2012). These methods can roughly be put into two different categories:



(a) Empirical edge strengths

(b) STCbinary [MM]

(c) LP1(Δ)

(d) Common Neighbors

(e)

Fig. 7 A case where our methods perform poorly. Terrorists dataset

- *Methods that make use of various meta-data* Jones et al. (2013) use frequencies of online interactions to predict the strength of ties with high accuracy. Gilbert and Karahalios (2009) characterize social ties based on similarity and interaction information. Similarly, Xiang et al. (2010) estimate relationship strength from the homophily principle and interaction patterns and extend the approach to heterogeneous types of relationships. Pham et al. (2016) incorporate spatio-temporal features of social interactions to increase accuracy of inferred tie strengths. Fire et al. (2011) use topological features to build a supervised learning classifer to identify missing links (although they are concerned with link prediction, their method can be applied to the edge strength classification setting).

- *Methods that only use structural information* Most well-known heuristics such as Common Neighbors, Jacard Index, Preferential Attachment, etc. fall in this category. A good overview of all the different heuristics are given by Hasan and Zaki (2011). Gupte and Eliassi-Rad (2011) measure the tie strength between individuals, given their attendence to mutual events. They propose a list of axioms that a measure of tie strength must satisfy.

Network Embedding methods (Grover and Leskovec 2016; Hamilton et al. 2017; Kang et al. 2019) map the nodes of a network into a low-dimensional Euclidean space. The mapping is such that 'similar' nodes are mapped to nearby points. The notion of 'similarity' can be based on topological features, or on additional information, such that these methods can be categorized similarly as discussed above. Zhou et al. (2018) learn embedding vectors for nodes by capturing evolutionary structual properties of a network. In particular, they propose a framework that quantifies the probability of an open triple developing into a closed triple. They use the empirical edge weights as an input variable of their embedding. Most of these methods are known to have excellent performance in link prediction and node classification problems. However, it is unclear how they perform when the downstream task is edge strength inference.

Another related research area is *Edge Role Discovery* in a network. Ahmed et al. (2017) leverage higher-order network motifs (graphlets) in an unsupervised learning settings, in order to capture edges that are 'similar'. However, the inference about the (relative) strength of the different assigned roles is not clear. Tang et al. (2012, 2011, 2016) propose a generative statistical model, which can be used to classify heterogeneous relationships. The model relies on social theories and incorporates structural properties of the network and node attributes. Their more recent works can also compute strengths of the predicted types of ties. Backstrom and Kleinberg (2014) focuses on the graph structure to identify a particular type of ties — romantic relationships in Facebook. Role discovery for nodes (Rossi and Ahmed 2014; Henderson et al. 2012) is less relevant to our work, since it deals with classification of nodes and not edges.

8 Conclusions and further work

8.1 Conclusions

We have proposed a sequence of Linear Programming (LP) relaxations of the **NP**-hard problem introduced by Sintos and Tsaparas (2014). These formulations have a number of advantages, most notably their computational complexity, they allow for multi-level strength inferences, and they make less arbitrary strength assignments w.r.t. the graph structure. Additionally, instead of maximizing the number of strong edges, we can change the objective of these LP's to more meaningful objectives (Sect. 3.3). This comes without a loss of the above discussed advantages.

Section 4 provides an extensive theoretical discussion of these LP's, providing half-integrality results and discussing the symmetries present in certain optimal solutions.

Section 5.2 discusses a fast combinatorial algorithm that can be used to solve the main LP relaxations.

The main goal of both the LP relaxations and the original **NP**-hard problem, is to infer the strengths of ties in (social) networks by leveraging the *Strong Triadic Closure* (STC) property. The empirical evaluation (Sect. 6) shows that the methods for both problems are often outperformed by some simple but powerful baselines, such as Common Neighbors or Preferential Attachment. This is interesting, because Table 1 indicates a presence of the STC property in almost all datasets: strong triples are closed more often than normal triples. We believe two important factors of the rather poor performance of the STC methods are the following:

- Most datasets used in this paper have heavily right-skewed distributions of their empirical edge strengths. It might not be justified to maximize the number of strong edges (or triangles), given that there are very few actual strong edges (or triangles) present in the data. See Fig. 7 for the Terrorists dataset as an example. The fact that the distributions are often heavily right-skewed, could have a temporal cause: some (social) networks are snapshots of an evolving process, and strong ties take time to form.[7]
- Only leveraging the STC principle for edge strength inference may be a bit too ambitious. Although STC is present in most datasets, exclusively leveraging this principle may fail to capture other important structural causes that lead to the devolpment of strong edges in networks (f.e., node degrees, common neighbors, community structure, other external information etc.).

This raises doubts about the usefulness of the STCbinary problem in real-life networks.

8.2 Further work

Our research results open up a large number of avenues for further research.

A first line of research is to investigate alternative problem formulations. An obvious variation would be to take into account community structure, and the fact that the STC property probably often fails to hold for wedges that span different communities. A trivial approach would be to simply remove the constraints for such wedges, but more sophisticated approaches could exist. Additionally, it would be interesting to investigate the possibility to allow for different relationship types and respective edge strengths, requiring the STC property to hold only within each type. Furthermore, the fact that the presented formulations are LPs, combined with the fact that many graph-theoretical properties can be expressed in terms of linear constraints, opens up the possibility to impose additional constraints on the optimal strength assignments without incurring significant computational overhead as compared to the interior point implementation. One line of thought is to impose upper bounds on the sum of edge strengths incident to any given edge, modeling the well-known fact that an individual is limited in how many strong social ties they can maintain. Another interesting question

[7] 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.

is whether we can leverage higher-order graphlets (Ahmed et al. 2015) in the LP formulations, beyond wedges and triangles. It's also natural to investigate how to leverage available temporal information: can we use STC to predict the strength of future edges, based on edges that were formed in the past?

A second line is to investigate whether more efficient algorithms can be found for inferring the range of edge strengths across the optimal face of the feasible polytope. A related research question is whether the marginal distribution of individual edge strengths, under the uniform distribution of the optimal polytope, can be characterized in a more analytical manner (instead of by uniform sampling). Both these questions seem important beyond the STC problem, and we are unaware of a definite solution to them.

A third line of research is whether an active learning approach can be developed, to quickly reduce the number of edges assigned an intermediate strength by our approaches.

Finally, perhaps the most important line of further research concerns the manner in which the STC property is leveraged for edge strength inference: could it be modified so as to become more widely applicable across real-life social networks?

Acknowledgements The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC Grant Agreement no. 615517, from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, from the FWO (project nos. G091017N, G0F9816N), from the European Union's Horizon 2020 research and innovation programme, 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).

Appendix: Proofs

Lemma 1.

Proof Nodes connected by a triangle edge have the same set of neighbors. Indeed, otherwise the supposed triangle edge would be part of a wedge with that non-common neighbor as one of its end-points. Conversely, an edge between nodes that have the same sets of neighbors is a triangle edge by definition. Thus, "being connected by a triangle edge" is equivalent with "having the same set of neighbors," which defines an equivalence relation over the nodes of the graph. This equivalence relation allows one to divide the nodes into equivalence classes, within which all nodes are connected by a triangle edge. Each of these equivalence classes corresponds to a fully connected component in the subgraph induced by all triangle edges. □

Lemma 2.

Proof This follows directly from the fact that "being connected by a triangle edge" is equivalent with "having the same set of neighbors," as shown in the proof of Lemma 1.

Alternatively, it can be proven directly by contradiction as follows. Assume the contrary, that is, that some node $k \in V \backslash C$ is connected to $i \in C$ but not to $j \in C$. This means that $(i, \{j, k\}) \in \mathcal{W}$. However, this contradicts the fact that $\{i, j\}$ is triangle edge. □

Lemma 3.

Proof Consider any triangle clique in the graph. Then either this triangle clique is a connected component on its own (a situation we excluded), or it has at least one bundle. As rays are wedge edges, and the bundle contains a ray for each node in the triangle clique, this means that each triangle edge is adjacent to a wedge edge. □

Proposition 1.

Proof The weight of wedge edges is trivially bounded by 1. From Lemma 3, we know that the weight of each triangle edge is bounded by a at least one triangle inequality where the strength of the edge on the right hand side is the strength of a wedge edge—i.e., it is also bounded by a finite number, thus proving the theorem. □

Proposition 2.

Proof From weak duality, we know that if the dual is feasible, then the primal is bounded. Hence, it suffices to construct one feasible point for the dual LP in order to show boundedness of the primal solution.

To do this, we first transform LP4 to the symmetric form by substituting $w_{ij} + \frac{1}{d} = x_{ij}$ for all $\{i, j\} \in E \cup \bar{E}$:

$$\begin{aligned}
&\max_{\mathbf{w}} \sum_{\{i,j\}\in E} x_{ij} - C \sum_{\{j,k\}\in \bar{E}} x_{jk} + \frac{1}{d}(C|\bar{E}| - |E|), \\
&\text{such that } x_{ij} + x_{ik} - d \cdot x_{jk} \le 1 + \frac{2}{d}, && \text{for all } (i, \{j,k\}) \in \mathcal{W}, \\
&\qquad x_{ij} + x_{ik} - d \cdot x_{jk} \le 1 + \frac{2}{d}, && \text{for all } \{i, j, k\} \in \mathcal{T}, \\
&\qquad x_{ij} \ge 0, && \text{for all } \{i, j\} \in E. \\
&\qquad x_{jk} \ge 0, && \text{for all } \{j, k\} \in \bar{E}. \quad (23)
\end{aligned}$$

For each wedge constraint we introduce the dual variable $y^{w}_{i,jk}$. For each triangle constraint that lower bounds x_{jk}, we introduce the dual variable $y^{t}_{i,jk}$. I.e. $y^{t}_{i,jk}$ refers to the triangle constraint associated with triangle $\{i, j, k\}$ where x_{jk} is multiplied with $-d$. The constraints of the dual LP of (23) are then given by:

$$\sum_{\substack{(i,\{j,k\})\in\mathcal{W},\\ e=\{i,j\}\vee\{i,k\}}} y^{w}_{i,jk} + \sum_{\substack{\{i,j,k\}\in\mathcal{T},\\ e=\{j,k\}}} \left(y^{t}_{j,ik} + y^{t}_{k,ij} - d y^{t}_{i,jk} \right) \ge 1, \quad \text{for all } e \in E, \quad (24)$$

$$\sum_{\substack{(i,\{j,k\})\in\mathcal{W},\\ e=\{j,k\}}} y^w_{i,jk} \le \frac{C}{d}, \qquad \text{for all } e \in \bar{E},$$

$$y^w_{i,jk}, y^t_{i,jk} \ge 0. \tag{25}$$

Let M be the maximum number of wedges in the graph that have the same end-points, and let L be the maximum number of triangles any edge in the graph is part of. Then, we will prove that the following is a dual feasible solution for sufficiently large C:

$$\begin{aligned} y^w_{i,jk} &= \frac{C}{Md} && \text{for all } w \in \mathcal{W},\\ y^t_{i,jk} &= 0 && \text{for } \{j,k\} \text{ a triangle edge,}\\ y^t_{i,jk} &= 1 && \text{for } \{j,k\} \text{ a wedge edge.} \end{aligned}$$

The constraints (25) are trivially satisfied for all $e \in \bar{E}$.

To show that constraints (24) are satisfied, we distinguish two cases: the case where e is a wedge edge, and the case where e is a triangle edge.

For $e \in E$ a wedge edge, the left summation in constraints (24) contains at least one term and is lower bounded by $\frac{C}{Md}$, while the right summation is lower bounded by $-Ld$ (it contains at most L terms that are each lower bounded by $-d$). Thus, the constraint is satisfied as long as $C \ge Md(1 + Ld)$.

For $e \in E$ a triangle edge, the left summation in constraints (24) disappears. In the right summation, all terms $-dy^t_{i,jk} = 0$ as $\{j,k\}$ is a triangle edge, such that all terms in this summation are positive. Moreover, from Lemma 3 we know that each triangle edge $e = \{j,k\}$ is adjacent to at least one wedge edge $\{i,j\}$. By definition of a triangle edge, $\{i,j,k\}$ must then form a triangle, and thus give rise to a term in the right summation. As $y^t_{k,ij} = 1$, this term, and therefore the entire summation, is lower bounded by 1 as required. □

Proposition 3.

Proof Each permutation $\alpha \in \Pi$ is an automorphism of G. This follows directly from Lemma 2 and the fact that α only permutes nodes *within* each triangle clique. Furthermore, it is clear that if $\alpha \in \Pi$ then also $\alpha^{-1} \in \Pi$, and if $\alpha_1, \alpha_2 \in \Pi$ then also $\alpha_1\alpha_2 \in \Pi$. Finally, Π contains at least the identity and is thus non-empty, proving that Π is a subgroup of $\mathcal{A}$. □

Corollary 2.

Proof A strength of 0 for a triangle edge can never be optimal, as triangle edges are upper bounded by at least 2 for $d \ge 1$, and the objective function is an increasing function of the edge strengths. The second statement follows from the fact that $\frac{2}{2-d}$ is the smallest possible value for triangle edges when $d < 1$, and Eq. (12) bounds the triangle edge strengths in triangle cliques A with $|A| \ge 3$ to that value. Thus, it is the only possible value for the vertex points of the optimal face of the feasible polytope, and thus for that entire optimal face. □

Theorem 1.

Proof Let w_{ij} be the optimal strength for the node pair $\{i, j\}$ in an optimal solution $\mathbf{w}$. Then, we claim that assigning a strength $\frac{1}{|\mathcal{A}|}\sum_{\alpha\in\mathcal{A}} w_{\alpha(i)\alpha(j)}$ to each node pair $\{i, j\}$ is also an optimal solution. This solution satisfies the condition in the theorem statement, so if true, the theorem is proven.

It is easy to see that this strength assignment has the same value of the objective function. Thus, we only need to prove that it is also feasible.

As α is a graph automorphism, it preserves the presence of edges, wedges, and triangles (e.g., $\{i, j\} \in E$ if and only if $\{\alpha(i), \alpha(j)\} \in E$). Thus, if a set of strengths w_{ij} for node pairs $\{i, j\}$ is a feasible solution, then also the set of strengths $w_{\alpha(i)\alpha(j)}$ is feasible for these node pairs. Due to convexity of the constraints, also the average over all α of these strengths is feasible, as required. □

Theorem 2.

Proof We will prove the claim by contradiction. Consider an optimum $\mathbf{w}$ for which this is not the case, i.e., two adjacent triangle edges can be found that have different strength. From Corollary 1 we know that we can construct from this optimal solution another optimal solution $\mathbf{w}^=$ for which adjacent triangle edges do have the same strength, equal to the average strength in $\mathbf{w}$ of all triangle edges in the triangle clique they are part of. Moreover, in $\mathbf{w}^=$ all rays within the same bundle have the same strength, equal to the average strength in $\mathbf{w}$ of all rays in the bundle. Let us denote the strength in $\mathbf{w}^=$ of the b-th bundle to the triangle clique as $w_b^=$ (i.e., b is an index to the bundle), and the strength of the edges in the triangle clique as $w_c^=$. We will prove that $\mathbf{w}^=$ is not optimal, reaching a contradiction.

In particular, we will show that there exists a solution $\mathbf{w}^*$ for which $w_b^* = w_b^=$ for all bundles b, but for which the strength within the triangle clique is strictly larger: $w_c^* > w_c^=$. We first note that the strengths of the triangle edges w_c^* are bounded in triangle constraints involving two rays and one triangle edge, namely $w_c^* \leq 2 + (d-1)\cdot w_b^=$. They are bounded also in triangle constraints involving only triangle edges, namely $(2-d)\cdot w_c^* \leq 2$. For $d \geq 2$ this constraint is trivially satisfied, but not for $d < 2$. Thus, we know that $w_c^* = 2 + \min_b \left\{(d-1)\cdot w_b^=\right\}$ for $d \geq 2$, and $w_c^* = \min\left\{2 + \min_b \left\{(d-1)\cdot w_b^=\right\}, \frac{2}{2-d}\right\}$ for $d < 2$. If this optimal value for w_c^* is larger than $w_c^=$ the contradiction is established.

First we show that $w_c^= < \frac{2}{2-d}$ when $d < 2$, again by contradiction. For each triangle $\{i, j, k\}$ in the triangle clique, the following triangle inequality is the tightest: $\max\{w_{ij}, w_{ik}, w_{jk}\} + \text{median}\{w_{ij}, w_{ik}, w_{jk}\} \leq 2 + d\cdot\min\{w_{ij}, w_{ik}, w_{jk}\}$. Averaging these constraints over all triangles within the triangle clique, we obtain:

$$w^+ + w^0 \leq 2 + d\cdot w^- \tag{26}$$

for some $w^+ \geq w^0 \geq w^-$ for which $w_c^= = \frac{1}{3}(w^+ + w^0 + w^-)$. Since we assumed (with the intention to reach a contradiction) that not all w_{ij} in the triangle clique are equal, we also know that $\frac{w^+ + w^0}{2} > w_c^= > w^-$. Given this, and if it were indeed the case that $w_c^{=} \geq \frac{2}{2-d}$, Eq. (26) would imply that $2 \cdot \frac{2}{2-d} < 2 + d \cdot w^{-}$, and thus, $\frac{2}{2-d} < w^{-}$, a contradiction.

Next, we show that $w_c^{=} < 2 + \min_b \left\{(d-1) \cdot w_b^{=}\right\}$. To show this, we need to distinguish two cases:

1. *The bundle b with smallest $(d-1) \cdot w_b^{=}$ has at least two different weights in* $\mathbf{w}$. Then, note that for each pair of ray strengths w_{bi} and w_{bj} from node b to triangle-edge (i, j), the following triangle constraints must hold: $w_{ij} \leq 2 + d \cdot \min\{x_{bi}, x_{bj}\} - \max\{x_{bi}, x_{bj}\}$. Summing this over all $\{i, j\}$ and dividing by $\frac{n(n-1)}{2}$ where n is the number of nodes in the triangle clique, yields: $w_c^{=} \leq 2 + d \cdot w_b^{-} - w_b^{+}$ for some $w_b^{-} < w_b^{+}$ with $w_b = \frac{w_b^{-} + w_b^{+}}{2}$. This means that $w_c^{=} < 2 + \min_b \{(d-1) \cdot x_b\}$, with a strict inequality since we assumed that there is at least one pair of rays $\{b, i\}$ and $\{b, j\}$ for which $w_{bi} < w_{bj}$. Thus, this shows that $w_c^{*} > w_c^{=}$, and a contradiction is reached.
2. *All rays in the bundle b with smallest $(d-1) \cdot w_b^{=}$ have equal strength $w_b = w_b^{=}$ in* $\mathbf{w}$. In this case, we know that $w_{ij} \leq 2 + (d-1) w_b^{=}$ (due to feasibility of the original optimum). Again, averaging this over all triangle edges $\{i, j\}$, yields: $w_c^{=} \leq 2 + (d-1) w_b^{=}$, with equality only if all terms are equal (since the right hand side is independent of i and j). Thus, again a contradiction is reached.

□

Theorem 3.

Proof It is easy to see that for a symmetric solution, the objective functions of Problems LP2SYM and LP2 are identical.

Thus, it suffices to show that:

1. the feasible region of Problem LP2SYM is contained within the feasible region of Problem LP2,
2. the set of *symmetric* feasible solutions of Problem LP2 is contained within the feasible region of Problem LP2SYM.

The latter is immediate, as all constraints in Problem LP2SYM are directly derived from those in Problem LP2 (see rest of the proof for clarification), apart from the reduction in variables which does nothing else than imposing symmetry.

To show the former, we need to show that all constraints of Problem LP2 are satisfied. This is trivial for the positivity constraints (13) and (14). The wedge inequalities (10) are also accounted in Problem LP2SYM, and thus trivially satisfied, too.

We consider three types of triangle constraints in Problem LP2: those involving two rays from the same bundle and one triangle edge with the triangle edge strength on the left hand side of the $<$ sign, those involving two rays from the same bundle and one triangle edge with the triangle edge strength on the right hand side of the $<$ sign, and those involving three triangle edges.

Constraint (11) covers all triangle constraints involving two rays (from the same bundle) and one triangle edge, with the triangle edge strength being upper bounded. Indeed, w_A^t is the strength of the triangle edges between nodes in A, and w_{AB}^w is the strength of the edges in the bundle from any node in B to the two nodes connected by any triangle edge in A.

Triangle constraints involving two rays and one triangle edge that lower bound the triangle edge strength are redundant and can thus be omitted. Indeed, they can be stated as $w^w_{AB} + w^w_{AB} \leq 2 + d \cdot w^t_A$, which is trivially satisfied as each wedge edge has strength at most 1.

Finally, triangle constraints involving three triangle edges within A reduce to $w^t_A + w^t_A \leq 2 + d \cdot w^t_A$. For $d \geq 2$ this constraint is trivially satisfied. For $d < 2$ it reduces to $w^t_A \leq \frac{2}{2-d}$. For $2 > d \geq 1$, this constraint is also redundant with the triangle constraints involving the triangle edge and two rays, which imply an upper bound of at most $d + 1 \leq \frac{2}{2-d}$ for $d \geq 1$ (namely for the ray strengths equal to 1). Thus, constraint $w^t_A \leq \frac{2}{2-d}$ must be included in Problem LP2SYM only for $d < 1$. Finally, note that such triangle constraints are only possible in triangle cliques A with $|A| \geq 3$. □

Theorem 4.

Proof Assume the contrary, i.e., that a vertex point of this convex feasible polytope can be found that has a different value for one of the edge strengths. To reach a contradiction, we will take such vertex point $\mathbf{w}^t, \mathbf{w}^w$ and nudge the *wedge edges'* strengths w^w_{AB} as follows:

- if $w^w_{AB} \in (0, \frac{1}{2})$, add ϵ,
- if $w^w_{AB} \in (\frac{1}{2}, 1)$, subtract ϵ.

Note that $0 \leq w^w_{AB} \leq 1$ for all wedge edges (due to the wedge constraints in Eq. (10)). Thus, all wedge edge strengths that are not exactly equal to 0, $\frac{1}{2}$, or 1 will be nudged.

For the *triangle edges* we need to distinguish between $d \geq 1$ and $d < 1$. For $d \geq 1$, we nudge their strengths as follows:

- if $w^t_A \in (0, 2)$, add ϵ,
- if $w^t_A \in \left(2, 2 + (d-1) \cdot \frac{1}{2}\right) = \left(2, \frac{d+3}{2}\right)$, add $(d-1)\epsilon$,
- if $w^t_A \in \left(2 + (d-1) \cdot \frac{1}{2}, 2 + (d-1) \cdot 1\right) = \left(\frac{d+3}{2}, d+1\right)$, subtract $(d-1)\epsilon$.

For $d < 1$, we nudge the strenghts as follows:

- if $w^t_A \in \left(0, \frac{2}{2-d}\right) \cup \left(\frac{2}{2-d}, d+1\right)$, add ϵ,
- if $w^t_A \in \left(2 + (d-1) \cdot 1, 2 + (d-1) \cdot \frac{1}{2}\right) = \left(d+1, \frac{d+3}{2}\right)$, subtract $(d-1)\epsilon$.
- if $w^t_A \in \left(2 + (d-1) \cdot \frac{1}{2}, 2\right) = \left(\frac{d+3}{2}, 2\right)$, add $(d-1)\epsilon$.

Note that for $d \geq 1$, it holds that $0 \leq w^t_A \leq d + 1$ for all triangle edges (due to the triangle constraints in Eq. (11) with ray strength equal to 1). For $d < 1$, it holds that $0 \leq w^t_A \leq 2$ for all triangle edges (due to the triangle constraints in Eq. (11) with ray strength equal to 0). Thus, also all triangle edge strengths that are not of one of the values specified in the theorem statement will be nudged.

For sufficiently small $|\epsilon|$ no loose constraint will become invalid by this. Furthermore, it is easy to verify that strengths in tight constraints are nudged in corresponding directions, such that all tight constraints remain tight and thus valid. Now, this nudging can be done for positive and negative ϵ, yielding two new feasible solutions of which the average is the supposed vertex point of the polytope—a contradiction. □

Theorem 5.

Proof Let us denote the optimal vector of weights found by solving Problem LP2FULLSYM as $\mathbf{w}^*$. It is clear that $\mathbf{w}^*$ is an optimal solution to Problem LP2, as it is constrained to be such.

Now, we will prove symmetry by contradiction: let us assume there is a graph automorphism $\alpha \in \mathcal{A}$ with respect to which it is not symmetric, such that there exists a set of edges $\{i, j\} \in E$ for which $w^*_{ij} \neq w^*_{\alpha(i)\alpha(j)}$. Due to convexity, $\mathbf{w}^{**}$ with $w^{**}_{ij} = \frac{w^*_{ij}+w^*_{\alpha(i)\alpha(j)}}{2}$ is then also a solution to Problem LP2FULLSYM and thus to Problem LP2. However, since $a^2+b^2 > 2\left(\frac{a+b}{2}\right)^2$ for any $a \neq b \in \mathbb{R}$, $\mathbf{w}^{**}$ has a smaller value for the objective of Problem LP2FULLSYM, such that $\mathbf{w}^*$ cannot be optimal—a contradiction. □

References

Ahmed N, Neville J, Rossi R, Duffield N (2015) Efficient graphlet counting for large networks, pp 1–10. https://doi.org/10.1109/ICDM.2015.141

Ahmed NK, Rossi RA, Willke TL, Zhou R (2017) Edge role discovery via higher-order structures. In: Kim J, Shim K, Cao L, Lee JG, Lin X, Moon YS (eds) Advances in knowledge discovery and data mining. Springer, Cham, pp 291–303

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

Backstrom L, Kleinberg J (2014) Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on Facebook. In: Proceedings of the 17th ACM conference on computer supported cooperative work & social computing. ACM, pp 831–841

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Chen Y, Dwivedi R, Wainwright MJ, Yu B (2017) Fast mcmc sampling algorithms on polytopes. arXiv:1710.08165

De Meo P, Ferrara E, Fiumara G, Provetti A (2014) On facebook, most ties are weak. Commun ACM 57(11):78–84. https://doi.org/10.1145/2629438

Easley D, Kleinberg J (2010) Networks, crowds, and markets: reasoning about a highly connected world. Cambridge University Press, New York

Edelsbrunner H, Rote G, Welzl E (1989) Testing the necklace condition for shortest tours and optimal factors in the plane. Theor Comput Sci 66(2):157–180

Fire M, Tenenboim L, Lesser O, Puzis R, Rokach L, Elovici Y (2011) Link prediction in social networks using computationally efficient topological features. In: 2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing, pp 73–80. https://doi.org/10.1109/PASSAT/SocialCom.2011.20

Gilbert E, Karahalios K (2009) Predicting tie strength with social media. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 211–220

Granovetter MS (1977) The strength of weak ties. In: Social networks. Elsevier, pp 347–367

Grant M, Boyd S (2014) CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx

Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. In: Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, New York, NY, USA, KDD'16, pp 855–864. https://doi.org/10.1145/2939672.2939754

Gupte M, Eliassi-Rad T (2011) Measuring tie strength in implicit social networks. In: Proceedings of the 3rd annual ACM web science conference, WebSci'12. https://doi.org/10.1145/2380718.2380734

Hamilton WL, Ying R, Leskovec J (2017) Representation learning on graphs: methods and applications. IEEE Data Eng Bull 40:52–74

Hasan MA, Zaki MJ (2011) A survey of link prediction in social networks. In: Aggarwal C (ed) Social network data analytics. Springer, Boston

Håstad J (1999) Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Math 182(1):105–142

Henderson K, Gallagher B, Eliassi-Rad T, Tong H, Basu S, Akoglu L, Koutra D, Faloutsos C, Li L (2012) Rolx: structural role extraction & mining in large graphs. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining. https://doi.org/10.1145/2339530.2339723

Hochbaum DS (1982) Approximation algorithms for the set covering and vertex cover problems. SIAM J Comput 11(3):555–556

Hochbaum DS (1983) Efficient bounds for the stable set, vertex cover and set packing problems. Discrete Appl Math 6(3):243–254

Hochbaum DS, Naor J (1994) Simple and fast algorithms for linear and integer programs with two variables per inequality. SIAM J Comput 23(6):1179–1192

Jones JJ, Settle JE, Bond RM, Fariss CJ, Marlow C, Fowler JH (2013) Inferring tie strength from online directed behavior. PLoS ONE 8(1):e52168

Kang B, Lijffijt J, De Bie T (2019) Conditional network embeddings. In: International conference on learning representations. https://openreview.net/forum?id=ryepUj0qtX

Kendall MGMG, Dickinson GJ (1990) Rank correlation methods, 5th edn. E. Arnold, London; Oxford University Press, New York. “A Charles Griffin title”. http://www.zentralblatt-math.org/zmath/en/search/?an=0732.62057

Leskovec J, Krevl A (2014) SNAP datasets: Stanford large network dataset collection. http://snap.stanford.edu/data

Lu L, Zhou T (2010) Link prediction in weighted networks: the role of weak ties. https://doi.org/10.1209/0295-5075/89/18001

Mehrotra S, Ye Y (1993) Finding an interior point in the optimal face of linear programs. Math Program 62(1):497–515

Nemhauser GL, Trotter LE (1975) Vertex packings: structural properties and algorithms. Math Program 8(1):232–248

Pham H, Shahabi C, Liu Y (2016) Inferring social strength from spatiotemporal data. ACM Trans Database Syst 41(1):7

Rademacher LA (2007) Approximating the centroid is hard. In: Proceedings of the twenty-third annual symposium on computational geometry, ACM, New York, NY, USA, SCG’07, pp 302–305. https://doi.org/10.1145/1247069.1247123

Rossi R, Ahmed N (2014) Role discovery in networks. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2014.2349913

Rossi RA, McDowell LK, Aha DW, Neville J (2012) Transforming graph data for statistical relational learning. J Artif Int Res 45(1):363–441

Rozenshtein P, Tatti N, Gionis A (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, pp 1017–1025

Simmel G (1908) Soziologie Untersuchungen über die Formen der Vergesellschaftung. Duncker & Humblot, Berlin

Sintos S, Tsaparas P (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, pp 1466–1475

Spielman DA, Teng SH (2004) Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. J ACM 51(3):385–463

Tang W, Zhuang H, Tang J (2011) Learning to infer social ties in large networks. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 381–397

Tang J, Lou T, Kleinberg J (2012) Inferring social ties across heterogenous networks. In: Proceedings of the fifth ACM international conference on Web search and data mining. ACM, pp 743–752

Tang J, Lou T, Kleinberg J, Wu S (2016) Transfer learning to infer social ties across heterogeneous networks. ACM Trans Inf Syst 34(2):7

Viswanath B, Mislove A, Cha M, Gummadi KP (2009) On the evolution of user interaction in Facebook. In: Proceedings of the 2nd ACM SIGCOMM workshop on social networks (WOSN’09)

Xiang R, Neville J, Rogati M (2010) Modeling relationship strength in online social networks. In: Proceedings of the 19th international conference on world wide web. ACM, pp 981–990

Zhou L, Yang Y, Ren X, Wu F, Zhuang Y (2018) Dynamic network embedding by modeling triadic closure process. In: AAAI Conference on Artificial Intelligence. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16572

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Florian Adriaens[1] · **Tijl De Bie**[1] · **Aristides Gionis**[2] · **Jefrey Lijffijt**[1] · **Antonis Matakos**[2] · **Polina Rozenshtein**[2]

Tijl De Bie
TijlDe.Bie@ugent.be

Aristides Gionis
Aristides.Gionis@aalto.fi

Jefrey Lijffijt
Jefrey.Lijffijt@ugent.be

Antonis Matakos
Antonis.Matakos@aalto.fi

Polina Rozenshtein
Polina.Rozenshtein@aalto.fi

[1] IDLab, Department of Electronics and Information Systems, Ghent University, Ghent, Belgium

[2] Department of Computer Science, Aalto University, Espoo, Finland