On the Best Interval Approximation Problem

Peter Blohm[*,3], Florian Chen[*,1], Aristides Gionis[2], and Stefan Neumann[3]

[1] University of Oxford, Oxford, UK florian.chen@cs.ox.ac.uk
[2] KTH Royal Institute of Technology, Stockholm, Sweden argioni@kth.se
[3] TU Wien, Vienna, Austria {peter.blohm,stefan.neumann}@tuwien.ac.at

Abstract. Best Interval Approximation (BIA) is a recent generalisation of correlation clustering that represents the vertices of a signed graph by intervals. The task is to find a set of intervals where positive edges correspond to overlapping intervals and negative edges correspond to disjoint intervals. In restricted settings, polynomial-time approximation schemes (PTAS) and effective heuristic algorithms exist, yet several theoretical and algorithmic questions about BIA remain open. In this paper, we answer three such open questions. First, we generalise the existing PTAS for complete graphs from a fixed to an arbitrary number of intervals. Second, we disprove an existing conjecture, which states that every instance of BIA admits a solution satisfying at least three quarters of all edges. Finally, we introduce pruning rules that decompose instances into smaller subproblems without loss of optimality. In experiments on real-world signed networks, these pruning rules remove up to 84% of vertices and 60% of edges, and, by relaxing strict optimality, we obtain an efficient greedy heuristic for the general BIA problem.

Keywords: Best Interval Approximation · Signed graphs · Graph pruning · Interval graphs · Correlation clustering · Approximation algorithms

1 Introduction

Social networks play an important role in modern societies and enable billions of interactions online. Although many such interactions are positive, a considerable number are also negative. This is particularly common between individuals with conflicting viewpoints [25,26]. Hence, it has become important to understand such opposing opinions and the negative interactions they induce.

A natural modelling approach for this setting is via signed graphs, where each edge can represent either a positive or a negative interaction. A standard method for analysing signed graphs is Correlation Clustering, which partitions the graph into disjoint vertex sets that maximise the positive interactions within, and the negative interactions between clusters. In our recent work [4], we argue that this approach is too restrictive for the often complex and nuanced real-world interaction structures. To overcome this, we proposed Best Interval Approximation (BIA) as a generalisation of Correlation Clustering that

* Equal contribution, authors ordered alphabetically.

aims to obtain an interpretable description of the opinions of individuals by assigning them *opinion intervals*:

*Problem 1 (*BEST INTERVAL APPROXIMATION *[4]).* Given a signed graph $G = (V, E^+ \cup E^-)$, find a vertex mapping $\mathcal{I}$ such that $\forall v \in V : I_v \coloneqq \mathcal{I}(v) \subseteq \mathbb{R}$ are intervals on the reals that maximise

$$\mathsf{agree}(G, \mathcal{I}) = \sum_{\{u,v\} \in E^+} \mathbb{1}(I_u \cap I_v \neq \emptyset) + \sum_{\{u,v\} \in E^-} \mathbb{1}(I_u \cap I_v = \emptyset), \tag{1}$$

where $\mathbb{1}(P)$ is the indicator function, which takes the value 1 if P is true and 0 otherwise.

In our previous work, we showed a general hardness result, a polynomial-time approximation scheme (PTAS) for complete signed graphs and a fixed number of intervals, and developed scalable heuristics that correctly represent a considerably larger fraction of interactions in real-world networks than CORRELATION CLUSTERING baselines. However, our work also left several open questions, some of which we address in this paper. In particular, we provide the following results.

- For complete graphs, we generalise the PTAS from a fixed to an arbitrary number of intervals.
- We disprove our conjecture: "For any signed graph, it holds that an optimal solution to BIA satisfies at least a 3/4-fraction of the edges." [4].
- We present efficient graph pruning rules that empirically remove up to 84% of vertices and 60% of edges in real-world social network graphs, and imply efficient heuristic algorithms for general BEST INTERVAL APPROXIMATION.

We first present our generalised PTAS in Section 3.1, followed by disproving the conjecture in Section 3.2. We then introduce our graph pruning rules and the resulting heuristics in Section 4, and demonstrate the practical use of these contributions experimentally in Section 5.

Remark 1. The ideas in Theorems 2 and 3 were developed through iterative dialogues with AI assistants. All claims in the manuscript were verified and manually proven by the authors.

2 Background

In this section, we provide the necessary preliminaries for the rest of the paper and give a brief overview of related work and existing results for BIA.

2.1 Preliminaries

A *signed graph* $G = (V, E^+ \cup E^-)$ is given by its vertices V, positive edges E^+, and negative edges E^-, where $E^+ \cap E^- = \emptyset$. The signed graph G is *complete* if $E^+ \cup E^- = \binom{V}{2}$. For $u \in V$, we write $N^+(u)$ to denote the neighbours of u

in E^+ and $N^-(u)$ to denote its neighbours in E^-. We let $\deg^+(u) = |N^+(u)|$ and $\deg^-(u) = |N^-(u)|$ denote its positive and negative degrees, respectively, and $\deg(u) = \deg^+(u) + \deg^-(u)$ its total degree. For a subset $S \subseteq V$, we write $G[S]$ for the induced subgraph with vertex set S and edge set consisting of all edges in $E^+ \cup E^-$ with both endpoints in S. Further, we write $G - v := G[V \setminus \{v\}]$ for the deletion of a vertex v. For a graph G, we write $\mathcal{C}(G)$ for the vertex sets of the connected components, so that $\{G[C] : C \in \mathcal{C}(G)\}$ are the connected components. For the rest of this paper, we use the conventions that $n = |V|$ and $N = \binom{n}{2}$. Hence, for complete graphs, with $E = E^+ \cup E^-$, it holds that $|E| = N$.

An unsigned graph $G = (V, E)$ is an *interval graph* if we can assign an interval $I_v \subseteq \mathbb{R}$ to each vertex $v \in V$ such that for all $u, v \in V$, it holds that $\{u, v\} \in E$ if and only if $I_u \cap I_v \neq \emptyset$. Without loss of generality, we define such an *interval representation* as a mapping $\mathcal{I} : v \mapsto [\ell_v, r_v] \subseteq \mathbb{R}$, where $\ell_v < r_v$ define a closed interval on the reals. For the remainder of this paper, we let $\mathfrak{I}_n$ denote the set of *labelled* interval graphs for some n, where the mapping of each interval to a vertex in the original graph G is fixed.

2.2 Best Interval Approximation

We define the *agreement* version of BIA as in Equation (1). We occasionally refer to the *disagreement* version of BIA, where we instead count *violated* edges for which $\mathcal{I}$ is inconsistent with G. For a given BIA instance G, we define $\mathsf{OPT}(G) = \max_{\mathcal{I}} \mathsf{agree}(G, \mathcal{I})$ as the agreement of the optimal solution. BIA is NP-hard, and the disagreement version is known to be approximation-hard within any constant factor [4].

For the agreement version, there exists a trivial 2-approximation: by either constructing $\mathcal{I}$ such that all intervals overlap or are pairwise disjoint, we obtain $\mathsf{agree}(G, \mathcal{I}) \geq \max\{|E^+|, |E^-|\} \geq |E|/2 \geq \mathsf{OPT}(G)/2$, and for G complete $\mathsf{agree}(G, \mathcal{I}) \geq N/2$. Besides the general problem, the *fixed-k* BIA restricts $\mathcal{I}$ to k unique intervals. There, we similarly define $\mathsf{OPT}_k(G)$ as the agreement of the optimal solution using k intervals. For this specific setting, a PTAS exists.

Theorem 1 (Fixed-k PTAS [4]). *For a complete signed graph G, $k \in \mathbb{N}$ and $\varepsilon, \delta \in (0, 1)$, there is an algorithm that, with probability at least $1 - \delta$, returns an interval representation using at most k distinct intervals with agreement at least $(1 - \varepsilon)\mathsf{OPT}_k(G)$, in time $2^{O(k^2 \log(k/(\varepsilon\delta))/\varepsilon^3)} \cdot n$.*

2.3 Related Work

We briefly ground BIA to related graph problems. For a more detailed description, we refer to our previous paper [4]. Correlation Clustering [3] is highly related to our work and is stated as follows: Given a signed graph $G = (V, E^+ \cup E^-)$, partition its vertices into disjoint clusters $C_1, \ldots, C_k \subseteq V$ such that the number of positive edges within the clusters C_i and the number of negative edges between different clusters C_i and C_j, $i \neq j$, is maximised. Here, the value of k can be picked by the algorithm. Correlation Clustering has received a lot of attention over the past two decades in social network analysis and image segmentation, spanning approximation algorithms [8,12,13,14,17,29], more expressive formulations [5], and results in dynamic, online, parallel, and streaming settings [2,9,10,11,23,24,28].

The similarity between the two problems is clear: As any solution for correlation clustering can trivially be reduced to an interval assignment, this implies that the optimal solution of Best Interval Approximation will always yield an agreement at least as large as for Correlation Clustering. Conversely, BIA is strictly more expressive than Correlation Clustering. Interval representations can correctly model many non-transitive node relations, whereas standard clustering cannot. The simplest example of such a non-transitive relation is illustrated in Figure 1. Any clustering solution for the depicted *odd triangle* will need to violate either of the three edge constraints.

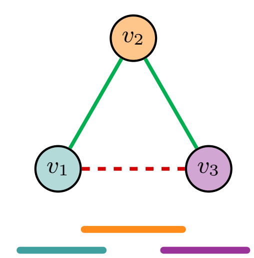

Fig. 1: A triangle with one negative edge and its interval representation.

Another closely related problem to BIA, especially on complete graphs, is that of Interval Editing. In this problem, the task is to transform an unsigned graph into an interval graph using a minimum number of edge deletions and insertions. This problem is known to be NP-hard [16], and it is fixed-parameter tractable (FPT) when parameterised by the number of edge insertions and deletions [6,7,21,31].

If G is a complete signed graph, then it can be represented without error in Best Interval Approximation if and only if $G^+ = (V, E^+)$ is an interval graph. This is because missing edges in G^+ correspond to negative edges in G (since G is complete). Thus, making the minimum number of edge deletions/insertions to turn G^+ into an interval graph is equivalent to flipping the minimum number of edge signs in G such that we have agreement for all edges. Hence, for complete graphs, we can rely on the rich literature on Interval Editing, which asks for the minimum number of edge changes to G^+ to make it an interval graph. In particular, this connection implies that BIA is FPT for complete graphs when only allowing a fixed number of sign changes (in one direction).

3 Extended Theoretical Results

3.1 A General PTAS on Complete Graphs

We resolve one of the questions left open in our previous work [4] about BIA: the *agreement* version of Best Interval Approximation admits a PTAS on complete graphs, even when the number of distinct intervals is *not* fixed. Throughout this section, let $G = (V, E^+ \cup E^-)$ be a *complete* signed graph. Our main result builds on the fixed-k PTAS in Theorem 1, but removes the dependence on k.

Theorem 2. *For a complete signed graph G and $\varepsilon, \delta \in (0,1)$, there is an algorithm that, with probability at least $1-\delta$, returns an interval assignment $\mathcal{I}$ such that $\mathsf{agree}(G,\mathcal{I}) \geq (1-\varepsilon)\mathsf{OPT}(G)$, in time $2^{O(\varepsilon^{-7}\log(1/(\varepsilon\delta)))} \cdot n$. Hence, the agreement version of* Best Interval Approximation *admits a randomised PTAS on complete graphs.*

Proof (sketch). The argument relies on a structural *rounding lemma* (Lemma 1): For a given integer $t \leq n$, we can "round" any general interval assignment $\mathcal{I}$ to $2t$ unique values for interval borders. This results in $\mathcal{O}(t^2)$ distinct intervals, yet importantly only reduces agreement by $\mathcal{O}(n^2/t)$. The theorem then follows by applying Theorem 1 for $k = \mathcal{O}(1/\varepsilon^2)$ and approximating $\mathsf{OPT}_k(G)$. By Lemma 1, we then bound the gap between $\mathsf{OPT}_k(G)$ and $\mathsf{OPT}(G)$. We provide a full proof in Appendix A.

This PTAS addresses a theoretical gap, yet remains unsuitable for practical use. In Section 4, we therefore turn to a pruning-based heuristic that scales to large graphs, and works without specifying the interval structure in advance.

3.2 Disproving the 3/4-Conjecture

In our earlier work [4], we conjectured that the optimal solution for Best Interval Approximation will always satisfy at least a $\frac{3}{4}$-fraction of the edges.

Conjecture 1 (3/4-Conjecture [4, Section 5]). For every BIA instance $G = (V, E^+ \cup E^-)$ with $E = E^+ \cup E^-$ it holds that $\mathsf{OPT}(G) \geq 3/4|E|$, and for complete graphs with $N = \binom{n}{2}$ it holds that $\mathsf{OPT}(G)/N \geq 3/4$.

Theorem 3 (The 3/4-Conjecture is false). *Let $\mathfrak{I}_n$ denote the set of labelled interval graphs over n vertices. Let $G = (V, E^+ \cup E^-)$ be a complete signed graph with $N = \binom{n}{2}$ edges, where edge-signs are chosen independently at random, i.e.,*

$$\forall \{u,v\} \in \binom{V}{2}: \quad \Pr[\{u,v\} \in E^+] = \Pr[\{u,v\} \in E^-] = \frac{1}{2}.$$

Then, with probability at least $1 - \frac{1}{|\mathfrak{I}_n|}$, it holds that

$$\frac{\mathsf{OPT}(G)}{N} \leq \frac{1}{2} + O\left(\sqrt{\frac{\log n}{n}}\right) \xrightarrow[n\to\infty]{} \frac{1}{2}.$$

Proof (sketch). The result follows from the fact that for any fixed interval assignment $\mathcal{I}$, the expectation $\mathbb{E}[\mathsf{agree}(G,\mathcal{I})]/N = \frac{1}{2}$. We then apply a Hoeffding bound for the tail of this random variable and a union bound over all possible, structurally unique interval representations in $\mathfrak{I}_n$. In essence, $\mathfrak{I}_n$ does not grow quickly enough with n to ensure that at least one interval assignment results in high enough agreement. We provide a full proof in Appendix B.

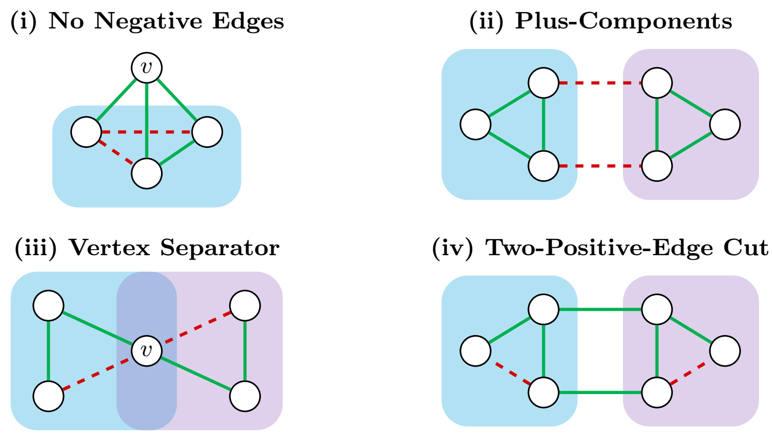

Fig. 2: Overview of the pruning rules.

This result is closely connected to bounds on the number of interval graphs [18,32] and to the edit distance of hereditary graph properties [1,30]. Theorem 3 is a stronger statement than just disproving the conjecture: it shows that the trivial $N/2$ solution is asymptotically tight in this random model. This also implies that the asymptotic bound for BIA on large random graphs is the same as that for CORRELATION CLUSTERING.

4 Pruning-Based Heuristics

To complement our extended theoretical results from the previous section, we now provide a scalable heuristic algorithm to solve the general BIA problem, where the interval structure is not fixed. Specifically, we present a family of *pruning rules* that decompose problem instances into smaller, independent subproblems without introducing edge violations.

Further, we show that these rules admit a *relaxed* variant that shrinks the instance further at the cost of a controlled number of violations. Relaxed pruning can be used either as a standalone heuristic for BIA or as a preprocessing step that shrinks the problem size for existing solvers.

4.1 Pruning Rules

Let $G = (V, E^+ \cup E^-)$ be an instance of BIA, where we assume that G is connected (otherwise, connected components can be treated separately). Then, several pruning rules can be applied to reduce problem size. Each pruning rule transforms G into a set of edge-disjoint connected subgraphs $\mathcal{S}$, such that optimal solutions on all $H \in \mathcal{S}$ can be combined into an optimal solution for G.

(i) Vertices without negative neighbours. If a vertex $v \in V$ has no negative edges, i.e. $\deg^-(v) = 0$, we can remove it and solve the remaining subproblems $\mathcal{S} = \{G[C] : C \in \mathcal{C}(G - v)\}$. Given optimal assignments $\mathcal{I}_H$ for all $H \in \mathcal{S}$, we construct an optimal assignment for G by combining all $\mathcal{I}_H$ and assigning v an interval I_v that contains all other intervals, thus satisfying all edges incident to v.

(ii) Positively connected components. If the vertex set V can be partitioned into two sets V_1 and $V_2 = V \setminus V_1$ such that no positive edge connects V_1 and V_2, the induced subproblems $\mathcal{S} = \{G[C] : C \in \mathcal{C}(G[V_1]) \cup \mathcal{C}(G[V_2])\}$ can be solved separately. To construct an optimal assignment on G, let $\mathcal{I}_H$ be an optimal assignment for each $H \in \mathcal{S}$, and place them on pairwise disjoint ranges of $\mathbb{R}$. Then, every edge between V_1 and V_2 is negative and satisfied by the disjointness.

(iii) Vertex separator. If a vertex $v \in V$ separates G, we can solve the subproblems $\mathcal{S} = \{G[C \cup \{v\}] : C \in \mathcal{C}(G - v)\}$ separately. To construct an optimal assignment on G, let $\mathcal{I}_H$ be an optimal assignment for each $H \in \mathcal{S}$. As affine maps preserve overlaps, we may apply one to each $\mathcal{I}_H$ so that they all give v the same interval. As no edge joins distinct components, their combination then satisfies exactly the edges the individual $\mathcal{I}_H$ do.

(iv) Two-positive-edge cut. If the vertex set V can be partitioned into two sets V_1 and $V_2 = V \setminus V_1$ such that at most two edges $\{a_j, b_j\}$ with $a_j \in V_1$, $b_j \in V_2$ cross the partition, all of which are positive, the subproblems $\mathcal{S} = \{G[C] : C \in \mathcal{C}(G[V_1]) \cup \mathcal{C}(G[V_2])\}$ can be solved separately. We can construct an optimal assignment on G by first combining optimal assignments of the connected components of $G[V_1]$ and $G[V_2]$ into $\mathcal{I}_1, \mathcal{I}_2$, and then applying an affine map to $\mathcal{I}_2$ under which every I_{b_j} overlaps I_{a_j}. This is possible as an affine map can take any two distinct points to any two other points. All other edges lie in one part and are thus unaffected.

Given these rules, we can prune any graph instance by exhaustively applying all four rules. This is done by starting with $\mathcal{S} = \{G\}$, and iteratively replacing graphs $H \in \mathcal{S}$ with the set of subproblems resulting from a pruning rule, discarding subgraphs without edges. An important property of these rules is that their combined exhaustive application results in a unique set of subproblems, regardless of the order in which the rules are applied.

Proposition 1. *The set of residual subgraphs resulting from exhaustively applying pruning rules* **(i)**–**(iv)** *is independent of the order of rule application.*

Proof (sketch). We characterise an edge as *deletable* if rule (i), (ii) or (iv) can delete it in its biconnected component. Removing edges preserves deletability. A terminal $\mathcal{S}^*$ consists of the biconnected components of its edges and contains no deletable edge. We then show that any edge deleted in one sequence of rule applications but retained in the terminal state of another sequence must also be deletable there. This contradicts terminality, showing that all terminal states must be equal. We provide a full proof in Appendix C.

4.2 Relaxed Pruning

While these pruning rules reduce problem size without introducing violations, they are applicable only in the presence of specific structures. To obtain a more flexible reduction, we relax rules (i) and (ii) to allow vertex removals when the conditions are not met exactly, at the cost of a controlled number of violations.

For a vertex v, define its *sign imbalance* by

$$r(v) := \frac{\max\{\deg^+(v), \deg^-(v)\}}{\deg(v)} .$$

Intuitively, $r(v)$ measures how close v is to exact pruning. If $\deg^+(v) \gg \deg^-(v)$, we assign v an interval containing all remaining intervals. This satisfies its positive edges and violates its negative edges. Conversely, if $\deg^-(v) \gg \deg^+(v)$, we assign v a disjoint interval, satisfying its negative edges and violating its positive edges. Once assigned, the relation of v to the remaining graph is fixed, so v can be removed. We therefore process vertices in decreasing order of $r(v)$, prioritising those whose removal incurs the smallest relative loss.

After each removal, we reapply the strict pruning rules to exhaustion. In practice, rules (i) and (ii) account for most of the reduction, while the others yield only marginal gains. Thus, we restrict the inner loop to these two rules for efficiency. While naively checking rule (ii) after each removal takes $O(|V|+|E^+|)$ time, it can be implemented in $O(\log^2 |V|)$ amortised time per positive edge removal using the data structure of Holm, de Lichtenberg, and Thorup [20].

To control the number of violations, we introduce an approximation threshold $\alpha \in [0.5, 1]$. A vertex is removed only if $r(v) \geq \alpha$, ensuring that at least an α-fraction of the edges removed by relaxed pruning are satisfied. Thus, α balances problem size reduction against violations. Notably, $\alpha = 1$ recovers exact pruning, while $\alpha = 0.5$ applies relaxed pruning until every vertex is removed, providing a fully heuristic solution. Algorithm 1 summarises the method.

Although this greedy heuristic generally does not produce optimal solutions, we show that running relaxed pruning with an approximation threshold α, optimally solving the resulting smaller independent subproblems, and combining their solutions yields an α-approximation to the globally optimal assignment. Hence, relaxed pruning retains at least an α-fraction of the best possible agreement. This is advantageous as optimal solutions may be much easier to obtain on small problem instances than on larger ones.

Proposition 2. *Let $G = (V, E^+ \cup E^-)$ be a signed graph and $\alpha \in [0.5, 1]$ an approximation threshold. Further, let $\widetilde{\mathcal{S}}$ be the set of residual subgraphs and ξ the number of violated edges returned by relaxed pruning on G with threshold α. Then, it holds that $\widetilde{\mathcal{S}}$ can satisfy at least an α-fraction of constraints of $\mathsf{OPT}(G)$:*

$$\mathsf{OPT}(\widetilde{\mathcal{S}}) := \sum_{H \in \widetilde{\mathcal{S}}} \mathsf{OPT}(H) + \big(|E \setminus \bigcup_{H \in \widetilde{\mathcal{S}}} E_H| - \xi\big) \geq \alpha \mathsf{OPT}(G).$$

Proof. Note that $|E \setminus \bigcup_{H \in \widetilde{\mathcal{S}}} E_H|$ is the number of edges removed during relaxed pruning, and thus $|E \setminus \bigcup_{H \in \widetilde{\mathcal{S}}} E_H| - \xi$ is the number of edges satisfied during

Algorithm 1: Relaxed Pruning

Input: Signed graph G, approximation threshold α
Output: Number of violated edges ξ, set of residual graphs $\widetilde{\mathcal{S}}$

$\mathcal{S} \leftarrow$ `PruneGraph`(G); `// Apply pruning exhaustively.`
$\xi \leftarrow 0$, $\widetilde{\mathcal{S}} \leftarrow \emptyset$;
while $\mathcal{S} \neq \emptyset$ **do**
  $H \leftarrow \mathcal{S}.\text{pop}()$;
  $v \leftarrow$ `FindMaxRatioVertex`(H);
  $r \leftarrow \max\left(\frac{\deg^+(v)}{\deg(v)}, \frac{\deg^-(v)}{\deg(v)}\right)$;
  if $r \geq \alpha$ **then**
    $\xi \leftarrow \xi + \min\{\deg^+(v), \deg^-(v)\}$;
    $H \leftarrow$ `RemoveVertex`(H, v);
    $\mathcal{S} \leftarrow \mathcal{S} \cup$ `PruneGraph`$_{\{i,ii\}}(H)$;
  end
  else
    $\widetilde{\mathcal{S}} \leftarrow \widetilde{\mathcal{S}} \cup \{H\}$;
  end
end
return ξ, $\widetilde{\mathcal{S}}$;

relaxed pruning. For any $\mathcal{I}_G \in \operatorname{argmax}_{\mathcal{I}}(\mathsf{agree}(G, \mathcal{I}))$ and $H \in \widetilde{\mathcal{S}}$, it holds that $\mathsf{OPT}(H) \geq \mathsf{agree}(H, \mathcal{I}_G)$. Thus, $\sum_{H \in \widetilde{\mathcal{S}}} \mathsf{OPT}(H) \geq \sum_{H \in \widetilde{\mathcal{S}}} \mathsf{agree}(H, \mathcal{I}_G)$. By definition of the approximation threshold $|E \setminus \bigcup_{H \in \widetilde{\mathcal{S}}} E_H| - \xi \geq \alpha |E \setminus \bigcup_{H \in \widetilde{\mathcal{S}}} E_H|$. Thus, we get that for any $\mathcal{I}_G \in \operatorname{argmax}_{\mathcal{I}}(\mathsf{agree}(G, \mathcal{I}))$

$$\begin{aligned}\sum_{H \in \widetilde{\mathcal{S}}} \mathsf{OPT}(H) + \big(|E \setminus \bigcup_{H \in \widetilde{\mathcal{S}}} E_H| - \xi\big) &\geq \sum_{H \in \widetilde{\mathcal{S}}} \mathsf{agree}(H, \mathcal{I}_G) + \alpha |E \setminus \bigcup_{H \in \widetilde{\mathcal{S}}} E_H| \\ &\geq \alpha \mathsf{OPT}(G). \end{aligned}$$ □

5 Experiments

In this section, we empirically evaluate our pruning rules and the relaxed pruning heuristic. In particular, we aim to answer the following research questions:

- **RQ1:** How much can the size of real-world social networks be reduced by applying our pruning rules?
- **RQ2:** How does relaxed pruning with varying approximation thresholds further reduce graph size at the cost of edge violations?
- **RQ3:** Can relaxed pruning be combined with existing BIA-algorithms to improve interval representations and reduce computation costs?

We evaluate our methods on several real-world signed social networks. We draw these datasets from the SNAP [27] and KONECT [22] repositories, together with the Bundestag dataset from our previous work [4]. A summary of these

Table 1: Effect of the pruning rules on the real-world datasets.

	Original			Pruned			Reduction	
Dataset	$\|V\|$	$\|E^+\|$	$\|E^-\|$	$\|V\|$	$\|E^+\|$	$\|E^-\|$	$\|V\|$	$\|E\|$
BitcoinOTC	5 878	18 281	3 153	1 079	6 167	2 132	81.6%	61.3%
Chess	6 822	19 046	13 604	4 489	16 820	12 215	34.2%	11.1%
WikiElec	7 115	78 440	21 915	3 234	62 208	19 257	54.5%	18.8%
Bundestag	1 480	320 956	76 541	1 398	303 168	76 541	5.5%	4.5%
Slashdot	82 062	380 933	117 599	26 137	226 401	103 026	68.1%	33.9%
Epinions	131 513	589 888	118 619	20 763	332 539	79 542	84.2%	41.8%
WikiSigned	138 564	628 000	84 337	22 659	229 724	59 857	83.6%	59.3%
WikiConflict	115 829	762 999	1 251 054	40 024	609 300	187 244	65.4%	60.5%

datasets is given in Table 1. Our code is available on GitHub[4] and preprocessing of graph instances and experimental setup are equivalent to our previous work [4].

RQ1: Reducing graph size. To investigate the practical impact of our pruning rules, we exhaustively apply them on real-world datasets. The sizes of the resulting pruned instances are shown in Table 1. On average, our methods reduce the number of vertices by 59.6% and the number of edges by 36.4%. This considerably shrinks the problem size without introducing any edge violations. The only instance where the size is barely reduced is Bundestag, which is also the densest graph and therefore offers fewer sparse cut structures for pruning. A further ablation study on the relative impact of the various pruning rules on reducing graph size is presented in Appendix D.

RQ2: Relaxed pruning. Next, we investigate how the graph size can be further reduced by accepting edge violations due to relaxed pruning. We test this by running relaxed pruning with varying approximation thresholds on the real-world datasets. For each approximation threshold, we measure the size of the remaining graph, as well as the number of violations introduced via the heuristic.

The results in Figure 3 show that, for a majority of datasets, the number of remaining edges decreases faster initially and then more slowly. The number of introduced violations grows roughly linearly with decreasing approximation threshold. Thus, the threshold induces residual graphs with different size–violation trade-offs. In particular, moderate threshold values can capture most of the size reduction while introducing only limited violations. The resulting residual graphs can be passed to further heuristic solvers, as studied in RQ3.

RQ3: Pruning and existing BIA-*algorithms.* Finally, we evaluate whether relaxed pruning can improve existing algorithms for Best Interval Approximation. We assess this by first using relaxed pruning with varying approximation

[4] https://github.com/Peter-Blohm/discovering_opinion_intervals/tree/kernelisation

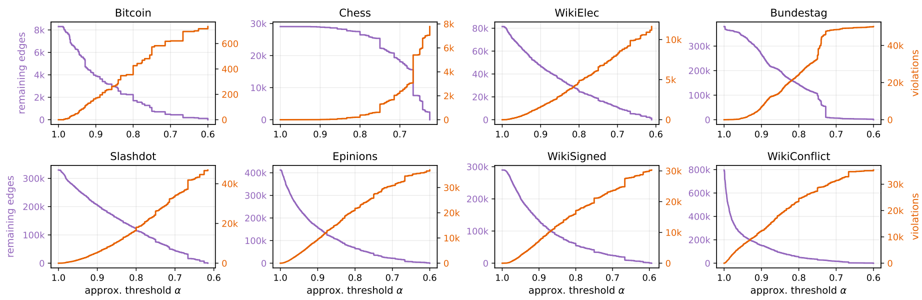

Fig. 3: Relaxed pruning trajectory on each dataset. Only pruning rules (i) and (ii) were applied after each vertex removal. The x-axis is the approximation threshold α: at $\alpha = 1$ no paid removal has happened yet. Lowering the threshold removes more vertices but accrues more violations.

thresholds to reduce the problem size, and then applying existing solvers on the remaining residual graphs. In particular, we employ the `GAIA` and `VENUS` heuristics proposed in our previous work [4], with the same structure of 8 consecutively overlapping intervals and the same choice of hyperparameters.

Table 2 shows that, for every dataset, pruning the graph improves the quality of the final interval representations. Further, for 7 out of 8 datasets, the best solution was found with approximation thresholds of 0.8 or 0.9. This indicates that the violations introduced during relaxed pruning can be offset by the improved performance of the heuristics on the smaller residual graphs. Notably, relaxed pruning allows for more general interval structures than the standard `GAIA`/`VENUS` approach using a fixed structure. This may contribute to the improved objective values obtained by the combined procedure. Lower thresholds, however, tend to hurt performance, as the violations introduced during relaxed pruning dominate the benefit of further size reduction.

Table 2: Lowest fraction of violated edges per dataset (in percent). The α columns apply relaxed pruning with that approximation threshold first and then solve the residual graphs with `GAIA`/`VENUS` for 50 random seeds. Reported violations are the sum of relaxed pruning violations and best of 50 `GAIA`/`VENUS` violations.

Dataset	no pruning		$\alpha=1.0$		$\alpha=0.9$		$\alpha=0.8$		$\alpha=0.7$		$\alpha=0.6$		$\alpha=0.5$	
	`GAIA`	`VENUS`	`GAIA`	`VENUS`	`GAIA`	`VENUS`	`GAIA`	`VENUS`	`GAIA`	`VENUS`	`GAIA`	`VENUS`	`GAIA`	`VENUS`
BitcoinOTC	3.32	3.55	2.98	2.92	2.74	**2.69**	2.86	2.85	3.05	3.05	3.42	3.42	3.42	3.42
Chess	19.82	19.73	19.23	19.11	19.13	19.00	19.04	**18.80**	19.18	19.06	23.83	23.83	23.83	23.83
WikiElec	11.24	11.26	10.99	10.98	10.60	**10.58**	10.60	10.59	10.82	10.80	11.58	11.58	11.58	11.58
Bundestag	0.25	0.25	0.52	**0.24**	1.88	1.87	6.15	6.15	12.21	12.21	12.62	12.62	12.62	12.62
Slashdot	9.07	8.94	8.44	8.39	7.77	7.74	7.72	**7.71**	8.32	8.30	9.41	9.41	9.41	9.41
Epinions	4.51	4.43	4.20	4.19	4.18	**4.16**	4.50	4.49	4.82	4.82	5.22	5.22	5.22	5.22
WikiSigned	4.97	4.88	3.87	3.81	3.86	**3.76**	3.84	3.80	3.94	3.91	4.24	4.24	4.25	4.25
WikiConflict	3.46	3.44	3.26	3.26	2.03	2.01	1.70	**1.69**	1.70	1.69	1.76	1.76	1.76	1.76

Besides facilitating better interval representations, applying graph pruning to shrink the problem size can also reduce the runtime of heuristic solvers. In Figure 4, we plot the total runtime of relaxed pruning followed by the `GAIA`/`VENUS` heuristics. For every dataset except the dense Bundestag, reducing the approximation threshold reduces the overall runtime, indicating that the cost of running more pruning steps is outweighed by the time savings due to applying the heuristic solvers on smaller residual graphs. Hence, preprocessing social networks with relaxed pruning before applying downstream solvers can simultaneously lead to both (i) better objectives and (ii) faster runtimes.

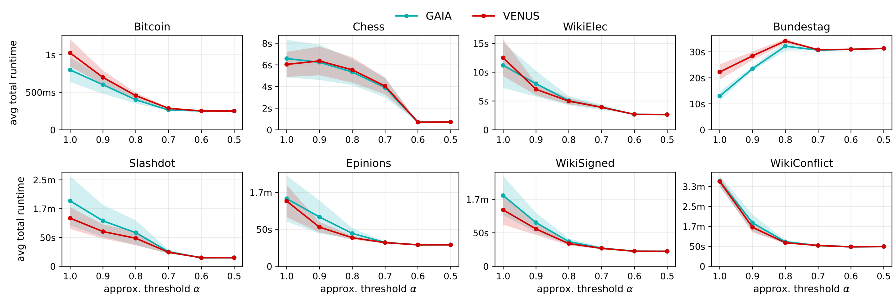

Fig. 4: Average total runtime (relaxed pruning followed by `GAIA`/`VENUS`) on each dataset as a function of the approximation threshold α. Shaded regions indicate the standard deviation over 50 runs. Lowering α shrinks the residual graphs and thereby the solver runtime on all datasets except the dense Bundestag.

6 Conclusion

In this paper, we closed a conceptual gap in the existing results for Best Interval Approximation by providing algorithmic results for the general problem setting: a PTAS for the agreement version in Theorem 2 and a scalable heuristic based on graph pruning rules in Section 4. We further showed a negative result for the existing 3/4-conjecture in Theorem 3. One question that remains for future work is the fully general case of BIA: *can we obtain a nontrivial approximation to* OPT *when G is not necessarily complete?* A further possible direction for future work would be a better heuristic for the general case, which builds on `GAIA` and `VENUS`, as Algorithm 1 shows a notable breakdown for dense graphs, where the pruning rules generally do not apply. In the same way as `GAIA` and `VENUS` are intuitive relaxations of Theorem 1, it might be possible to instead model interval *borders* individually and rely on Lemma 1 to obtain high-quality solutions for large instances.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alon, N., Stav, U.: What is the furthest graph from a hereditary property? Random Structures & Algorithms **33**(1), 87–104 (2008)
2. Assadi, S., Shah, V., Wang, C.: Streaming algorithms and lower bounds for estimating correlation clustering cost. In: NeurIPS (2023)
3. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Mach. Learn. **56**(1-3), 89–113 (2004)
4. Blohm, P., Chen, F., Gionis, A., Neumann, S.: Discovering opinion intervals from conflicts in signed graphs. Advances in Neural Information Processing Systems (2025)
5. Bonchi, F., Gionis, A., Ukkonen, A.: Overlapping correlation clustering. Knowledge and Information Systems **35**, 1–32 (2011)
6. Cao, Y.: Unit interval editing is fixed-parameter tractable. Inf. Comput. **253**, 109–126 (2017)
7. Cao, Y., Marx, D.: Interval deletion is fixed-parameter tractable. ACM Trans. Algorithms **11**(3), 21:1–21:35 (2015)
8. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. FOCS pp. 524–533 (2003)
9. Cohen-Addad, V., Lattanzi, S., Maggiori, A., Parotsidis, N.: Online and consistent correlation clustering. In: ICML. vol. 162, pp. 4157–4179 (2022)
10. Cohen-Addad, V., Lattanzi, S., Maggiori, A., Parotsidis, N.: Dynamic correlation clustering in sublinear update time. In: ICML (2024)
11. Cohen-Addad, V., Lattanzi, S., Mitrovic, S., Norouzi-Fard, A., Parotsidis, N., Tarnawski, J.: Correlation clustering in constant many parallel rounds. In: ICML. vol. 139, pp. 2069–2078 (2021)
12. Cohen-Addad, V., Lee, E., Li, S., Newman, A.: Handling correlated rounding error via preclustering: A 1.73-approximation for correlation clustering. In: FOCS. pp. 1082–1104 (2023)
13. Cohen-Addad, V., Lee, E., Newman, A.: Correlation Clustering with Sherali-Adams. In: FOCS. pp. 651–661 (2022)
14. Cohen-Addad, V., Lolck, D.R., Pilipczuk, M., Thorup, M., Yan, S., Zhang, H.: Combinatorial correlation clustering. In: STOC. pp. 1617–1628. ACM (2024)
15. Diestel, R.: Graph Theory, Graduate Texts in Mathematics, vol. 173. Springer, 5 edn. (2017)
16. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
17. Giotis, I., Guruswami, V.: Correlation clustering with a fixed number of clusters. Theory Comput. **2**(13), 249–266 (2006)
18. Hanlon, P.: Counting interval graphs. Transactions of the American Mathematical Society **272**(2), 383–426 (1982)
19. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association **58**(301), 13–30 (1963)
20. Holm, J., De Lichtenberg, K., Thorup, M.: Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. Journal of the ACM (JACM) **48**(4), 723–760 (2001)
21. Kaplan, H., Shamir, R., Tarjan, R.E.: Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. SIAM J. Comput. **28**, 1906–1922 (1999)
22. Kunegis, J.: KONECT: the Koblenz network collection. WWW (2013)

23. Kuroki, Y., Miyauchi, A., Bonchi, F., Chen, W.: Query-efficient correlation clustering with noisy oracle. In: NeurIPS (2024)
24. Lattanzi, S., Moseley, B., Vassilvitskii, S., Wang, Y., Zhou, R.: Robust online correlation clustering. In: NeurIPS. pp. 4688–4698 (2021)
25. Leskovec, J., Huttenlocher, D.P., Kleinberg, J.M.: Predicting positive and negative links in online social networks. In: WWW. pp. 641–650 (2010)
26. Leskovec, J., Huttenlocher, D.P., Kleinberg, J.M.: Signed networks in social media. In: CHI. pp. 1361–1370 (2010)
27. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data (Jun 2014)
28. Makarychev, K., Chakrabarty, S.: Single-pass pivot algorithm for correlation clustering. Keep it simple! In: NeurIPS (2023)
29. Swamy, C.: Correlation clustering: maximizing agreements via semidefinite programming. In: SODA (2004)
30. Thomason, A.: Graphs, colours, weights and hereditary properties, p. 333–364. London Mathematical Society Lecture Note Series, Cambridge University Press (2011)
31. Villanger, Y., Heggernes, P., Paul, C., Telle, J.A.: Interval completion is fixed parameter tractable. SIAM J. Comput. **38**(5), 2007–2020 (2009)
32. Yang, J., Pippenger, N.: On the enumeration of interval graphs. Proceedings of the American Mathematical Society, Series B **4**(1), 1–3 (2017)

A Omitted proofs from Section 3.1

Theorem 2. *For a complete signed graph G and $\varepsilon, \delta \in (0,1)$, there is an algorithm that, with probability at least $1-\delta$, returns an interval assignment $\mathcal{I}$ such that $\mathsf{agree}(G,\mathcal{I}) \geq (1-\varepsilon)\mathsf{OPT}(G)$, in time $2^{O(\varepsilon^{-7}\log(1/(\varepsilon\delta)))} \cdot n$. Hence, the agreement version of* Best Interval Approximation *admits a randomised PTAS on complete graphs.*

Before proving the theorem, we show that interval assignments admit rounding of interval endpoints with small reductions in agreement.

Lemma 1 (Interval rounding). *For a complete signed graph G, let $\mathcal{I}$ be any interval assignment, and let $t \geq 1$ be an integer. Then there is an assignment $\mathcal{I}'$ using at most $\binom{t+1}{2}$ distinct intervals with $\mathsf{agree}(G,\mathcal{I}') \geq \mathsf{agree}(G,\mathcal{I}) - \frac{2n^2}{t}$.*

Proof. We first formalise a notion of rounding the interval assignment $\mathcal{I}$ to limit the number of distinct intervals. For a fixed interval assignment $\mathcal{I}$, we consider the set of its $2n$ labelled interval endpoints

$$B(\mathcal{I}) \coloneqq \biguplus_{v \in V} \{\ell_v, r_v\}.$$

We define a total order $\prec$ on $B(\mathcal{I})$ by numerical value. Among endpoints with equal numerical value, left endpoints precede right endpoints; all remaining ties are broken arbitrarily. To formalise rounding, write $b_1 \prec \cdots \prec b_{2n}$ for the endpoints in this order, and assign to each endpoint b_i the block index

$$q(b_i) \coloneqq \left\lceil \frac{it}{2n} \right\rceil, \qquad p \coloneqq \left\lceil \frac{2n}{t} \right\rceil.$$

Then q is non-decreasing in $\prec$, and each block index is assigned to at most p endpoints.

We define the rounded interval assignment $\mathcal{I}'$ by

$$I'_v \coloneqq [2q(\ell_v), 2q(r_v)+1].$$

As q takes only integral values between 1 and t, $\mathcal{I}'$ uses at most $\binom{t+1}{2}$ distinct, nonempty intervals.

Next, we bound the difference in agreement between $\mathcal{I}$ and $\mathcal{I}'$ by the amount of overlaps created/removed by rounding.

No overlaps are removed by rounding. Suppose that $I_u \cap I_v \neq \emptyset$. Since the intervals are closed,

$$\ell_u \leq r_v \qquad \text{and} \qquad \ell_v \leq r_u.$$

By the tie-breaking rule, these inequalities imply $\ell_u \prec r_v$ and $\ell_v \prec r_u$, and therefore

$$q(\ell_u) \leq q(r_v) \qquad \text{and} \qquad q(\ell_v) \leq q(r_u).$$

Consequently, $I'_u \cap I'_v \neq \emptyset$.

Overlaps created by rounding. Consider two disjoint intervals. W.l.o.g., assume $r_u < \ell_v$. Then $q(r_u) \le q(\ell_v)$. If their rounded intervals overlap, then

$$2q(\ell_v) \le 2q(r_u) + 1.$$

Since the block indices are integers, this implies $q(r_u) = q(\ell_v)$.

For fixed u, the block containing r_u contains at most $p-1$ other endpoints, and hence at most $p-1$ left endpoints ℓ_v that can create an overlap to the right of I_u. As every disjoint pair has a unique left-to-right orientation, every newly created overlap is counted exactly once. Thus, their number is at most

$$n(p-1) = n\left(\left\lceil \frac{2n}{t} \right\rceil - 1\right) \le \frac{2n^2}{t}.$$

Each change in overlaps can decrease $\mathsf{agree}(G, \cdot)$ by at most 1, while unchanged interval overlaps contribute equally under $\mathcal{I}$ and $\mathcal{I}'$. Hence

$$\mathsf{agree}(G, \mathcal{I}') \ge \mathsf{agree}(G, \mathcal{I}) - \frac{2n^2}{t}.$$

□

Proof (of Theorem 2). The theorem now follows by applying Theorem 1 for some k depending on ε and approximating $\mathsf{OPT}_k(G)$. By Lemma 1, we then bound the gap between $\mathsf{OPT}_k(G)$ and $\mathsf{OPT}(G)$, giving our result.

We recall that the algorithm from Theorem 1 for fixed k has a runtime of $2^{O(k^2 \log(k/(\varepsilon\delta))/\varepsilon^3)} \cdot n$ to find a solution $\mathcal{I}_k$ with $\mathsf{agree}(G, \mathcal{I}_k) \ge \mathsf{OPT}_k(G) - \frac{\varepsilon}{2}n^2$. Now let $t := \lceil 24/\varepsilon \rceil$ and $k := \binom{t+1}{2} = \mathcal{O}(1/\varepsilon^2)$. By Lemma 1, we then have that $\mathsf{OPT}_k(G) \ge \mathsf{OPT}(G) - \frac{\varepsilon n^2}{12}$. When we invoke Theorem 1, rewritten in additive form, with $\varepsilon' = \varepsilon/12$ and failure probability δ, we obtain a solution $\mathcal{I}$ with

$$\begin{aligned}
\mathsf{agree}(G, \mathcal{I}) \ge& \mathsf{OPT}_k(G) - \frac{\varepsilon'}{2}n^2 && (2)\\
=& \mathsf{OPT}_k(G) - \frac{\varepsilon}{24}n^2 && (3)\\
\ge& \mathsf{OPT}(G) - \frac{\varepsilon n^2}{12} - \frac{\varepsilon n^2}{24} && (4)\\
\ge& \mathsf{OPT}(G) - \frac{\varepsilon n^2}{8}. && (5)
\end{aligned}$$

Finally, to express this additive bound in multiplicative form, we observe that, for $n \ge 2$,

$$\mathsf{OPT}(G) \ge \frac{1}{2}\binom{n}{2} \ge \frac{n^2}{8}, \qquad (6)$$

because of the trivial lower bound for OPT, which implies that

$$\begin{aligned}
\mathsf{agree}(G, \mathcal{I}) \ge& \mathsf{OPT}(G) - \frac{\varepsilon n^2}{8} && (7)\\
\ge& \mathsf{OPT}(G) - \varepsilon\mathsf{OPT}(G) && (8)\\
=& (1-\varepsilon)\mathsf{OPT}(G) && (9)
\end{aligned}$$

with probability at least $1-\delta$. The runtime for invoking Theorem 1 is then by the choice of k equal to $2^{\mathcal{O}(k^2 \log(k/(\varepsilon\delta))/\varepsilon^3)} \cdot n = 2^{\mathcal{O}(\log(1/(\varepsilon\delta))/\varepsilon^7)} \cdot n$. $\square$

B Omitted proofs from Section 3.2

Theorem 3 (The 3/4-Conjecture is false). *Let $\mathfrak{I}_n$ denote the set of labelled interval graphs over n vertices. Let $G = (V, E^+ \cup E^-)$ be a complete signed graph with $N = \binom{n}{2}$ edges, where edge-signs are chosen independently at random, i.e.,*

$$\forall \{u,v\} \in \binom{V}{2}: \quad \Pr[\{u,v\} \in E^+] = \Pr[\{u,v\} \in E^-] = \frac{1}{2}.$$

Then, with probability at least $1 - \frac{1}{|\mathfrak{I}_n|}$, it holds that

$$\frac{\mathsf{OPT}(G)}{N} \leq \frac{1}{2} + O\left(\sqrt{\frac{\log n}{n}}\right) \xrightarrow[n\to\infty]{} \frac{1}{2}.$$

The result follows from two simple lemmas.

Lemma 2. *The number of labelled interval graphs with n vertices is bounded by*

$$|\mathfrak{I}_n| \leq (2n)! \leq (2n)^{2n}.$$

Proof. We can define an interval graph uniquely by a total ordering over the interval borders of V. For labelled graphs, we consequently have at most one graph for each permutation of the $2n$ elements. This shows the first inequality. The second inequality is straightforward. $\square$

Lemma 3. *Let $G = (V, E^+ \cup E^-)$ be a complete signed graph with random edge signs as in Theorem 3. Let $\mathcal{I}$ be a fixed interval assignment over $n = |V|$ vertices. Then it holds that $\mathsf{agree}(G, \mathcal{I})$ is a binomial random variable with distribution $Bin(N, \frac{1}{2})$, and consequently*

$$\mathbb{E}(\mathsf{agree}(G, \mathcal{I})) = \sum_{e \in E^+ \cup E^-} \Pr(\mathsf{agree}(e, \mathcal{I})) = \frac{N}{2}.$$

Proof. For each edge $e \in E^+ \cup E^-$ it holds that $\mathsf{agree}(e, \mathcal{I})$ is a Bernoulli random variable with success probability 0.5, dependent only on the sign of e. Consequently, $\mathsf{agree}(G, \mathcal{I})$ is the sum of N i.i.d. Bernoulli trials and hence follows $\mathrm{Bin}(N, \frac{1}{2})$. The expectation is then straightforward. $\square$

Definition 1 (Hoeffding Bound [19]). *Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{E}[X_i] = \mu$ and $\Pr(a \leq X_i \leq b) = 1$. Then*

$$\Pr\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \varepsilon\right) \leq 2e^{-2n\varepsilon^2/(b-a)^2}.$$

Proof (of Theorem 3). Given Lemmas 2 and 3, we can now show the result with a Hoeffding bound: The expectation for each interval assignment $\mathcal{I}$ is $\frac{N}{2}$ and, for large n, $|\mathfrak{I}_n|$ does not grow fast enough to ensure any interval graph is far enough from the expected agreement to satisfy the conjecture.

We now choose $\varepsilon = 3\sqrt{\frac{\ln 2n}{n}} = \mathcal{O}\left(\sqrt{\frac{\ln n}{n}}\right)$. In the Hoeffding bound in Definition 1, we let $b - a = 1$, $\mu = \frac{1}{2}$ and $X_i = \mathsf{agree}(e, \mathcal{I})$ to obtain, for $n \geq 2$,

$$\Pr\left(\left|\frac{1}{N}\mathsf{agree}(G,\mathcal{I}) - \frac{1}{2}\right| \geq \varepsilon\right) = \Pr\left(\left|\frac{1}{N}\sum_{i=1}^{N}\mathsf{agree}(e,\mathcal{I}) - \frac{1}{2}\right| \geq \varepsilon\right) \tag{10}$$
$$\leq 2\exp(-2N\varepsilon^2) \tag{11}$$
$$= 2\exp\left(-18\binom{n}{2}\frac{\ln 2n}{n}\right) \tag{12}$$
$$= 2\exp\left(-9(n-1)\ln 2n\right) \tag{13}$$
$$= 2(2n)^{-9(n-1)} \tag{14}$$
$$\leq \frac{1}{(2n)^{4n}} \tag{15}$$
$$\leq \frac{1}{|\mathfrak{I}_n|^2} \tag{16}$$

Given the previous bound for a fixed interval graph, we obtain our result via a union bound:

$$\Pr\left(\exists \mathcal{I} \in \mathfrak{I}_n : \frac{\mathsf{agree}(G,\mathcal{I})}{N} \geq \frac{1}{2} + \varepsilon\right) \leq \sum_{\mathcal{I}\in\mathfrak{I}_n}\Pr\left(\frac{\mathsf{agree}(G,\mathcal{I})}{N} \geq \frac{1}{2} + \varepsilon\right) \leq \frac{1}{|\mathfrak{I}_n|}. \tag{17}$$

□

C Omitted proofs from Section 4

Proposition 1. *The set of residual subgraphs resulting from exhaustively applying pruning rules* **(i)–(iv)** *is independent of the order of rule application.*

Recall that a run starts from $\mathcal{S} = \{G\}$ and repeatedly replaces a graph $H \in \mathcal{S}$ by the subproblems a pruning rule produces from H, discarding those without edges, until no rule applies. The resulting $\mathcal{S}^*$ is called *terminal*. We write $G_{\mathcal{S}} := \left(\bigcup_{H\in\mathcal{S}} V_H, \bigcup_{H\in\mathcal{S}} E_H\right)$ for the graph of the edges present in a set $\mathcal{S}$. A vertex v of a connected graph G is a *cut vertex* if $G - v$ is disconnected. The *biconnected components* of a graph are its maximal connected subgraphs without a cut vertex. Every edge lies in exactly one of them, and a connected graph with more than one biconnected component has a cut vertex [15], which makes rule (iii) applicable.

We call an edge e *deletable* in a graph H if one of the pruning rules (i), (ii), or (iv) can be applied to H to delete it, and *deletable* in $\mathcal{S}$ if it is deletable in its biconnected component in $G_{\mathcal{S}}$.

Lemma 4. *Let H' be a subgraph of H and let e be an edge of H' that is deletable in H. Then e is deletable in H'.*

Proof. The applicability of rules (i), (ii) and (iv) is retained in subgraphs, as they require the absence of (i) negative neighbours of a vertex, (ii) positive edges crossing a partition, or (iv) negative or more than two positive edges crossing a partition. As subgraphs do not add edges, the same rule deletes e in H'. $\square$

Lemma 5. *In every $\mathcal{S}$ reachable from $\{G\}$, the graphs have pairwise disjoint edge sets, and the biconnected component of an edge of $G_{\mathcal{S}}$ is a subgraph of the graph of $\mathcal{S}$ containing it.*

Proof. We induct on the rule applications, the claim being immediate for $\mathcal{S} = \{G\}$. Let $H \in \mathcal{S}$ be replaced, producing $\mathcal{S}'$. By construction of the pruning rules, the replacements have disjoint edge-subsets in E_H. As $G_{\mathcal{S}'} \subseteq G_{\mathcal{S}}$, a biconnected component B of $G_{\mathcal{S}'}$ lies by induction in some graph of $\mathcal{S}$. If that graph is H, consider the subgraphs replacing it. For rules (i), (ii) and (iv) the subgraphs are vertex-disjoint with no edges between them. Since B is connected, it must lie in one of them. Further, as B is biconnected, it cannot be split by rule (iii). $\square$

Lemma 6. *If $\mathcal{S}^*$ is terminal, its graphs are exactly the biconnected components of $G_{\mathcal{S}^*}$, and no edge is deletable in $\mathcal{S}^*$.*

Proof. By construction, every $H \in \mathcal{S}^*$ is connected and has an edge. As rule (iii) does not apply, H is biconnected. Fix $e \in E_H$, and let B be its biconnected component in $G_{\mathcal{S}^*}$. By Lemma 5, $B \subseteq H$. Since H is itself biconnected, from maximality of biconnected components it follows that $B = H$. As the edge sets of $\mathcal{S}^*$ partition that of $G_{\mathcal{S}^*}$, its graphs are exactly the biconnected components of $G_{\mathcal{S}^*}$, so $\mathcal{S}^*$ is determined by its edges. Finally, toward a contradiction, assume an edge is deletable in $\mathcal{S}^*$. Then, it is also deletable in the H containing it, so rule (i), (ii) or (iv) would apply to H, contradicting the terminal condition. $\square$

Proof (of Proposition 1). Rules (i), (ii) and (iv) each delete at least one edge, and rule (iii) replaces a graph with subgraphs with strictly fewer vertices. Hence, any exhaustive application of pruning rules terminates.

Let $\mathcal{S}_1^*, \mathcal{S}_2^*$ be the terminal sets of any two exhaustive sequences of rule applications. Toward a contradiction, assume $\mathcal{S}_1^* \neq \mathcal{S}_2^*$, so that the terminal set of subgraphs is different. By Lemma 6 both are determined by their edges, so w.l.o.g. some edge e occurs in $\mathcal{S}_1^*$ but not in $\mathcal{S}_2^*$. Let it be the first such edge that the second run deletes, say from the state $\mathcal{S}'$. That deletion applies a rule to some $H \in \mathcal{S}'$, so e is deletable in H. By Lemma 5, the biconnected component of e in $G_{\mathcal{S}'}$ is a subgraph of H, and thus by Lemma 4, e is also deletable in $\mathcal{S}'$. As e is the first edge in $G_{\mathcal{S}_1^*}$ that the second run deletes, $G_{\mathcal{S}_1^*}$ is a subgraph of $G_{\mathcal{S}'}$. Hence, the biconnected component of e in $G_{\mathcal{S}_1^*}$ lies in the one in $G_{\mathcal{S}'}$, and e is deletable in $\mathcal{S}_1^*$ by Lemma 4, contradicting Lemma 6. Hence $\mathcal{S}_1^* = \mathcal{S}_2^*$. $\square$

D Additional Experiment Results

Effect of pruning rules. To investigate the relative impact of the individual pruning rules, we cumulatively apply them in order (i)–(iv) and measure the remaining vertices after each additional rule is applied. The results in Table 3 show that rules (i) and (ii) account for the largest reduction in problem size, while rules (iii) and (iv) yield only marginal further reduction, removing fewer than 100 vertices on even the largest graphs.

Table 3: Remaining vertices after cumulatively applying pruning rules, starting from the original number of vertices $|V|$. (i) removes vertices without negative edges, (ii) solves plus-components separately, (iii) splits at vertex separators, and (iv) applies two-positive-edge cuts.

Dataset	$\lvert V\rvert$	(i)	(+ii)	(+iii)	(+iv)
BitcoinOTC	5 878	1 571	1 079	1 079	1 079
Chess	6 822	5 647	4 505	4 505	4 489
WikiElec	7 115	4 220	3 234	3 234	3 234
Bundestag	1 480	1 398	1 398	1 398	1 398
Slashdot	82 062	35 113	26 163	26 163	26 137
Epinions	131 513	43 052	20 847	20 845	20 763
WikiSigned	138 564	41 667	22 757	22 691	22 659
WikiConflict	115 829	115 829	40 076	40 032	40 024