Received: 21 December 2018 | Revised: 26 August 2019 | Accepted: 26 August 2019 | Published on: 3 October 2019

DOI: 10.1002/net.21907

RESEARCH ARTICLE

WILEY

A modeling and computational study of the frustration index in signed networks

Samin Aref[1,2] | **Andrew J. Mason**[3] | **Mark C. Wilson**[2]

[1]Laboratory of Digital and Computational Demography, Max Planck Institute for Demographic Research, Rostock, Germany
[2]School of Computer Science, University of Auckland, Auckland, New Zealand
[3]Department of Engineering Science, University of Auckland, Auckland, New Zealand

Correspondence
Samin Aref, Laboratory of Digital and Computational Demography, Max Planck Institute for Demographic Research, Rostock, Germany.
Email: sare618@aucklanduni.ac.nz

Funding information
This research was supported by the Open Access funding provided by Projekt DEAL.

The copyright line for this article was changed on 19 October 2020 after original online publication.

Abstract

Computing the frustration index of a signed graph is a key step toward solving problems in many fields including social networks, political science, physics, chemistry, and biology. The frustration index determines the distance of a network from a state of total structural balance. Although the definition of the frustration index goes back to the 1950s, its exact algorithmic computation, which is closely related to classic NP-hard graph problems, has only become a focus in recent years. We develop three new binary linear programming models to compute the frustration index exactly and efficiently as the solution to a global optimization problem. Solving the models with prioritized branching and valid inequalities in Gurobi, we can compute the frustration index of real signed networks with over 15 000 edges in less than a minute on inexpensive hardware. We provide extensive performance analysis for both random and real signed networks and show that our models outperform all existing approaches by large factors. Based on resolution time, algorithm output, and effective branching factor we highlight the superiority of our models to both exact and heuristic methods in the literature.

KEYWORDS

0-1 integer linear programming, balance theory, binary programming, frustration index, line index of balance, optimization, signed graph, signed networks

1 | INTRODUCTION

Local ties between entities lead to global structures in networks. Ties can be formed as a result of interactions and individual preferences of the entities in the network. The dual nature of interactions in various contexts means that the ties may form in two opposite types, namely positive ties and negative ties. In a social context, this is interpreted as friendship versus enmity or trust versus distrust between people. The term *signed network* embodies a multitude of concepts involving relationships characterizable by ties with plus and minus signs. *Signed graphs* are used to model such networks where edges have positive and negative signs. *Structural balance* in signed graphs is a macroscale structural property that has become a focus in network science. Balance theory was the first attempt to understand the sources of tensions and conflicts in groups of people with signed ties [37]. According to balance theory, some structural configurations of people with signed ties lead to social tension and therefore are not balanced.

In network context, if the vertex set of a signed network can be partitioned into $k \leq 2$ subsets such that each negative edge joins vertices belonging to different subsets, it is called a *balanced* network [13]. Using graph-theoretic concepts, Cartwright and Harary identified cycles of the graph as the origins of tension, in particular cycles containing an odd number of negative edges [13]. By definition, signed graphs in which no such cycles are present satisfy the property of structural balance. For graphs

that are not totally balanced, a distance from total balance (a measure of partial balance [7]) can be computed. Among various measures is the *frustration index* that indicates the minimum number of edges whose removal (or equivalently, negation) results in balance [1, 35, 64]. In what follows, we discuss previous works related to the frustration index (also called the *line index of balance* [35]). We use both names, line index of balance and frustration index, interchangeably in this paper. We refer to the time taken by an algorithm on an instance of a problem as *resolution time* (which is equal to *solve time* if the instance is solved).

1.1 | Motivation

In the past few decades, different measures of balance [13, 25, 44, 55, 61] have been suggested and deployed to analyze balance in real-world signed networks resulting in conflicting observations [25, 26, 46]. Measures based on cycles [13, 55], triangles [44, 61], and closed-walks [25] are not generally consistent and do not satisfy key axiomatic properties [7]. Among all the measures, a normalized version of the frustration index is shown to satisfy many basic axioms [7]. This measure provides a clear understanding of the transition to balance in terms of the number of edges to be modified to reduce the tension, as opposed to graph cycles that were first suggested as origins of tension in unbalanced networks [13].

The frustration index is a key to frequently stated problems in many different fields of research [20, 21, 36, 39, 40]. In biological networks, optimal decomposition of a network into monotone subsystems is made possible by computing the frustration index [39]. In finance, performance of a portfolio can be linked to the balance of its underlying signed graph [36]. In physics, the frustration index provides the minimum energy state in models of atomic magnets known as Ising models [40]. In political science [6] and international relations [20], networks can be partitioned into cohesive clusters using the line index of balance. In chemistry, bipartite edge frustration has applications to the stability of fullerene, a carbon allotrope [21]. For discussions on applications of the frustration index, one may refer to [8].

1.2 | Complexity

Computing the frustration index is related to the well-known unsigned graph optimization problem EDGE-BIPARTIZATION, which requires minimization of the number of edges whose deletion makes the graph bipartite. Given an instance of the latter problem, by declaring each edge to be negative we convert it to the problem of computing the frustration index. Since EDGE-BIPARTIZATION is known to be NP-hard [63], so is computing the frustration index. In the converse direction there is a reduction of the frustration index problem to EDGE-BIPARTIZATION which increases the number of edges by a factor of at most 2 [38]. If the reduction preserves planarity, the frustration index can be computed in polynomial time for such planar graphs [33], which is equivalent to the ground state calculation of a two-dimensional spin glass model with no periodic boundary conditions and no magnetic field [18, 30].

The classic graph optimization problem MAXCUT is also a special case of the frustration index problem, as can be seen by assigning all edges to be negative (an edge is frustrated if and only if it does not cross the cut).

1.3 | Approximation

In general graphs, the frustration index is even NP-hard to approximate within any constant factor (assuming Khot's unique games conjecture [41]) [38]. That is, for each $C > 0$, the problem of finding an approximation to the frustration index that is guaranteed to be within a factor of C is believed to be NP-hard.

The frustration index can be approximated to a factor of $\mathcal{O}(\sqrt{\log n})$ [2] or $\mathcal{O}(k \log k)$ [9] where n is the number of vertices and k is the frustration index. Coleman et al. provide a review on the performance of several approximation algorithms of the frustration index [14].

1.4 | Heuristics and local optimization

Doreian and Mrvar have reported numerical values as the line index and suggest that determining this index is in general a polynomial-time hard problem [20]. However, their algorithm does not provide optimal solutions and the results are not equal to the line index of balance [5]. Data-reduction schemes [38] and ground state search heuristics [39] are used to obtain estimates for the frustration index. Facchetti et al. suggested a nonlinear energy function minimization model for finding the frustration index [26]. Their model was analyzed using various techniques [23, 39, 49, 50]. Using the ground state search heuristic algorithms [39], the frustration index is estimated in biological networks with $n \approx 1.5 \times 10^3$ [39] and social networks with $n \approx 10^5$ [26, 27].

1.5 | Exact computation

Using a parametrized algorithmics approach, Hüffner et al. show that the frustration index (under a different name) is *fixed parameter tractable* and can be computed in $\mathcal{O}(2^k m^2)$ [38], where m is the number of edges and k is the fixed parameter (the

frustration index). Binary (quadratic and linear) programming models were recently suggested as methods for computing the exact value of the frustration index [5] capable of processing graphs with $m \approx 10^3$ edges.

1.6 | Related works on a similar problem

Despite the lack of exact computational methods for the frustration index, a closely related and more general problem in signed networks has been investigated comprehensively. According to Davis's definition of *generalized balance*, a signed network is *weakly balanced* (k-balanced) if and only if its vertex set can be partitioned into k subsets such that each negative edge joins vertices belonging to different subsets [17]. The problem of finding the minimum number of frustrated edges for general k (an arbitrary number of subsets) is referred to as the *correlation clustering* problem [10].

For every fixed k, there is a polynomial-time approximation scheme for the correlation clustering problem [32]. For arbitrary k, exact [12, 29] and heuristic methods [22, 47, 48] are developed based on a mixed integer programming model [19]. Denoting the order of a graph by n, exact algorithms fail for $n > 21$ [12] and $n > 40$ [29], while greedy algorithms [22] and local search heuristics [47] are used for larger instances with $n \approx 10^3$ and $n \approx 10^4$ respectively.

After extending the nonlinear energy minimization model suggested by Facchetti et al. [26] to generalized balance, Ma et al. have experimented on the correlation clustering problem in networks with $n \approx 10^5$ using various heuristics [49, 50]. Esmailian et al. have also extended the work of Facchetti et al. [26] focusing on the role of negative ties in signed graph clustering [23, 24].

1.7 | Our contribution

The principal focus of this research study is to provide further insight into computing the frustration index by developing efficient computational methods outperforming previous methods by large factors. We systematically investigate several formulations for exact computation of the frustration index and compare them based on resolution time as well as other performance measures.

The advantage of formulating the problem as an optimization model is not only exploring the details involved in a fundamental NP-hard problem, but also making use of powerful mathematical programming solvers like Gurobi [34] to solve the NP-hard problem exactly and efficiently. We provide numerical results on a variety of undirected signed networks, both randomly generated and inferred from well-known data sets (including real signed networks with over 15 000 edges).

A recent study by the current authors has investigated computing the frustration index in smaller scales using quadratic and linear optimization models [5]. The linear model is used for computing the frustration index in several small random and real networks with up to 3200 edges. We improve the contributions of [5] by providing three new binary linear formulations which not only outperform the models in [5] by large factors, but also facilitate a more direct and intuitive interpretation. We discuss more efficient speed-up techniques that require substantially fewer additional constraints compared to [5]. This allows Gurobi's branch and bound algorithm to start with a better root node solution and explore considerably (several orders of magnitude) fewer nodes leading to a substantially shorter resolution time. Moreover, our new models handle order-of-magnitude larger instances that are not solvable by the models in [5]. We provide in-depth performance analysis using extensive numerical results showing the resolution times of our worst-performing model to be 2-9 times faster than the best-performing model in [5].

This paper begins by laying out the theoretical dimensions of the research in Section 2. Binary linear programming models are formulated in Section 3. Section 4 provides different techniques to improve the formulations and reduce resolution time. The numerical results on the models' performance are presented in Section 5. Section 6 provides comparison against the literature using both random and real networks. Other formulations and extensions to the models are provided in Section 7 followed by Section 8 which sums up the research highlights.

2 | PRELIMINARIES

We recall some standard definitions.

2.1 | Basic notation

We consider undirected signed networks $G = (V, E, \sigma)$. The ordered set of nodes is denoted by $V = \{1, 2, \ldots, n\}$, with $|V| = n$. The set E of edges is partitioned into the set of positive edges E^+ and the set of negative edges E^- with $|E^-| = m^-$, $|E^+| = m^+$, and $|E| = m = m^- + m^+$. The sign function is denoted by $\sigma\colon E \to \{-1, +1\}$.

We represent the m undirected edges in G as ordered pairs of vertices $E = \{e_1, e_2, \ldots, e_m\} \subseteq \{(i, j) \mid i, j \in V, i < j\}$, where a single edge e_k between nodes i and j, $i < j$, is denoted by $e_k = (i, j)$, $i < j$. We denote the graph density by $\rho = 2\,m/(n(n-1))$.

The entries, a_{ij}, of the *signed adjacency matrix*, **A**, are defined in (1).

$$a_{ij} = \begin{cases} \sigma_{(i,j)} & \text{if } (i,j) \in E \\ \sigma_{(j,i)} & \text{if } (j,i) \in E \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

The number of edges incident to the node $i \in V$ represents the degree of node i and is denoted by $d(i)$. A directed cycle (for simplicity *cycle*) of length k in G is a sequence of nodes $v_0, v_1, \ldots, v_{k-1}, v_k = v_0$ such that for each $i = 1, 2, \ldots, k$ there is an edge from v_{i-1} to v_i. The *sign* of a cycle is the product of the signs of its edges. A cycle with negative sign is unbalanced. A balanced cycle is one with positive sign. A balanced graph is one with no negative cycles.

2.2 | Node coloring and frustration count

Satisfied and *frustrated* edges are defined based on colorings of the nodes. Coloring each node with black or white, a frustrated (satisfied) edge (i, j) is either a positive (negative) edge with different colors on the endpoints i, j or a negative (positive) edge with the same colors on the endpoints i, j. Subfigure 1A illustrates an example signed graph in which positive and negative edges are represented by solid lines and dotted lines respectively. Subfigures 1B,C illustrates node colorings and their impacts on the frustrated edges that are represented by thick lines.

Let $X \subseteq V$ be a subset of vertices. This defines a partition $(X, V\backslash X)$ of V. We call X a coloring set. Let binary variable x_i denote the color of node $i \in V$ under coloring set X. We consider $x_i = 1$ if $i \in X$ (black node) and $x_i = 0$ if $i \in V\backslash X$ (white node).

Definition. We define the *frustration count* of signed graph G under coloring X as $f_G(X) := \sum_{(i,j)\in E} f_{ij}(X)$ where $f_{ij}(X)$ is the frustration state of edge (i, j), given by

$$f_{ij}(X) = \begin{cases} 0, & \text{if } x_i = x_j \text{ and } (i,j) \in E^+ \\ 1, & \text{if } x_i = x_j \text{ and } (i,j) \in E^- \\ 0, & \text{if } x_i \neq x_j \text{ and } (i,j) \in E^- \\ 1, & \text{if } x_i \neq x_j \text{ and } (i,j) \in E^+. \end{cases} \tag{2}$$

The optimization problem consists in finding a subset $X^* \subseteq V$ of G that minimizes the frustration count $f_G(X)$, that is, solving Equation (3). The globally optimal solution to this problem gives the frustration index $L(G)$ of signed graph G.

$$L(G) = \min_{X\subseteq V} f_G(X). \tag{3}$$

It follows that $f_G(X)$ gives an upper bound on $L(G)$ for any $X \subseteq V$. Note that the coloring in Figure 1B does not minimize $f_G(X)$, while in Figure 1C $f_G(X)$ is minimum.

3 | BINARY LINEAR PROGRAMMING FORMULATIONS

In this section, we introduce three binary linear programming models in (4)-(6) to minimize the frustration count as the objective function. There are various ways to form the frustration count using variables defined over graph nodes and edges which lead to various mathematical programming models that we discuss in this section.

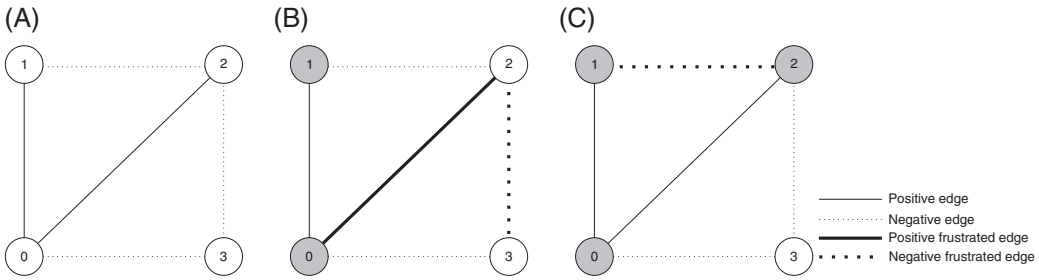

FIGURE 1 Node colorings and the respective frustrated edges for an example signed graph. (A) An example graph with four nodes, two positive edges, and three negative edges. (B) An arbitrary node coloring resulting in two frustrated edges (0, 2), (2, 3). (C) Another node coloring resulting in one frustrated edge (1, 2)

3.1 | The AND model

We start with an objective function to minimize the frustration count. Note that the frustration state of a positive edge (i, j) can be represented by $f_{ij} = x_i + x_j - 2x_ix_j \ \forall (i, j) \in E^+$ using the two binary variables $x_i, x_j \in \{0, 1\}$ for the endpoint colors. For a negative edge, we have $f_{ij} = 1 - (x_i + x_j - 2x_ix_j) \ \forall (i, j) \in E^-$.

The term x_ix_j can be replaced by binary variable $x_{ij} = x_ix_j$ for each edge (i, j) that takes value 1 whenever $\text{AND}_{(x_i,x_j)} = 1$ (both endpoints are colored black) and 0 otherwise. This gives our first binary linear model in (4) that calculates the frustration index in the minimization objective function.

$$\begin{aligned}
\min_{x_i : i\in V, x_{ij} : (i,j)\in E} Z = & \sum_{(i,j)\in E^+} x_i + x_j - 2x_{ij} + \sum_{(i,j)\in E^-} 1 - (x_i + x_j - 2x_{ij}) \\
\text{s.t.} \quad & x_{ij} \le x_i \quad \forall (i,j) \in E^+ \\
& x_{ij} \le x_j \quad \forall (i,j) \in E^+ \\
& x_{ij} \ge x_i + x_j - 1 \quad \forall (i,j) \in E^- \\
& x_i \in \{0,1\} \quad \forall i \in V \\
& x_{ij} \in \{0,1\} \quad \forall (i,j) \in E. \qquad (4)
\end{aligned}$$

The optimal solution represents a subset $X^* \subseteq V$ of G that minimizes the frustration count. The optimal value of the objective function in Equation (4) is denoted by Z^* which represents the frustration index.

The dependencies between the x_{ij} and x_i, x_j values are taken into account using standard AND constraints. The AND model has $n + m$ variables and $2m^+ + m^-$ constraints. Note that x_{ij} variables are dependent variables because of the constraints and the minimization objective function. Therefore, we may drop the integrality constraint of the x_{ij} variables and consider them as continuous variables in the unit interval, $x_{ij} \in [0, 1]$. The next subsection discusses an alternative binary linear model for computing the frustration index.

3.2 | The XOR model

The XOR model is designed to directly count the frustrated edges using binary variables $f_{ij} \in \{0, 1\}, \forall (i, j) \in E$. As before, we use $x_i \in \{0, 1\}, \forall i \in V$ to denote the color of node i. This model is formulated by observing that the frustration state of a positive edge $(i, j) \in E^+$ is given by $f_{ij} = \text{XOR}_{(x_i,x_j)}$. Similarly for $(i, j) \in E^-$, we have $f_{ij} = 1 - \text{XOR}_{(x_i,x_j)}$. Therefore, the minimum frustration count under all node colorings is obtained by solving (5).

$$\begin{aligned}
\min_{x_i : i\in V, f_{ij} : (i,j)\in E} Z = & \sum_{(i,j)\in E} f_{ij} \\
\text{s.t.} \quad & f_{ij} \ge x_i - x_j \quad \forall (i,j) \in E^+ \\
& f_{ij} \ge x_j - x_i \quad \forall (i,j) \in E^+ \\
& f_{ij} \ge x_i + x_j - 1 \quad \forall (i,j) \in E^- \\
& f_{ij} \ge 1 - x_i - x_j \quad \forall (i,j) \in E^- \\
& x_i \in \{0,1\} \quad \forall i \in V \\
& f_{ij} \in \{0,1\} \quad \forall (i,j) \in E. \qquad (5)
\end{aligned}$$

The dependencies between the f_{ij} and x_i, x_j values are taken into account using two standard XOR constraints per edge. Therefore, the XOR model has $n + m$ variables and $2m$ constraints. Note that f_{ij} variables are dependent variables because of the constraints and the minimization objective function. Therefore, we may specify f_{ij} variables as continuous variables in the unit interval, $f_{ij} \in [0, 1]$. A third linear formulation of the problem is provided in the next subsection.

3.3 | The ABS model

In this subsection, we propose the ABS model, a binary linear model in which we use two edge variables to represent the frustration state of an edge. We start by observing that for a given node coloring, $|x_i - x_j| = 1$ for a positive frustrated edge and $|x_i - x_j| = 0$ for a positive satisfied edge $(i, j) \in E^+$. Similarly, $1 - |x_i - x_j| = |x_i + x_j - 1|$ gives the frustration state of a negative edge $(i, j) \in E^-$.

To linearize the absolute value terms, we introduce additional binary variables $e_{ij}, h_{ij} \in \{0,1\}, \forall (i,j) \in E$. We replace $|x_i - x_j|$ with $e_{ij} + h_{ij}$ to represent the frustration state of a positive edge $(i,j) \in E^+$. This requires adding the constraint $x_i - x_j = e_{ij} - h_{ij}$ $\forall (i,j) \in E^+$. Similarly, we replace $|x_i + x_j - 1|$ with $e_{ij} + h_{ij}$ to represent the frustration state of a negative edge $(i,j) \in E^-$. Accordingly, we add the constraint $x_i + x_j - 1 = e_{ij} - h_{ij}$ $\forall (i,j) \in E^-$.

These two replacements allow us to linearize the two absolute value terms and formulate the ABS model in (6) which has $n + 2m$ variables and m constraints. Note that in an optimal solution, variables e_{ij} and h_{ij} both take the value 0 for a satisfied edge $(i,j) \in E$, whereas for a frustrated edge $(i,j) \in E$ exactly one of the two variables e_{ij} and h_{ij} takes the value 1. The objective function in (6) sums the frustration states of all edges and its optimal value equals the frustration index.

$$\begin{aligned} \min_{x_i : i \in V, e_{ij}, h_{ij} : (i,j) \in E} Z &= \sum_{(i,j) \in E} e_{ij} + h_{ij} \\ \text{s.t.} \quad x_i - x_j &= e_{ij} - h_{ij} \quad \forall (i,j) \in E^+ \\ x_i + x_j - 1 &= e_{ij} - h_{ij} \quad \forall (i,j) \in E^- \\ x_i &\in \{0,1\} \quad \forall i \in V \\ e_{ij} &\in \{0,1\} \quad \forall (i,j) \in E \\ h_{ij} &\in \{0,1\} \quad \forall (i,j) \in E. \end{aligned} \tag{6}$$

3.4 | Comparison of the models

In this subsection we compare the three models introduced above and two of the models suggested in [5], based on the number and type of constraints. Table 1 summarizes the comparison.

In optimal solutions of our three suggested models, the frustration state of edge (i,j) is represented by the corresponding term for edge (i,j) in the objective function. This leads to Equation (7) which makes a connection between the optimal values of the decision variables in the three models.

$$f_{ij} = e_{ij} + h_{ij} = (1 - a_{ij})/2 + a_{ij}(x_i + x_j - 2x_{ij}). \tag{7}$$

Note that not only does the number of constraints scale linearly with graph size, each constraint involves at most four variables. Thus the worst-case space usage for solving these models is $\mathcal{O}(n^2)$. The three linear models may perform differently in terms of resolution time and the number of branch and bound (B&B) nodes required to solve a given instance.

Solving large-scale binary programming models is not easy in general [11] and therefore there is a limit to the size of the largest graph whose frustration index can be computed in a given time. In the next section, we discuss some techniques for improving the performance of Gurobi in solving our suggested binary linear models.

4 | SPEED-UP TECHNIQUES

In this section we discuss techniques to speed up the branch and bound algorithm for solving the binary linear models described in the previous section. The branch and bound algorithm can be provided with a list of prioritized variables for branching which may speed up the solver if branching on these variables leads more quickly to integer solutions.

Another technique often deployed in solving binary and integer programming models is using *valid inequalities* which we discuss briefly. Two key features of valid inequalities are that they are satisfied by the optimal integer solutions (validity), but are violated by undesired feasible solutions (usefulness). We implement some valid inequalities as *lazy* constraints which are given to the solver, but only added to the model if they are violated by a solution [34, 42]. Implementing valid inequalities as lazy constraints restricts the model by removing the undesired solutions that violate them. Such valid and useful restrictions reduce resolution time [42].

TABLE 1 Comparison of optimization models developed for computing the frustration index

	Aref et al. UBQP [5]	Aref et al. binary linear [5]	AND (4)	XOR (5)	ABS (6)
Variables	n	$n+m$	$n+m$	$n+m$	$n+2\,m$
Constraints	0	$m^+ + m^-$	$2\,m^+ + m^-$	$2\,m^+ + 2\,m^-$	$m^+ + m^-$
Constraint type	-	Linear	Linear	Linear	Linear
Objective	quadratic	Linear	Linear	Linear	Linear

4.1 | Preprocessing data reduction

Standard graph preprocessing can be used to reduce graph size and order without changing the frustration index. This may reduce resolution time in graphs containing nodes of degree 0 and 1 (also called isolated and pendant vertices respectively) and nodes whose removal increases the number of connected components (also called articulation points) [38]. We have tested iterative reduction of isolated and pendant vertices as well as decomposing graphs by cutting them into smaller subgraphs using articulation points. Our experiments show that reducing isolated and pendant vertices does not considerably affect the resolution time. Moreover, the scarcity of articulation points in many graphs in which isolated and pendant vertices have been removed, makes decomposition based on articulation points not particularly useful.

4.2 | Branching priority and fixing a color

We relax the integrality constraints and observe in the Linear Programming (LP) relaxation of all three models that there always exists a fractional solution of $x_i = 0.5, \forall\, i \in V$ which gives an optimal objective function value of 0. We can increase the root node objective by fixing one node variable to value 1. Fixing a node variable also breaks the symmetry that exists and allows changing all node colors to give an equivalent solution. This is similar to fixing the *ghost spin* in the ground state calculation of a spin glass model [18] and is also used in [5].

When the color of node k is fixed by imposing $x_k = 1$, the variables associated with edges incident to node k take value 0.5 (in the ABS model one of the two variables e_{ij} and h_{ij} takes value 0.5). In all three models, this changes the fractional solution of the LP relaxation from 0 to $d(k)/2$ because all edges incident to node k contribute 0.5 to the objective function. This observation shows that the best node variable to be fixed is the one associated with the highest degree which allows for an increase of $\max_{i \in V} d(i)/2$ in the LP relaxation optimal objective function value. We formulate this as a constraint in (8).

$$x_k = 1, \quad k = \arg\max_{i \in V} d(i). \tag{8}$$

In our experiments, we always observed an improvement in the root node objective value when Equation (8) was added, which shows it is useful. We provide more detailed results on the root node objective values for several instances in Section 6.

Based on the same idea, we may modify the branch and bound algorithm so that it branches first on the node with the highest degree. This modification is implemented by specifying a branching priority for the node variables in which variable x_i has a priority given by its degree $d(i)$.

4.3 | Unbalanced triangle constraints

We consider one valid inequality for each negative cycle of length 3 (unbalanced triangle) in the graph. Under arbitrary coloring X, every negative cycle of the graph contains an odd number of frustrated edges. This means that any coloring of the nodes in an unbalanced triangle must produce at least one frustrated edge. Recalling that under coloring X, the variable f_{ij} is 1 if edge (i, j) is frustrated (and 0 otherwise), then for any node triple (i, j, k) defining an unbalanced triangle in G, inequality (9) is valid.

$$f_{ij} + f_{ik} + f_{jk} \geq 1 \quad \forall (i, j, k) \in T^-. \tag{9}$$

In (9), $T^- = \{(i, j, k) \in V^3 \mid a_{ij}a_{ik}a_{jk} = -1\}$ denotes the set of node triples that define an unbalanced triangle. The expression in inequality (9) denotes the sum of frustration states for the three edges (i, j), (i, k), (j, k) making an unbalanced triangle. Note that in order to implement the unbalanced triangle constraints (9), f_{ij} must be represented using the decision variables in the particular model. Equation (7) shows how f_{ij} can be defined in the AND and ABS models. We implement the valid inequality in (9) using Gurobi's feature for adding lazy constraints and ensure that lazy constraints that cut off the relaxation solution at the root node are also pulled into the model (see *lazy* as a tunable parameter in linear constraint attributes in [34]).

4.4 | Overall improvement made by the speed-up techniques

In this subsection, we report the resolution time improvement obtained by implementing the speed-up techniques. The evaluation is based on 100 Erdős-Rényi (ER) graphs with uniformly random parameters from the ranges $40 \leq n \leq 50$, $0 \leq \rho \leq 1$, and $0 \leq m^-/m \leq 1$. The total resolution time reduction observed when both speed-up techniques (Subsections 4.2 and 4.3) are implemented is 67% for the AND model, 90% for the XOR model, and 78% for the ABS model. Table 2 shows the resolution time improvements made by implementing the speed-up techniques individually and collectively.

TABLE 2 Usefulness of the speed-up techniques based on 100 Erdős-Rényi graphs

	Average resolution time (s)			Time improvement (%)		
	AND	XOR	ABS	AND	XOR	ABS
Without speed-up	14.80	41.60	19.71	-	-	-
With branching priority	5.90	4.91	5.50	60%	88%	72%
With triangle inequalities	9.21	31.72	17.26	38%	24%	12%
With both speed-up techniques	4.93	4.08	4.42	67%	90%	78%

5 | COMPUTATIONAL PERFORMANCE

In this section, our optimization models are tested on various random instances using 64-bit Gurobi version 7.5.2 on a desktop computer with an Intel Core i5 7600 @ 3.50 GHz (released in 2017) and 8.00 GB of RAM running 64-bit Microsoft Windows 10. We use *NetworkX* package in Python for generating random graphs. The models were created using Gurobi's Python environment in 64-bit Anaconda3 5.0.1 Jupyter.

5.1 | Comparison of the models' performance

In this subsection, we discuss the time performance of Gurobi for solving the extended binary linear models which include both speed-up techniques (Subsections 4.2 and 4.3).

In order to compare the performance of the three binary linear models, we consider 12 test cases each containing 10 Barabási-Albert (BA) random graphs with various combinations of density and proportion of negative edges. The results in Table 3 show that the three models have relatively similar performance in terms of resolution time.

Comparing values of the same column, it can be seen that graphs with a higher density (more edge variables) have a longer resolution time. For graphs of a given order and density, we observe the shortest resolution times for $m^-/m \in \{0.3, 1\}$ in most cases which are also associated with the two smallest averages of values of Z^*.

5.2 | Convergence of the models with and without the speed-up techniques

We investigate the algorithm convergence by running the three models with and without the speed-up techniques for one ER random graph and one BA random graph with $n = 100$, $m = 900$, $m^- = 600$ and plotting the upper and lower bounds over time. Figure 2 shows normalized bounds over time on a log scale where the vertical axes represent upper and lower bounds normalized by dividing by the optimal objective function value.

TABLE 3 Resolution time comparison of the three models based on test cases of 10 Barabási-Albert graphs

					Resolution time (s) mean ± SD		
n	m	ρ	$\frac{m^-}{m}$	Average Z^*	AND (4)	XOR (5)	ABS (6)
60	539	0.3	0.3	157.4	1.13 ± 0.48	1.59 ± 0.3	0.84 ± 0.1
			0.5	185.0	1.48 ± 0.56	2.95 ± 0.28	1.1 ± 0.19
			0.7	172.9	1.07 ± 0.41	2.55 ± 0.8	0.84 ± 0.16
			1	55.0	0.04 ± 0.01	0.04 ± 0.01	0.06 ± 0.02
	884	0.5	0.3	262.4	1.4 ± 0.16	0.45 ± 0.08	0.41 ± 0.04
			0.5	325.8	37.41 ± 11.53	27.09 ± 27.09	25.15 ± 8.46
			0.7	329.4	36.73 ± 8.28	39.8 ± 7.82	30.44 ± 5.73
			1	272.4	1 ± 0.17	0.77 ± 0.26	6.12 ± 4.61
70	741	0.3	0.3	217.0	4.07 ± 1.67	4.55 ± 0.77	1.52 ± 0.34
			0.5	260.6	4.56 ± 0.89	12.28 ± 1.72	2.84 ± 0.46
			0.7	248.0	2.94 ± 0.37	9.72 ± 2.32	1.87 ± 0.26
			1	78.0	0.07 ± 0	0.05 ± 0.01	0.1 ± 0.03
	1209	0.5	0.3	361.7	3.27 ± 0.34	0.76 ± 0.09	0.96 ± 0.1
			0.5	460.4	471.18 ± 77.27	322.99 ± 112.29	324.72 ± 131.86
			0.7	457.7	308.05 ± 130.31	369.14 ± 208.88	251.21 ± 96.75
			1	382.2	4.07 ± 1.08	2.93 ± 1.31	20.67 ± 14.28

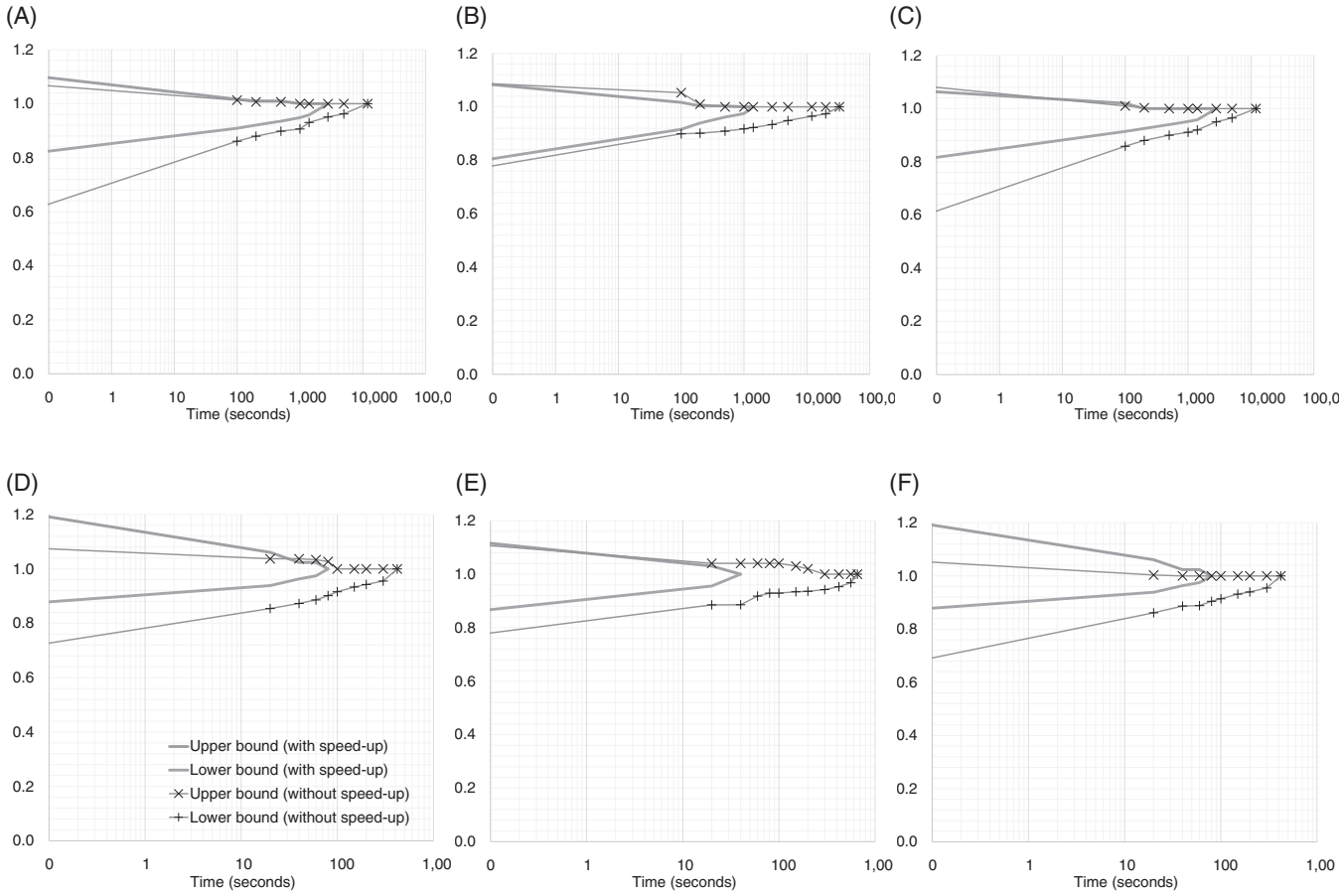

FIGURE 2 Normalized upper and lower bounds over time with and without the speed-up techniques for one ER graph and one BA graph with $n = 100$, $m = 900$, $m^- = 600$ on a log scale. Vertical axes show normalized upper and lower bounds. (A) The AND model, ER graph. (B) The XOR model, ER graph. (C) The ABS model, ER graph. (D) The AND model, BA graph. (E) The XOR model, BA graph. (F) The ABS model, BA graph [Colour figure can be viewed at wileyonlinelibrary.com]

For the randomly generated ER graph in Subfigures 2A-C, the resolution times of all three models without the speed-up techniques are over 12 000 seconds (and in one case 33 000 seconds). These resolution times are reduced to less than 2800 seconds (and in one case 1400 seconds) when the speed-up techniques are implemented.

Subfigures 2D-F show a considerable resolution time improvement for the randomly generated BA graph. It takes 420 seconds (80 seconds) for the AND model and the ABS model to find an optimal solution without (with) the speed-up techniques. The XOR model without (with) the speed-up techniques reaches optimality in 655 seconds (40 seconds).

5.3 | Largest instances solvable in 10 hours

Our experiments allow us to discuss the size of the largest graph whose frustration index can be computed in a reasonable time using one of our proposed extended binary linear model. Two important factors must be taken into consideration in this regard: network properties and processing capacities. As it is expected from our degree-based prioritized branching in Subsection 4.2, network properties such as degree heterogeneity could have an impact on the resolution time. Moreover, the numerical results in [5] suggest that reaching optimality in real signed networks takes a considerably shorter time compared to randomly generated signed networks of comparable size and order, confirming the observations of [16, 38]. Processing capacities of the computer that runs the optimization models are also relevant to the size of the largest solvable instance because Gurobi allows using multiple processing cores for exploring the feasible space in parallel [34]. Besides, exploring a large binary tree may require a considerable amount of memory which might be a determining factor in resolution time of some instances due to memory limits.

Given a maximum resolution time of 10 hours on the current hardware configuration (Intel Core i5 7600 @ 3.50 GHz and 8.00 GB of RAM), random instances with up to 2000 edges were observed to be solvable to global optimality. Regarding real signed graphs which have regularities favoring Gurobi's solver performance, graphs with up to 30 000 edges are solvable (to global optimality) within 10 hours. If we use more advanced processing capacities (32 Intel Xeon CPU E5-2698 v3 @ 2.30 GHz processors and 32 GB of RAM), real signed graphs with up to 100 000 edges are solvable (to global optimality) within 10 hours [8].

We have observed in most of our numerical experiments that the branch and bound algorithm finds the globally optimal solution in a fraction of the total resolution time, but it takes more time and computations to guarantee the optimality. To give an example, Subfigures 2A-C show that a considerable proportion of the resolution time, ranging in 30%-90%, is used for guaranteeing optimality after finding the globally optimal solution. One may consider using a nonzero mixed integer programming gap to find solutions within a guaranteed proximity of an optimal solution even if the instance has more than 100 000 edges.

6 | EVALUATING PERFORMANCE AGAINST THE LITERATURE

In this section, we use both random and real networks to evaluate not only the resolution time, but also the output of our models against other methods in the literature.

6.1 | Resolution time in random graphs

In this subsection, we compare the resolution time of our algorithm against other algorithms suggested for computing the frustration index. Besides [5], our review of the literature finds only two methods claiming exact computation of the frustration index [12, 38]. Brusco and Steinley suggested a branch and bound algorithm for minimizing the overall frustration (under a different name) for a predefined number of colors [12]. Hüffner et al. have suggested a data-reduction schemes and an iterative compression algorithm for computing the frustration index [38].

Brusco and Steinley have reported running times for very small graphs with only up to $n = 21$ vertices. While, their exact algorithm fails to solve graphs as large as $n = 30$ in a reasonable time [12], our binary linear models solve such instances in split seconds. Hüffner et al. have generated random graphs of order n with low densities ($\rho \leq 0.04$) to test their algorithm [38]. The largest of such random graphs solvable by their algorithm in 20 hours has 500 nodes. They also reported that only three out of five random graphs with $n \in \{100, 200, 300, 400, 500\}$ can be solved by their method in 20 hours. Our three binary linear models solve all such instances in less than 100 seconds.

6.2 | Resolution time and algorithm output in real networks

In this section we use signed network data sets from biology and international relations. The frustration index of biological networks has been a subject of interest to measure the distance to *monotonicity* [16, 39]. In international relations, the frustration index is used to measure distance to balance for a network of countries [20]. In this section, the frustration index is computed in real biological and international relations networks by solving the three binary linear models coupled with the two speed-up techniques (Subsections 4.2 and 4.3).

We use *effective branching factor* as a performance measure. If the solver explores b branch and bound nodes to find an optimal solution of a model with v variables, the effective branching factor is $\sqrt[v]{b}$. The most effective branching is obtained when the solver only explores one branch and bound node to reach optimality. The effective branching factor for such a case would take value 1 which represents the strength of the mathematical formulation.

6.2.1 | Biological data sets

We use the four signed biological networks that were previously analyzed by [16, 39]. The epidermal growth factor receptor (EGFR) pathway [57] is a signed network with 779 edges. The molecular interaction map of a macrophage (macro.) [56] is another well-studied signed network containing 1425 edges. We also investigate two gene regulatory networks, related to two organisms: a eukaryote, the *yeast Saccharomyces cerevisiae* (yeast) [15], and a bacterium, *Escherichia coli* (*E. coli*) [59]. The yeast and *E. coli* networks have 1080 and 3215 edges respectively. The data sets for real networks used in this study are publicly available in a Figshare research data repository [4]. For more details on the four biological data sets, one may refer to [39].

We use root node objective, number of B&B nodes, effective branching factor, and resolution time as performance measures. The performance of three binary linear models can be compared based on these measures in Table 4 in which values in brackets show the corresponding measure for the case in which speed-up techniques were not used.

DasGupta et al. have suggested approximation algorithms [16] that are later tested on the four biological networks in [38]. Their approximation method provides $196 \leq L(G)_{\text{EGFR}} \leq 219$ which our exact model proves to be incorrect. The bounds obtained by implementing their approximation are not incorrect for the other three networks, but they have very large gaps between lower and upper bounds.

Hüffner et al. have previously investigated frustration in the four biological networks suggesting a data-reduction schemes and (an attempt at) an exact algorithm [38]. Their suggested data-reduction schemes can take more than 5 hours for yeast, more

TABLE 4 Performance measures for the three binary linear models with (and without) the speed-up techniques

	Graph	EGFR	Macro.	Yeast	*E. coli*
	n, m	329, 779	678, 1425	690, 1080	1461, 3215
Root node objective	AND	28.5	67	11.5	130.5
		(13)	(53)	(0)	(4)
	XOR	28.5	67	11.5	130.5
		(13)	(53)	(0)	(4)
	ABS	28.5	67	11.5	130.5
		(13)	(53)	(0)	(4)
Number of B&B nodes	AND	3	1	1	31
		(91)	(199)	(7)	(279)
	XOR	1	1	1	3
		(25)	(1)	(1)	(19)
	ABS	1	1	3	36
		(47)	(456)	(7)	(357)
Effective branching factor	AND	1.0010	1	1	1.0007
		(1.0041)	(1.0025)	(1.0011)	(1.0012)
	XOR	1	1	1	1.0002
		(1.0029)	(1)	(1)	(1.0006)
	ABS	1	1	1.0004	1.0006
		(1.0027)	(1.0022)	(1.0008)	(1.0010)

than 15 hours for EGFR, and more than 1 day for macrophage if the parameters are not perfectly tuned. Besides the resolution time issue, their algorithm provides $L(G)_{\text{EGFR}} = 210$, $L(G)_{\text{macrophage}} = 374$, both of which are proven to be incorrect by our results. They report that their algorithm fails to terminate for *E. coli* [38].

Iacono et al. have also investigated frustration in these four networks [39]. Their heuristic algorithm provides upper and lower bounds for EGFR, macrophage, yeast, and *E. coli* with 96.37%, 90.96%, 100%, and 98.38% ratio of lower to upper bound respectively. The comparison of our outputs against those reported in the literature is provided in Table 5.

Iacono et al. also suggest an upper bound for the frustration index ([39], page 227). However, some values of the frustration index in complete graphs with all negative edges show that their suggested upper bound is incorrect (take a complete graph with 9 nodes and 36 negative edges which has a frustration index of 16 while the bound suggested in [39] gives a value of 15). For a more detailed discussion on bounds for the frustration index, one may refer to [5, 53].

We also compare our resolution times to the best results reported for heuristics and approximation algorithms in the literature. Hüffner et al. have provided resolution time results for their suggested algorithm [38] (if parameters are perfectly tuned for each instance) as well as the algorithm suggested by DasGupta et al. [16]. Iacono et al. have only mentioned that their heuristic requires a fairly limited amount of time (a few minutes on an ordinary PC [39]) that we conservatively interpret as 60 seconds.

Table 6 sums up the resolution time comparison of our suggested models against the literature in which the values for running our models without the speed-up techniques are provided inside brackets. As the hardware configuration is not reported in [16, 39], we conservatively evaluate the order-of-magnitude improvements in resolution time with respect to the differences in computing power in different years.

According to Moore's law [54], the exponential increase in transistor density on integrated circuits leads to computer power doubling almost every 2 years. Moore's prediction has been remarkably accurate from 1965 to 2013, while the actual rate of increase in computer power has slowed down since 2013 [51].

TABLE 5 Our algorithm output against the best results reported in the literature

Author	DasGupta et al.	Hüffner et al.	Iacono et al.	Aref et al.	AND	XOR	ABS
Reference	[16]	[38]	[39]	[5]	(4)	(5)	(6)
EGFR	[196, 219][a]	210[a]	[186, 193]	193	193	193	193
Macro.	[218383]	374[a]	[302, 332]	332	332	332	332
Yeast	[0, 43]	41	41	41	41	41	41
E. coli	[0, 385]	[b]	[365, 371]	371	371	371	371

[a]Incorrect results.
[b]The algorithm does not converge.

TABLE 6 Algorithm resolution time in seconds with (and without) the speed-up techniques against the results reported in the literature

Year	2010	2010	2010	2018	2018	2018	2018
Reference	[16]	[38]	[39]	[5]	AND (4)	XOR (5)	ABS (6)
EGFR	420	6480	>60	0.68	0.27 (0.82)	0.21 (0.67)	0.23 (0.66)
Macro.	2640	60	>60	1.85	0.34 (1.24)	0.26 (1.37)	0.49 (1.30)
Yeast	4620	60	>60	0.33	0.18 (0.45)	0.11 (0.28)	0.15 (0.39)
E. coli	[a]	[b]	>60	18.14	0.99 (1.91)	1.97 (4.73)	0.74 (1.86)

[a]Not reported.
[b]The algorithm does not converge.

Moore's law ballpark figures allow us to compare computations executed on different hardware in different years. We conservatively estimate a factor of 16 times for the improvements in computer power between 2010 and 2018 to be attributable to hardware improvements. The resolution times of the slowest (fastest) model among AND, XOR, and ABS in Table 6 show a factor of improvement ranging between 30 and 333 (81-545) compared to the fastest resolution time in 2010 [16, 38, 39]. This shows our resolution time improvements are not merely resulted from hardware differences.

While data-reduction schemes [38] can take up to 1 day for these data sets and heuristic algorithms [39] only provide bounds with up to 9% gap from optimality, our three binary linear models equipped with the speed-up techniques (Subsections 4.2 and 4.3) solve the four instances to optimality in a few seconds.

6.2.2 | International relations data sets

We also compute the frustration index for two data sets of international relations networks. In international relation networks, countries and their relations are represented by nodes and edges of signed graphs. We use the Correlates of War (CoW) [58] data set which has 51 instances of networks with up to 1247 edges [20] and the United Nations General Assembly (UNGA) [52] data set which has 62 instances with up to 15 531 edges when converted into signed networks by [28]. Figueiredo and Frota provide detailed explanation on the process of creating signed networks from the UNGA data [28].

The CoW signed network data set is constructed by Doreian and Mrvar [20] based on signed international relations between countries in 1946-1999. In their analysis, some numerical results provided on the CoW data set are referred to as line index [20]. However, the values of $L(G)$ we have obtained using our optimization models prove that values reported in [20] for the 51 time frames of the network are never the smallest number of edges whose removal results in balance. Doreian and Mrvar have not reported any resolution time, but have suggested that determining their line index is in general a polynomial-time hard problem [20]. The resolution times of our models for each instance of the CoW data set is $\leq$0.1 seconds.

We also tested our three models on the UNGA instances. The UNGA data set is based on voting on the UN resolutions. In this data set, instances refer to annual UNGA sessions between 1946 and 2008. Figure 3 shows the resolution times of our models for instances of this data set.

As it can be seen in Figure 3, most UNGA instances can be solved in less that 5 seconds using any of the three models. The XOR and the ABS models solve all UNGA instances in less than a minute, while solving the AND model for instance 21 and

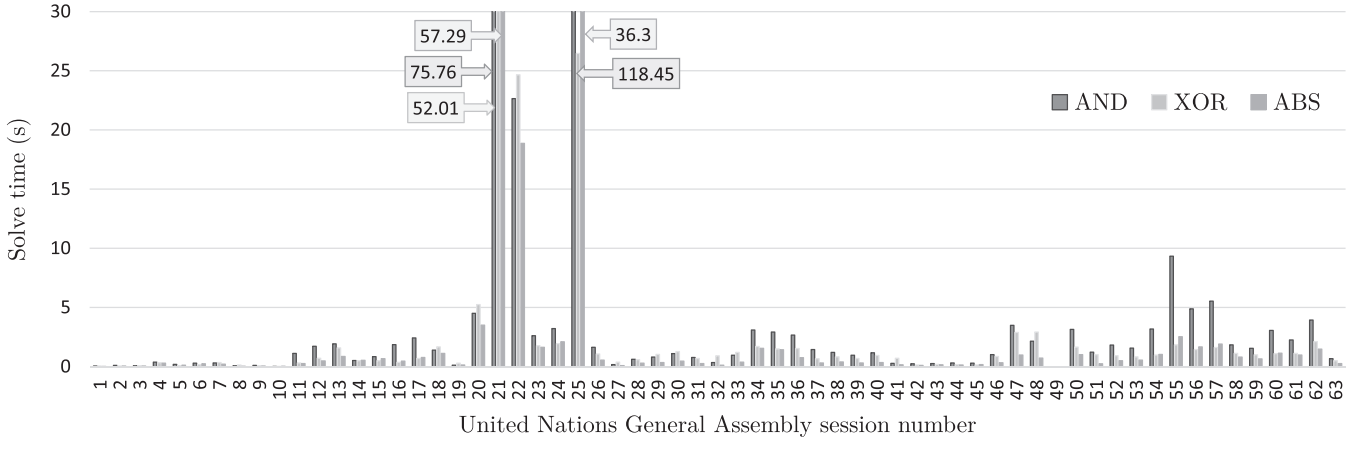

FIGURE 3 Resolution times of AND, XOR, and ABS models tested on the UNGA instances [Colour figure can be viewed at wileyonlinelibrary.com]

instance 25 takes about 75 and 118 seconds respectively. These two harder instances have the highest values of the frustration index ($L(G) = 616$ and $L(G) = 611$ respectively) in the UNGA data set.

7 | OTHER FORMULATIONS AND EXTENSIONS

In this section we provide an alternative formulation and two extensions to the 2-color minimum frustration count optimization problem.

7.1 | Max (2, 2)-CSP formulation and theoretical results

In this subsection, we formulate the problem of computing the frustration index as a constraint satisfaction problem in (10) and provide theoretical results on the fastest known algorithms. Computation of the frustration index can be formulated as a maximum 2-constraint satisfaction problem with 2 states per variable (Max (2, 2)-CSP) with n variables and m constraints.

The signed graph, $G(V, E, \sigma)$, is the input constraint graph. We consider a score for each edge (i, j) depending on its sign σ_{ij} and the assignment of binary values to its endpoints. In the formulation provided in (10), the dyadic score function $S_{(i,j)} : \{0, 1\}^2 \rightarrow \{0, 1\}$ determines the satisfaction of edge (i, j) accordingly (score 1 for satisfied and score 0 for frustrated). The output of solving this problem is the coloring function ϕ: $V \rightarrow \{0, 1\}$ which maximizes the total number of satisfied edges as score function $S(\phi)$.

Denoting the maximum score function value by $S^*(\phi)$, the frustration index can be calculated as the number of edges that are not satisfied $m - S^*(\phi)$.

$$\begin{aligned} \max_{\phi} S(\phi) &= \sum_{(i,j)\in E} S_{(\phi(i),\phi(j))} \\ S_{(i,j)} &= \{((0,0),(1+\sigma_{ij})/2), \\ &\quad ((0,1),(1-\sigma_{ij})/2), \\ &\quad ((1,0),(1-\sigma_{ij})/2), \\ &\quad ((1,1),(1+\sigma_{ij})/2)\}. \end{aligned} \tag{10}$$

According to worst-case analyses, the fastest known algorithm [43] with respect to n solves Max (2,2)-CSP in $\mathcal{O}(nm2^{n\omega/3})$, where ω is the matrix multiplication exponent. Since $\omega < 2.373$ [45], the running time of the algorithm from [43] is $\mathcal{O}(1.7303^n)$. It improves on the previous fastest algorithm [62] only in the polynomial factor of the running time. With respect to n, the algorithm in [43] is the fastest known algorithm for MAXCUT, and therefore for computing the frustration index as well. Both algorithms [43, 62] use exponential space and it is open whether MAXCUT can be solved in $\mathcal{O}(c^n)$ for some $c < 2$ when only polynomial space is allowed.

With respect to the number m of edges, the Max (2,2)-CSP formulation in (10) enables the use of algorithms from [31, 60]. The first algorithm uses $2^{(9m/50+\mathcal{O}(m))}$ time and polynomial space [31], while the second algorithm uses $2^{(13m/75+\mathcal{O}(m))}$ time and exponential space. With respect to the number of edges, these two algorithms [31, 60] are also the fastest algorithms known for MAXCUT, and therefore for computing the frustration index.

7.2 | Weighted minimum frustration count optimization problem

We extend the 2-color minimum frustration count optimization problem for a graph with weights $w_{ij} \in [-1, 1]$ instead of the signs $a_{ij} \in \{-1, 1\}$ on the edges. We call such a graph a *weighted signed graph*.

Taking insights from (7), the frustration of edge $(i, j) \in E$ with weight w_{ij} can be represented by $f_{ij} = (1 - w_{ij})/2 + w_{ij}(x_i + x_j - 2x_{ij})$ using the binary variables x_i, x_j, x_{ij} of the AND model (4). Note that, the frustration of an edge in a weighted signed graph is a continuous variable in the unit interval $f_{ij} \in [0, 1]$.

Note that, $a_{ij}x_{ij} \leq (3a_{ij} - 1)(x_i + x_j)/4 + (1 - a_{ij})/2$ embodies all constraints for edge (i, j) in the AND model regardless of the edge sign. Accordingly, the constraints of the AND model can be modified to incorporate weights w_{ij}. The weighted minimum frustration count optimization problem can be formulated as a binary linear programming model in (11).

$$\begin{aligned} \min_{x_i : i\in V, x_{ij} : (i,j)\in E} Z &= \sum_{(i,j)\in E} (1 - w_{ij})/2 + w_{ij}(x_i + x_j - 2x_{ij}) \\ \text{s.t.} \quad w_{ij}x_{ij} &\leq (3w_{ij} - 1)(x_i + x_j)/4 + (1 - w_{ij})/2 \quad \forall (i,j) \in E \\ x_i &\in \{0, 1\} \quad \forall i \in V \\ x_{ij} &\in \{0, 1\} \quad \forall (i,j) \in E. \end{aligned} \tag{11}$$

We have generated random weighted signed graphs to test the model in (11). Our preliminary results show that the weighted version of the problem (11) is solved faster than the original models for signed graphs.

7.3 | Multicolor minimum frustration count optimization problem

We formulate another extension to the 2-color minimum frustration count optimization problem by allowing more than two colors to be used. As previously mentioned in Subsection 1.6, a signed network is k-balanced if and only if its vertex set can be partitioned into k subsets (for some fixed $k \geq 1$) such that each negative edge joins vertices belonging to different subsets [17]. Figure 4 demonstrates an example graph and the frustrated edges for various numbers of colors. Subfigure 4D shows that the graph is weakly balanced.

The harder problem of finding the minimum number of frustrated edges where k is not specified in advance (an arbitrary number of node colors) is referred to as the correlation clustering problem. As mentioned in Subsection 1.6, another integer programming formulation for the correlation clustering problem is suggested by [19] which is widely used in the literature [22, 29, 47].

In the multicolor minimum frustration count optimization problem, each node may be given one of a set of colors $C = \{1, 2, 3, \ldots, k := |C|\}$. Assume $c_i \in C$ is the color of node i. We consider that a positive edge $(i, j) \in E^+$ is frustrated (indicated by $f_{ij} = 1$) if its endpoints i and j are colored differently, that is, $c_i \neq c_j$; otherwise it is not frustrated (indicated by $f_{ij} = 0$). A negative edge $(i, j) \in E^-$ is frustrated (indicated by $f_{ij} = 1$) if $c_i = c_j$; otherwise it is not frustrated (indicated by $f_{ij} = 0$).

Using binary variables $x_{ic} = 1$ if node $i \in V$ has color $c \in C$ (and $x_{ic} = 0$ otherwise), we formulate this as the following binary linear model in Equation (12).

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in E} f_{ij} \\
\text{s.t.} \quad & \sum_{c\in C} x_{ic} = 1 \quad \forall i \in V \\
& f_{ij} \geq x_{ic} - x_{jc} \quad \forall (i,j) \in E^+, \forall c \in C \\
& f_{ij} \geq x_{ic} + x_{jc} - 1 \quad \forall (i,j) \in E^-, \forall c \in C \\
& x_{ic} \in \{0,1\} \quad \forall i \in V, \forall c \in C \\
& f_{ij} \in \{0,1\} \quad \forall (i,j) \in E.
\end{aligned} \tag{12}
$$

If we have just two colors, then we use $x_i \in \{0, 1\}$ to denote the color of node i. This gives the XOR model expressed in Equation (5).

Solving the problem in (12) provides us with the minimum number of frustrated edges in the k-color setting. This number determines how many edges should be removed to make the network k-balanced. For a more general formulation of partitioning graph vertices into k sets, one may refer to [3] where numerical results for graphs with up to 20 nodes are provided.

8 | CONCLUSION

In this study, we provided an efficient method for computing a standard measure in signed graphs which has many applications in different disciplines. The present study suggested efficient mathematical programming models and speed-up techniques for computing the frustration index in graphs with up to 15 000 edges on inexpensive hardware.

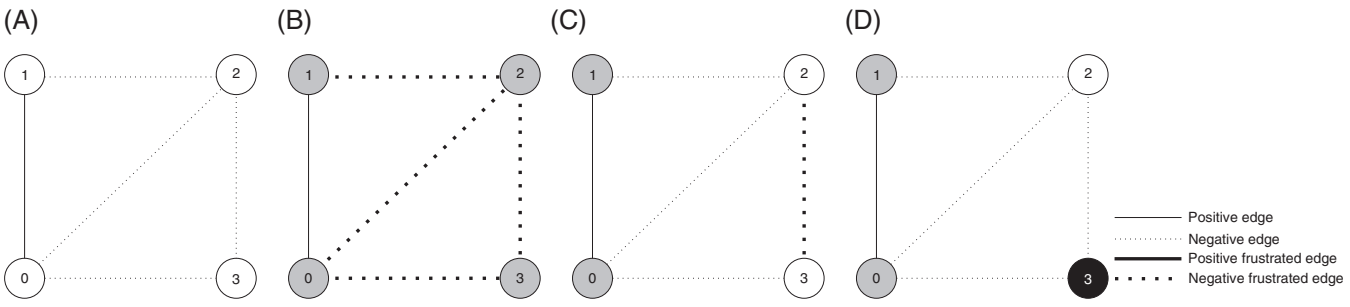

FIGURE 4 The frustrated edges represented by dashed lines for the multicolor minimum frustration count optimization problem. (A) An example graph with $n = 4$, $m^- = 4$, $m^+ = 1$. (B) One color resulting in four frustrated edges. (C) Two colors resulting in one frustrated edge. (D) Three colors resulting in no frustrated edge

We developed three new binary optimization models which outperform previous methods by large factors. We also suggested prioritized branching and valid inequalities which make the binary linear optimization models several times (Table 6) faster than recently developed models [5] and capable of processing relatively large instances.

Extensive numerical results on random and real networks were provided to evaluate computational performance and underline the superiority of our models to other methods in the literature in both resolution time and algorithm output. We also formulated the problem as a constraint satisfaction model and provided theoretical results on the fastest known algorithms for computing the frustration index with respect to the number of nodes and the number of edges. We also provided two extensions to the model for future investigation.

ACKNOWLEDGMENTS

The authors thank the anonymous referees, Serge Gaspers, Gregory Gutin, and Jeffrey Linderoth for valuable comments, Yuri Frota for providing data on United Nations General Assembly instances, and Serge Gaspers for his contribution to Subsection 7.1. Open Access funding provided by Projekt DEAL.

ORCID

Samin Aref https://orcid.org/0000-0002-5870-9253
Andrew J. Mason https://orcid.org/0000-0001-9848-6595
Mark C. Wilson https://orcid.org/0000-0002-3343-7458

REFERENCES

[1] R.P. Abelson and M.J. Rosenberg, *Symbolic psycho-logic: A model of attitudinal cognition*, Behav. Sci. **3** (1958), 1–13.
[2] A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev, *$\mathcal{O}(\sqrt{\log n})$ approximation algorithms for min UnCut, min 2CNF deletion, and directed cut problems*, Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing, STOC '5, ACM, New York, NY, 2005, pp. 573–581.
[3] Z. Ales, A. Knippel, and A. Pauchet, *Polyhedral combinatorics of the k-partitioning problem with representative variables*, Discrete Appl. Math. **211** (2016), 1–14.
[4] S. Aref, Signed networks from sociology and political science, systems biology, international relations, finance, and computational chemistry, Figshare research data repository, 2017, https://doi.org/10.6084/m9.figshare.5700832.v2.
[5] S. Aref, A.J. Mason, and M.C. Wilson, "*Computing the line index of balance using integer programming optimisation*," *Optimization Problems in Graph Theory*, B. Goldengorin (ed.), Springer, Cham, Switzerland, 2018, pp. 65–84.
[6] S. Aref and Z. Neal, *Legislative effectiveness hangs in the balance: Studying balance and polarization through partitioning signed networks*, arXiv:1906.01696, 2019, http://arxiv.org/pdf/1906.01696.
[7] S. Aref and M.C. Wilson, *Measuring partial balance in signed networks*, J. Complex Netw. **6** (2018), 566–595.
[8] S. Aref and M.C. Wilson, *Balance and frustration in signed networks*, J. Complex Netw. **7** (2019), 163–189.
[9] A. Avidor and M. Langberg, *The multi-multiway cut problem*, Theoret. Comput. Sci. **377** (2007), 35–42.
[10] N. Bansal, A. Blum, and S. Chawla, *Correlation clustering*, Mach. Learn. **56** (2004), 89–113.
[11] A. Bilitzky and A. Sadeh, *Efficient solutions for special zero-one programming problems*, J. Combin. Optim. **10** (2005), 227–238.
[12] M. Brusco and D. Steinley, *K-balance partitioning: An exact method with applications to generalized structural balance and other psychological contexts*, Psychol. Methods **15** (2010), 145–157.
[13] D. Cartwright and F. Harary, *Structural balance: A generalization of Heider's theory*, Psychol. Rev. **63** (1956), 277–293.
[14] T. Coleman, J. Saunderson, and A. Wirth, *A local-search 2-approximation for 2-correlation-clustering*, European Symposium on Algorithms, Berlin, Germany, 2008, pp. 308–319.
[15] M.C. Costanzo, M.E. Crawford, J.E. Hirschman, J.E. Kranz, P. Olsen, L.S. Robertson, M.S. Skrzypek, B.R. Braun, K.L. Hopkins, P. Kondu, C. Lengieza, J.E. Lew-Smith, M. Tillberg, and J.I. Garrels, *YPD™, PombePD™ and WormPD™: Model organism volumes of the BioKnowledge™ Library, an integrated resource for protein information*, Nucleic Acids Res. **29** (2001), 75–79.
[16] B. DasGupta, G.A. Enciso, E. Sontag, and Y. Zhang, *Algorithmic and complexity results for decompositions of biological networks into monotone subsystems*, Biosystems **90** (2007), 161–178.
[17] J.A. Davis, *Clustering and structural balance in graphs*, Hum. Relat. **20** (1967), 181–187.
[18] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi, *Exact ground states of Ising spin glasses: New experimental results with a branch-and-cut algorithm*, J. Stat. Phys. **80** (1995), 487–496.
[19] E.D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica, *Correlation clustering in general weighted graphs*, Theoret. Comput. Sci. **361** (2006), 172–187.
[20] P. Doreian and A. Mrvar, *Structural balance and signed international relations*, J. Soc. Struct. **16** (2015), 1–49.
[21] T. Došlić and D. Vukicevic, *Computing the bipartite edge frustration of fullerene graphs*, Discrete Appl. Math. **155** (2007), 1294–1301.
[22] L. Drummond, R. Figueiredo, Y. Frota, and M. Levorato, "*Efficient solution of the correlation clustering problem: An application to structural balance*," *On the Move to Meaningful Internet Systems: OTM 2013 Workshops*Springer, Berlin, Heidelberg, 2013, pp. 674–683.
[23] P. Esmailian, S.E. Abtahi, and M. Jalili, *Mesoscopic analysis of online social networks: The role of negative ties*, Phys. Rev. E **90** (2014), 042817.
[24] P. Esmailian and M. Jalili, *Community detection in signed networks: The role of negative ties in different scales*, Sci. Rep. **5** (2015), 14339.
[25] E. Estrada and M. Benzi, *Walk-based measure of balance in signed networks: Detecting lack of balance in social networks*, Phys. Rev. E **90** (2014), 1–10.
[26] G. Facchetti, G. Iacono, and C. Altafini, *Computing global structural balance in large-scale signed social networks*, Proc Natl. Acad. Sci. U. S. A. **108** (2011), 20953–20958.

[27] G. Facchetti, G. Iacono, and C. Altafini, *Exploring the low-energy landscape of large-scale signed social networks*, Phys. Rev. E **86** (2012), 036116.

[28] R. Figueiredo and Y. Frota, *The maximum balanced subgraph of a signed graph: Applications and solution approaches*, Eur. J. Oper. Res. **236** (2014), 473–487.

[29] R. Figueiredo and G. Moura, *Mixed integer programming formulations for clustering problems related to structural balance*, Soc. Netw. **35** (2013), 639–651.

[30] N.G. Fytas, P.E. Theodorakis, and A.K. Hartmann, *Revisiting the scaling of the specific heat of the three-dimensional random-field Ising model*, Eur. Phys. J. B **89** (2016), 200.

[31] S. Gaspers and G.B. Sorkin, "*Separate, measure and conquer: Faster polynomial-space algorithms for Max 2-CSP and counting dominating sets*," *Automata, Languages, and Programming*, Springer, Berlin, Heidelberg, 2015, pp. 567–579.

[32] I. Giotis and V. Guruswami, *Correlation clustering with a fixed number of clusters*, Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, Society for Industrial and Applied Mathematics, SODA '06, Philadelphia, PA, 2006, pp. 1167–1176.

[33] M. Grötschel and W.R. Pulleyblank, *Weakly bipartite graphs and the MAX-CUT problem*, Oper. Res. Lett. **1** (1981), 23–27.

[34] Gurobi Optimization Inc., 2017. Gurobi optimizer reference manual. http://www.gurobi.com/documentation/7.5/refman/index.html (Accessed May 1, 2017).

[35] F. Harary, *On the measurement of structural balance*, Behav. Sci. **4** (1959), 316–323.

[36] F. Harary, M.H. Lim, and D.C. Wunsch, *Signed graphs for portfolio analysis in risk management*, IMA J. Manage. Math. **13** (2002), 201–210.

[37] F. Heider, *Social perception and phenomenal causality*, Psychol. Rev. **51** (1944), 358–378.

[38] F. Hüffner, N. Betzler, and R. Niedermeier, *Separator-based data reduction for signed graph balancing*, J. Combin. Optim. **20** (2010), 335–360.

[39] G. Iacono, F. Ramezani, N. Soranzo, and C. Altafini, *Determining the distance to monotonicity of a biological network: A graph-theoretical approach*, IET Syst. Biol. **4** (2010), 223–235.

[40] P.W. Kasteleyn, *Dimer statistics and phase transitions*, J. Math. Phys. **4** (1963), 287–293.

[41] S. Khot, *On the power of unique 2-prover 1-round games*, Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing, STOC '02, Montreal, Quebec, Canada, 2002, pp. 767–775.

[42] E. Klotz and A.M. Newman, *Practical guidelines for solving difficult mixed integer linear programs*, Surv. Oper. Res. Manage. Sci. **18** (2013), 18–32.

[43] M. Koivisto, *Optimal 2-constraint satisfaction via sum-product algorithms*, Inform. Process. Lett. **98** (2006), 24–28.

[44] J. Kunegis, *Applications of structural balance in signed social networks*, arXiv:1402.6865, 2014, http://arxiv.org/pdf/1402.6865.

[45] F. Le Gall, *Powers of tensors and fast matrix multiplication*, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, ISSAC '14, ACM, New York, NY, 2014, pp. 296–303.

[46] J. Leskovec, D. Huttenlocher, and J. Kleinberg, *Signed networks in social media*, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, ACM, New York, NY, 2010, pp. 1361–1370.

[47] M. Levorato, L. Drummond, Y. Frota, and R. Figueiredo, *An ILS algorithm to evaluate structural balance in signed social networks*, Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC '15, New York, NY, 2015, pp. 1117–1122.

[48] M. Levorato, R. Figueiredo, Y. Frota, and L. Drummond, *Evaluating balancing on social networks through the efficient solution of correlation clustering problems*, EURO J. Comput. Optim. **5** (2017), 467–498.

[49] L. Ma, M. Gong, H. Du, B. Shen, and L. Jiao, *A memetic algorithm for computing and transforming structural balance in signed networks*, Knowl. Based Syst. **85** (2015), 196–209.

[50] L. Ma, M. Gong, J. Yan, F. Yuan, and H. Du, *A decomposition-based multi-objective optimization for simultaneous balance computation and transformation in signed networks*, Inform. Sci. **378** (2017), 144–160.

[51] C. Mack, *The multiple lives of Moore's law*, IEEE Spectrum **52** (2015), 31–37.

[52] K.T. Macon, P.J. Mucha, and M.A. Porter, *Community structure in the United Nations General Assembly*, Physica A **391** (2012), 343–361.

[53] F. Martin, *Frustration and isoperimetric inequalities for signed graphs*, Discrete Appl. Math. **217** (2017), 276–285.

[54] G.E. Moore, *Cramming more components onto integrated circuits*, Electronics **38** (1965), 114–117.

[55] R.Z. Norman and F.S. Roberts, *A derivation of a measure of relative balance for social structures and a characterization of extensive ratio systems*, J. Math. Psychol. **9** (1972), 66–91.

[56] K. Oda, T. Kimura, Y. Matsuoka, A. Funahashi, M. Muramatsu, and H. Kitano, *Molecular interaction map of a macrophage*, AfCS Res. Rep. **2** (2004), 1–12.

[57] K. Oda, Y. Matsuoka, A. Funahashi, and H. Kitano, *A comprehensive pathway map of epidermal growth factor receptor signaling*, Mol. Syst. Biol. **1** (2005), 1–17.

[58] J. Pevehouse, T. Nordstrom, and K. Warnke, *The correlates of war 2 international governmental organizations data version 2.0*, Conflict Manage. Peace Sci. **21** (2004), 101–119.

[59] H. Salgado, S. Gama-Castro, M. Peralta-Gil, E. Díaz-Peredo, F. Sánchez-Solano, A. Santos-Zavaleta, I. Martinez-Flores, V. Jiménez-Jacinto, C. Bonavides-Martínez, J. Segura-Salazar, A. Martínez-Antonio, and J. Collado-Vides, *Regulondb (version 5.0):* Escherichia coli *k-12 transcriptional regulatory network, operon organization, and growth conditions*, Nucleic Acids Res. **34** (2006), D394–D397.

[60] A.D. Scott and G.B. Sorkin, *Linear-programming design and analysis of fast algorithms for Max 2-CSP*, Discrete Optim. **4** (2007), 260–287.

[61] E. Terzi and M. Winkler, *A spectral algorithm for computing social balance*, Proceedings of International Workshop on Algorithms and Models for the Web-Graph, WAW 2011, Springer, Berlin, Heidelberg, 2011, pp. 1–13.

[62] R. Williams, *A new algorithm for optimal 2-constraint satisfaction and its implications*, Theoret. Comput. Sci. **348** (2005), 357–365.

[63] M. Yannakakis, *Edge-deletion problems*, SIAM J. Comput. **10** (1981), 297–309.

[64] T. Zaslavsky, *Balanced decompositions of a signed graph*, J Combin. Theory Ser. B **43** (1987), 1–13.

How to cite this article: Aref S, Mason AJ, Wilson MC. A modeling and computational study of the frustration index in signed networks. *Networks.* 2020;75:95–110. https://doi.org/10.1002/net.21907