Maximizing the strong triadic closure in split graphs and proper interval graphs*

Athanasios L. Konstantinidis† Charis Papadopoulos‡

Abstract

In social networks the Strong Triadic Closure is an assignment of the edges with strong or weak labels such that any two vertices that have a common neighbor with a strong edge are adjacent. The problem of maximizing the number of strong edges that satisfy the strong triadic closure was recently shown to be NP-complete for general graphs. Here we initiate the study of graph classes for which the problem is solvable. We show that the problem admits a polynomial-time algorithm for two incomparable classes of graphs: proper interval graphs and trivially-perfect graphs. To complement our result, we show that the problem remains NP-complete on split graphs, and consequently also on chordal graphs. Thus, we contribute to define the first border between graph classes on which the problem is polynomially solvable and on which it remains NP-complete.

1 Introduction

Predicting the behavior of a network is an important concept in the field of social networks [9]. Understanding the strength and nature of social relationships has found an increasing usefulness in the last years due to the explosive growth of social networks (see e.g., [2]). Towards such a direction the Strong Triadic Closure principle enables us to understand the structural properties of the underlying graph: it is not possible for two individuals to have a strong relationship with a common friend and not know each other [12]. Such a principle stipulates that if two people in a social network have a "strong friend" in common, then there is an increased likelihood that they will become friends themselves at some point in the future. Satisfying the Strong Triadic Closure is to characterize the edges of the underlying graph into weak and strong such that any two vertices that have a strong neighbor in common are adjacent. Since users interact and actively engage in social networks by creating strong relationships, it is natural to consider the MaxSTC problem: maximize the number of strong edges that satisfy the Strong Triadic Closure. The problem has been shown to be NP-complete for general graphs while its dual problem of minimizing the number of weak edges admits a constant factor approximation ratio [32]. More recently, interesting variations of the MaxSTC problem have been considered which impose additional information than the network structure (e.g., through a collection of strong subgraphs) [30, 31].

*A preliminary version of this paper appeared as an extended abstract in the proceedings of ISAAC 2017. The research work done by A. L. Konstantinidis is co-financed by Greece and the European Union (European Social Fund – ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning" in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" (MIS–5000432), implemented by the State Scholarships Foundation (IKY). The research work done by C. Papadopoulos was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant", Project FANTA (eFficient Algorithms for NeTwork Analysis), number HFRI-FM17-431.

†Department of Mathematics, University of Ioannina, Greece. E-mail: `a.konstantinidis@uoi.gr`

‡Department of Mathematics, University of Ioannina, Greece. E-mail: `charis@uoi.gr`

In this work, we initiate the computational complexity study of the MaxSTC problem in important classes of graphs. If the input graph is a P_3-free graph (i.e., a graph having no induced path on three vertices which is equivalent with a graph that consists of vertex-disjoint union of cliques) then there is a trivial solution by labeling strong all the edges. Such an observation might falsely lead into a graph modification problem, known as Cluster Deletion problem (see e.g., [3, 14]), in which we want to remove the minimum number of edges that correspond to the weak edges, such that the resulting graph does not contain a P_3 as an induced subgraph. More precisely the obvious reduction would consist in labeling the deleted edges in the instance of Cluster Deletion as weak, and the remaining ones as strong. However, this reduction fails to be correct due to the fact that the graph obtained by deleting the weak edges in an optimal solution of MaxSTC may contain an induced P_3, so long as those three vertices induce a triangle in the original graph (prior to deleting the weak edges). We stress that there are examples on split graphs (Figure 1) and proper interval graphs (Figure 4) showing such a difference.

To the best of our knowledge, no previous results were known prior to our work when restricting the input graph for the MaxSTC problem. It is not difficult to see that for bipartite graphs the MaxSTC problem has a simple polynomial-time solution by considering a maximum matching that represent the strong edges [15]. In fact such an argument regarding the maximum matching generalizes to the larger class of triangle-free graphs. Also notice that for triangle-free graphs a set of edges is a maximum matching if and only if it is formed by a solution for the Cluster Deletion problem. It is well-known that a maximum matching of a graph corresponds to a maximum independent set of its line graph that represents the adjacencies between the edges [10]. As previously noted, for general graphs it is not necessarily the case that a maximum matching corresponds to the optimal solution for MaxSTC. Here we show a similar characterization for MaxSTC by considering the adjacencies between the edges of a graph that participate in induced P_3's. Such a characterization allows us to exhibit structural properties towards an optimal solution of MaxSTC.

Due to the nature of the P_3 existence that enforce the labeling of weak edges, there is an interesting connection to problems related to the *square root* of a graph; a graph H is a *square root* of a graph G and G is the *square* of H if two vertices are adjacent in G whenever they are at distance one or two in H. Any graph does not have a square root (for example consider a simple path), but every graph contains a subgraph that has a square root. Although it is NP-complete to determine if a given chordal graph has a square root [22], there are polynomial-time algorithms when the input is restricted to bipartite graphs [21], or proper interval graphs [22], or trivially-perfect graphs [26]. Among several square roots that a graph may have, one can choose the square root with the maximum or minimum number of edges [5, 24]. The relationship between MaxSTC and to that of determining square roots can be seen as follows. In the MaxSTC problem we are given a graph G and we want to select the maximum possible number of edges, at most one from each induced P_3 in G. Thus we need to find the largest subgraph (in terms of the number of its edges) H of G such that the square of H is a subgraph of G. However the known results related to square roots were concerned with deciding if the whole graph has a (maximum or minimum) square root and there are no such equivalent formulations related to the largest square root.

Our main motivation is to understand the complexity of the problem on subclasses of chordal graphs, since the class of chordal graphs (i.e., graphs having no chordless cycle of length at least four) finds important applications in both theoretical and practical areas related to social networks [1, 19, 27]. More precisely two famous properties can be found in social networks. For most known social and biological networks their diameter, that is, the length of the longest shortest path between any two vertices of a graph, is known to be a small constant [17]. On the other hand it has been shown that the most prominent social network subgraphs are cliques, whereas highly infrequent induced subgraphs are cycles of length four [33]. Thus it is evident that subclasses of chordal graphs are close related to such networks, since they have rather small diameter (e.g., split graphs or trivially-perfect graphs) and are characterized by the absence of chordless cycles (e.g., proper interval graphs). Towards such a direction we show that MaxSTC is NP-complete on split graphs and consequently also on chordal graphs. On the positive side, we present the first polynomial-time algorithm for computing MaxSTC on proper interval graphs. Proper interval graphs, also known as unit interval graphs or indifference graphs, form a subclass of interval graphs and they are unrelated to split graphs [29]. By our result they form the first graph class, other than triangle-free graphs, for which MaxSTC is shown to be polynomial time solvable. In order to obtain our algorithm, we take advantage of their clique path (consecutive arrangement of maximal cliques) and apply a dynamic programming on subproblems defined by passing the clique path in its natural ordering. Our structural proofs together with its polynomial solution on proper interval graphs can be seen as useful tools towards settling the complexity of MaxSTC on interval graphs. Furthermore by considering the characterization of the induced P_3's mentioned earlier, we show that MaxSTC admits a simple polynomial-time solution on trivially-perfect graphs (i.e., graphs having no induced P_4 or C_4). Thus we contribute to define the first borderline between graph classes on which the problem is polynomially solvable and on which it remains NP-complete.

2 Preliminaries

All graphs considered here are simple and undirected. A graph is denoted by $G = (V, E)$ with vertex set V and edge set E. We use the convention that $n = |V|$ and $m = |E|$. The *neighborhood* of a vertex v of G is $N(v) = \{x \mid vx \in E\}$ and the *closed neighborhood* of v is $N[v] = N(v) \cup \{v\}$. For $S \subseteq V$, $N(S) = \bigcup_{v \in S} N(v) \setminus S$ and $N[S] = N(S) \cup S$. A graph H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For $X \subseteq V(G)$, the subgraph of G *induced* by X, $G[X]$, has vertex set X, and for each vertex pair u, v from X, uv is an edge of $G[X]$ if and only if $u \neq v$ and uv is an edge of G. For $R \subseteq E(G)$, $G \setminus R$ denotes the graph $(V(G), E(G) \setminus R)$, that is a subgraph of G and for $S \subseteq V(G)$, $G - S$ denotes the graph $G[V(G) - S]$, that is an induced subgraph of G. Two adjacent vertices u and v are called *twins* if $N[u] = N[v]$.

A *clique* of G is a set of pairwise adjacent vertices of G, and a *maximal clique* of G is a clique of G that is not properly contained in any clique of G. An *independent set* of G is a set of pairwise non-adjacent vertices of G. For $k \geq 2$, the chordless path on k vertices is denoted by P_k and the chordless cycle on k vertices is denoted by C_k. For an induced path P_k, the vertices of degree one are called endvertices. A vertex v is *universal* in G if $N[v] = V(G)$ and v is *isolated* if $N(v) = \emptyset$. The complete bipartite graph $K_{1,3}$ on four vertices is called *claw*. For two vertices u and v we say that u *sees* v if $\{u, v\} \in E(G)$; otherwise, we say that u *misses* v. We extend this notion to vertex sets: a set A sees (resp., misses) a vertex set B if every vertex of A is adjacent (resp., non-adjacent) to every vertex of B. We say that two *edges are non-adjacent* if they have no common endpoint; otherwise we call them *adjacent edges*.

Strong Triadic Closure. Given a graph $G = (V, E)$, a *strong-weak labeling* on the edges of G is a function λ that assigns to each edge of $E(G)$ one of the labels *strong* or *weak*; i.e., $\lambda : E(G) \to \{\text{strong}, \text{weak}\}$. An edge that is labeled strong (resp., weak) is simply called *strong* (resp. *weak*). A *strong triadic closure* of a graph G is a strong-weak labeling λ such that for any two strong edges $\{u, v\}$ and $\{v, w\}$ there is a (weak or strong) edge $\{u, w\}$. In

Figure 1: A split graph G is shown to the left side. The right side depicts a solution for MaxSTC on G where the weak edges are exactly the edges of G that are not shown.

other words, in a strong triadic closure there is no pair of strong edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E(G)$.

The problem of computing a maximum strong triadic closure, denoted by MaxSTC, is stated as follows:

MaxSTC

Input: A graph $G = (V, E)$.

Task: Find a strong-weak labeling of $E(G)$ that satisfies the strong triadic closure and has the maximum number of strong edges.

Note that its dual problem asks for the minimum number of weak edges. Here we focus on maximizing the number of strong edges in a strong triadic closure.

Let G be a strong-weak labeled graph. We denote by (E_S, E_W) the partition of $E(G)$ into strong edges E_S and weak edges E_W. The graph spanned by E_S is the graph $G \setminus E_W$. For a vertex $v \in V(G)$ we say that the *strong neighbors* of v are the other endpoints of the strong edges incident to v. We denote by $N_S(v) \subseteq N(v)$ the strong neighbors of v. Similarly we say that a vertex u is *strongly adjacent* to v if u is adjacent to v and the edge $\{u, v\}$ is strong.

Observation 2.1. *Let G be a strong-weak labeled graph with edge partition (E_S, E_W). Then G satisfies the strong triadic closure if and only if for every P_3 in $G \setminus E_W$, the vertices of P_3 induce a K_3 in G.*

Proof. Observe that $G \setminus E_W$ is the graph spanned by the strong edges. If for two strong edges $\{u, v\}$ and $\{v, w\}$, $\{u, w\} \notin E(G \setminus E_W)$ then $\{u, w\}$ is an edge in G and, thus, u, v, w induce a K_3 in G. On the other hand notice that any two strong edges of $G \setminus E_W$ are either non-adjacent or share a common vertex. If they share a common vertex then the vertices must induce a K_3 in G, implying that (E_S, E_W) satisfies the strong triadic closure. □

Therefore in the MaxSTC problem we want to minimize the number of the removal (weak) edges E_W from G such that every three vertices that induce a P_3 in $G \setminus E_W$ form a clique in G. Then it is not difficult to see that G satisfies the strong triadic closure if and only if for every vertex v, $N_S[v]$ induces a clique in G.

2.1 Basic Results

It is interesting to settle the complexity of MaxSTC on graphs of small structural parameter, such as treewidth. Towards such a direction observe that every problem expressible in monadic second order logic of second type (MSO_2) with quantification over vertex sets and edge sets can be solved in linear time for graphs of bounded treewidth [7]. Indeed, MaxSTC can be formulated in MSO_2: (i) the edges are partitioned into two subsets E_S, E_W (i.e., a strong-weak labeling), (ii) the endpoints of every path of length two spanned by the edges of E_S have an edge in G (i.e., satisfy the strong triadic closure), and (iii) $|E_S|$ is as large as possible. In particular, the following two expressions correspond to properties (i) and (ii), respectively:

- $\forall x \forall y \; adj(x,y) \rightarrow (E_W(x,y) \vee E_S(x,y)) \wedge (\neg E_W(x,y) \vee \neg E_S(x,y))$
- $\forall x \forall y \forall z \; (E_S(x,y) \wedge E_S(y,z)) \rightarrow adj(x,z)$

Therefore there is a linear-time algorithm for MaxSTC on graphs of bounded treewidth [7]. Notice that a similar observation holds for the Cluster Deletion problem in which the only difference is that the endpoints considered in case (ii) must have an edge of E_S (i.e., the strong edges span a P_3-free graph where the strong edges reflect the edges of the graph resulting after removing the weak edges). We remark that the graphs considered hereafter, such as split graphs, proper interval graphs, and trivially perfect graphs do not have bounded treewidth.

2.1.1 The line-incompatibility graph and twin vertices

We next provide some interesting characterizations related to the MaxSTC problem that might be of independent interest with respect to our main results concerning split graphs and proper interval graphs. First we give an equivalent transformation of MaxSTC related to the independent set of an auxiliary graph and, then, we show how to contract twin vertices. As a consequence of the former characterization, we show that its application leads to a polynomial solution for the class of trivially-perfect graphs.

Instead of maximizing the strong edges of the original graph G, we will look at the maximum independent set of the following graph that we call the *line-incompatibility* graph $\widehat{G}$ of G: for every edge of G there is a node in $\widehat{G}$ and two nodes of $\widehat{G}$ are adjacent if and only if the vertices of the corresponding edges induce a P_3 in G. In a different context the notion of line-incompatibility has already been considered under the term *Gallai graph* in [23] or as an auxiliary graph in [5, 32]. Note that the line-incompatibility graph of G is a subgraph of the line graph[1] of G. Moreover, observe that for a graph G, its line graph and its line-incompatibility graph coincide if and only if G is a triangle-free graph.

Proposition 2.2. *A subset S of edges $E(G)$ is an optimal solution for* MaxSTC *of G if and only if S is a maximum independent set of $\widehat{G}$.*

Proof. By Observation 2.1 for every P_3 in G at least one of its two edges must be labeled weak in S. This means that these two edges are adjacent in $\widehat{G}$ and they cannot belong to an independent set of $\widehat{G}$. On the other hand, by construction two nodes of $\widehat{G}$ are adjacent if and only if there is a P_3 in G. Thus the nodes of an independent set of $\widehat{G}$ can be labeled strong in G satisfying the strong triadic closure. □

Therefore, for the optimal solution of G one may look at a solution for a maximum independent set of $\widehat{G}$. Also note that for any graph G, computing a maximum independent set of $\widehat{G}$ is an NP-complete problem [23]. As a byproduct, if we are interested in minimizing the number of weak edges then we ask for the minimum vertex cover of $\widehat{G}$. To distinguish the vertices of $\widehat{G}$ with those of G, we refer to the former as nodes and to the latter as vertices. For an edge $\{u,v\}$ of G we denote by uv the corresponding node of $\widehat{G}$. Figure 2 shows a graph and its line-incompatibility graph.

Due to Proposition 2.2 it is natural to characterize the graphs for which their line-incompatibility graph is perfect. Such a characterization will lead to polynomial cases of MaxSTC, since the problem of finding a maximum independent set of perfect graphs admits

[1]The *line graph* of G is the graph having the edges of G as vertices and two vertices of the line graph are adjacent if and only if the two original edges are incident in G.

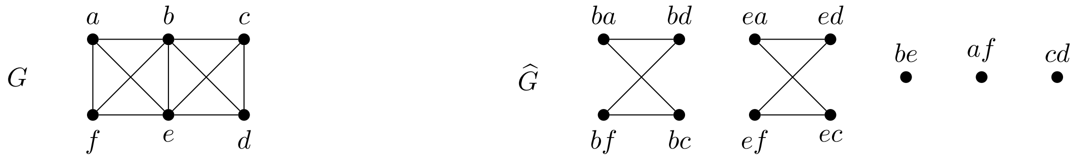

Figure 2: A graph G on the left side and its line-incompatibility graph $\widehat{G}$ on the right side. Notice that if the nodes of an independent set in $\widehat{G}$ are labeled strong and the rest are labeled weak then such a strong-weak labeling satisfies the strong triadic closure.

a polynomial solution [13]. A typical example is the class of bipartite graphs for which their line graph coincides with their line-incompatibility graph and it is known that the line graph of a bipartite graph is perfect (see for e.g., [4]). As we show next, another paradigm of this type is the class of trivially-perfect graphs.

A graph G is called *trivially-perfect* (also known as *quasi-threshold*) if for each induced subgraph H of G, the number of maximal cliques of H is equal to the maximum size of an independent set of H. It is known that the class of trivially-perfect graphs coincides with the class of (P_4, C_4)-free graphs, that is every trivially-perfect graph has no induced P_4 or C_4 [11]. Notice that the graph given in Figure 2 is trivially-perfect. A *cograph* is a graph without an induced P_4, that is a cograph is a P_4-free graph. Hence trivially-perfect graphs form a subclass of cographs.

Theorem 2.3. *The line-incompatibility graph of a trivially-perfect graph is a cograph.*

Proof. Let G be a trivially-perfect graph, that is G is a (P_4, C_4)-free graph. We will show that the line-incompatibility graph $\widehat{G}$ of G is a P_4-free graph. Consider any P_3 in $\widehat{G}$. Due to the construction of $\widehat{G}$, the P_3 has one of the following forms: (i) v_1v_2, v_2v_3, v_3v_4 or (ii) v_1x, v_2x, v_3x. We prove that the P_3 has the second form because G has no induced P_4 or C_4. If (i) applies then $\{v_1, v_3\}, \{v_2, v_4\} \notin E(G)$ and $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\} \in E(G)$ which implies that $v_4 \neq v_1$. Thus G contains a P_4 or a C_4 depending on whether there is the edge $\{v_1, v_4\}$ in G. Hence every P_3 in $\widehat{G}$ has the form v_1x, v_2x, v_3x where v_1, v_2, v_3, x are distinct vertices of G. Now, assume for contradiction that $\widehat{G}$ contains a P_4. Then the P_4 is of the form v_1x, v_2x, v_3x, v_4x because it contains two induced P_3's. The structure of the P_4 implies that $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\} \notin E(G)$ and $\{v_1, v_3\}, \{v_2, v_4\}, \{v_4, v_1\} \in E(G)$. This however shows that the vertices v_3, v_1, v_4, v_2 induce a P_4 in G leading to a contradiction that G is a (P_4, C_4)-free graph. Therefore $\widehat{G}$ is a P_4-free graph. □

By Theorem 2.3 and the fact that a maximum independent set of a cograph can be computed in linear time [6], MaxSTC can be solved in polynomial time on trivially-perfect graphs. We would like to note that the line-incompatibility graph of a cograph or a split graph or a proper interval graph is not necessarily a perfect graph. It is also interesting to note that the line-incompatibility graph of the complement of a bipartite graph (co-bipartite graph) is bipartite.

A natural contraction for several graph problems is to group twin vertices since they play the same role on the given graph. For the Cluster Deletion problem, such an approach has already been used [3, 28]. With the next result, we show that this is indeed the case for the MaxSTC problem. We denote by $I_{\widehat{G}}$ a maximum independent set of $\widehat{G}$.

Lemma 2.4. *Let x and y be twin vertices of a graph G. Then there is an optimal solution $I_{\widehat{G}}$ such that $xy \in I_{\widehat{G}}$ and for every vertex $u \in N(x)$, $xu \in I_{\widehat{G}}$ if and only if $yu \in I_{\widehat{G}}$.*

Proof. First, we show that xy is an isolated node in $\widehat{G}$. If xy is adjacent to xu then y is non-adjacent to u in G which contradicts the fact that x and y are twins. Thus xy is an isolated node in $\widehat{G}$ which implies $xy \in I_{\widehat{G}}$. For the second argument observe that for every vertex $u \in N(x)$, xu and yu are non-adjacent in $I_{\widehat{G}}$. Let $u \in N(x)$. Then notice that $u \in N(y)$. This means that if $xu \in I_{\widehat{G}}$ (resp., $yu \in I_{\widehat{G}}$) then yu (resp., xu) is a node of $\widehat{G}$. We define the following sets of nodes in $\widehat{G}$:

- Let A_x be the set of nodes xa such that $xa \in I_{\widehat{G}}$ and $ya \notin I_{\widehat{G}}$ and let A_y be the set of nodes ya such that $xa \in A_x$.
- Let B_y be the set of nodes yb such that $yb \in I_{\widehat{G}}$ and $xb \notin I_{\widehat{G}}$ and let B_x be the set of nodes xb such that $yb \in B_y$.

It is clear that $A_x \subseteq I_{\widehat{G}}$, $B_y \subseteq I_{\widehat{G}}$, and $A_x \cap B_y = \emptyset$. Also note that $|A_x| = |A_y|$ and $|B_y| = |B_x|$, since $N[x] = N[y]$.

Let $I_{xy} = I_{\widehat{G}} \setminus (A_x \cup B_y)$ so that $I_{\widehat{G}} = A_x \cup B_y \cup I_{xy}$. We show that every node of A_y is non-adjacent to any node of $I_{\widehat{G}} \setminus B_y$. Let ya be a node of A_y. If there is a node $az \in I_{\widehat{G}} \setminus B_y$ that is adjacent to ya then z and y are non-adjacent in G which implies that z and x are non-adjacent in G. This however leads to a contradiction because $xa, az \in I_{\widehat{G}}$ and xa is adjacent to az in $\widehat{G}$. If there is a node $yb \in I_{\widehat{G}}$ that is adjacent to ya then a is non-adjacent to b in G so that xa is also adjacent to xb in $\widehat{G}$. This means that $xb \notin I_{\widehat{G}}$ implying that $yb \in B_y$. Thus every node of A_y is non-adjacent to any node of $I_{\widehat{G}} \setminus B_y$ and with completely symmetric arguments, every node of B_x is non-adjacent to any node of $I_{\widehat{G}} \setminus A_x$. Hence both sets $I_1 = A_x \cup A_y \cup I_{xy}$ and $I_2 = B_x \cup B_y \cup I_{xy}$ form independent sets in $\widehat{G}$. By the facts that $|A_x| = |A_y|$ and $|B_y| = |B_x|$ we have $|I_1| \geq |I_{\widehat{G}}|$ whenever $|A_x| \geq |B_y|$ and $|I_2| \geq |I_{\widehat{G}}|$ whenever $|A_x| < |B_y|$. Therefore we can safely replace one of the sets A_x or B_y by B_x or A_y and obtain the solutions I_2 or I_1, respectively. Now observe that in both solutions I_1 and I_2 we have $xu \in I_i$ if and only if $yu \in I_i$, for $i \in \{1, 2\}$, and this completes the proof. $\square$

3 MaxSTC on split graphs

Here we provide an NP-hardness result for MaxSTC on split graphs. In fact, our main result stated in Theorem 3.4 implies that computing a maximum independent set of the line-incompatibility graph of a split graph is NP-hard, improving a result in [23] which states that computing a maximum independent set of the Gallai graph of a graph is NP-hard. A graph $G = (V, E)$ is a *split graph* if V can be partitioned into a clique C and an independent set I, where (C, I) is called a *split partition* of G. Split graphs form a subclass of the larger and widely known graph class of *chordal graphs*, which are the graphs that do not contain induced cycles of length 4 or more as induced subgraphs. It is known that split graphs are self-complementary, that is, the complement of a split graph remains a split graph. First we need the following two results.

Lemma 3.1. *Let $G = (V, E)$ be a split graph with a split partition (C, I). Let E_S be the set of strong edges in a solution for* MaxSTC *on G and let I_W be the vertices of I that are incident to at least one edge of E_S. Furthermore, for every vertex w_i of I we denote by A_i its strong neighbors in C and we denote by B_i the set of vertices in C that are non-adjacent to w_i. Then,*

$$|E_S| \leq |E(C)| + \sum_{w_i \in I_W} |A_i| \left(1 - \left\lfloor \frac{|B_i|}{2} \right\rfloor \right).$$

Proof. Let w_i be a vertex of I. For the edges of the clique, there are $|A_i||B_i|$ weak edges due to the strong triadic closure. Moreover any vertex w_j of $I \setminus \{w_i\}$ cannot have a strong neighbor in A_i. This means that $A_i \cap A_j = \emptyset$. Notice, however, that both sets $B_i \cap B_j$ and $A_i \cap B_j$ are not necessarily empty.

Observe that I_W contains the vertices of I that are incident to at least one strong edge. Let $E(A, B)$ be the set of weak edges that have one endpoint in A_i and the other endpoint in B_i, for every $1 \leq i \leq |I_W|$. We show that $2|E(A, B)| \geq \sum_{w_i \in I_W} |A_i||B_i|$. Let $\{a, b\} \in E(A, B)$ such that $a \in A_i$ and $b \in B_i$. Assume that there is a pair A_j, B_j such that $\{a, b\}$ is an edge between A_j and B_j, for $j \neq i$. Then a cannot belong to A_j since $A_i \cap A_j = \emptyset$. Thus, $a \in B_j$ and $b \in A_j$. Therefore, for every edge $\{a, b\} \in E(A, B)$ there are at most two pairs (A_i, B_i) and (A_j, B_j) for which $a \in A_i \cap B_j$ and $b \in B_i \cap A_j$. This means that every edge of $E(A, B)$ is counted at most twice in $\sum_{w_i \in I_W} |A_i||B_i|$.

For any two edges $\{u, v\}, \{v, z\} \in E(C) \setminus E(A, B)$, observe that they satisfy the strong triadic closure since there is the edge $\{u, z\}$ in G. Thus, the strong edges of the clique are exactly the set of edges $E(C) \setminus E(A, B)$. In total by counting the number of strong edges between the independent set and the clique, we have $|E_S| = |E(C) \setminus E(A, B)| + \sum_{w_i \in I_W} |A_i|$. Since $2|E(A, B)| \geq \sum_{w_i \in I_W} |A_i||B_i|$, we get the stated formula. □

Lemma 3.2. *Let $G = (V, E)$ be a split graph with a split partition (C, I). Let E_S be the set of strong edges in an optimal solution for* MaxSTC *on G and let I_W be the vertices of I that are incident to at least one edge of E_S.*

1. *If every vertex of I_W misses at least three vertices of C in G, then $E_S = E(C)$.*
2. *If every vertex of I_W misses exactly one vertex of C in G, then $|E_S| \leq |E(C)| + \lfloor \frac{|I_W|}{2} \rfloor$.*

Proof. We apply Lemma 3.1 in each case. The first claim of the lemma holds because $|B_i| \geq 3$ so that $I_W = \emptyset$. For the second claim we show that for every vertex of I_W, $|A_i| = 1$. Let $w_i \in I_W$ such that $|A_i| \geq 2$ and let $B_i = \{b_i\}$. Recall that no other vertex of I_W has strong neighbors in A_i. Also note that there is at most one vertex w_j in I_W that has b_i as a strong neighbor. If such a vertex w_j exists and for the vertex b_j of the clique that misses w_j it holds $b_j \in A_i$, then we let $v = b_j$; otherwise we choose v as an arbitrary vertex of A_i. Observe that no vertex of $I \setminus \{w_i\}$ has a strong neighbor in $A_i \setminus \{v\}$ and only $w_j \in I_W$ is strongly adjacent to b_i. Then we label weak the $|A_i| - 1$ edges between w_i and the vertices of $A_i \setminus \{v\}$ and we label strong the $|A_i| - 1$ edges between b_i and the vertices of $A_i \setminus \{v\}$. Making strong the edges between b_i and the vertices of $A_i \setminus \{v\}$ does not violate the strong triadic closure since every vertex of $C \cup \{w_j\}$ is adjacent to every vertex of $A_i \setminus \{v\}$. Therefore for every vertex $w_i \in I_W$, $|A_i| = 1$ and by substituting $|B_i| = 1$ in the formula for $|E_S|$ we get the claimed bound. □

In order to give the reduction, we introduce the following problem that we call *maximum disjoint non-neighborhood* (MaxDisjointNN):

> MaxDisjointNN
>
> *Input:* A split graph (C, I) where every vertex of I misses three vertices from C and a nonnegative integer k.
>
> *Task:* Find a subset S_I of I such that the non-neighborhoods of the vertices of S_I are pairwise disjoint and $|S_I| \geq k$.

The polynomial-time reduction to MaxDisjointNN is given from the classical NP-complete problem 3-Set Packing [18]: given a universe $\mathcal{U}$ of n elements, a family $\mathcal{F}$ of triplets of $\mathcal{U}$, and an integer k, the problem asks for a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ with $|\mathcal{F}'| \geq k$ such that all triplets of $\mathcal{F}'$ are pairwise disjoint.

Corollary 3.3. MaxDisjointNN *is NP-complete.*

Proof. Given a split graph $G = (C, I)$ and $S_I \subseteq I$, checking whether S_I is a solution for MaxDisjointNN amounts to checking whether every pair of vertices of S_I have common neighborhood. As this can be done in polynomial time the problem is in NP. We will give a polynomial-time reduction to MaxDisjointNN from the classical NP-complete problem 3-Set Packing [18]: given a universe $\mathcal{U}$ of n elements, a family $\mathcal{F}$ of triplets of $\mathcal{U}$, and an integer k, the problem asks for a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ with $|\mathcal{F}'| \geq k$ such that all triplets of $\mathcal{F}'$ are pairwise disjoint.

Let $(\mathcal{U}, \mathcal{F}, k)$ be an instance of the 3-Set Packing. We construct a split graph $G = (C, I)$ as follows. The clique of G is formed by the n elements of $\mathcal{U}$. For every triplet F_i of $\mathcal{F}$ we add a vertex v_i in I that is adjacent to every vertex of C except the three vertices that correspond to the triplet F_i. Thus every vertex of I misses exactly three vertices from C and sees the rest of C. Now it is not difficult to see that there is a solution $\mathcal{F}'$ for 3-Set Packing$(\mathcal{U}, \mathcal{F}, k)$ of size at least k if and only if there is a solution S_I for MaxDisjointNN(G, k) of size at least k. For every pair (F_i, F_j) of $\mathcal{F}'$ we know that $F_i \cap F_j = \emptyset$ which implies that the vertices v_i and v_j have disjoint non-neighborhood since F_i corresponds to the non-neighborhood of v_i. By the one-to-one mapping between the sets of $\mathcal{F}$ and the vertices of I, every set F_i belongs to $\mathcal{F}'$ if and only if v_i belongs to S_I. $\square$

Now we turn to our original problem MaxSTC. The decision version of MaxSTC takes as input a graph G and an integer k and asks whether there is strong-weak labeling of the edges of G that satisfies the strong triadic closure with at least k strong edges.

Theorem 3.4. *The decision version of* MaxSTC *is NP-complete on split graphs.*

Proof. Given a strong-weak labeling (E_S, E_W) of a split graph $G = (C, I)$, checking whether (E_S, E_W) satisfies the strong triadic closure amounts to check in $G \setminus E_W$ whether there is a non-edge in G between the endvertices of every P_3 according to Observation 2.1. Thus by listing all P_3's of $G \setminus E_W$ the problem belongs to NP. Next we give a polynomial-time reduction to MaxSTC from the MaxDisjointNN problem on split graphs which is NP-complete by Corollary 3.3. Let (G, k) be an instance of MaxDisjointNN where $G = (C, I)$ is a split graph such that every vertex of the independent set I misses exactly three vertices from the clique C. For a vertex $w_i \in I$, we denote by B_i the set of the three vertices in C that are non-adjacent to w_i. Let $n = |C|$. We extend G and construct another split graph G' as follows (see Figure 3):

- We add n vertices $y_1, \ldots, y_n$ in the clique that constitute the set C_Y.
- We add n vertices $x_1, \ldots, x_n$ in the independent set that constitute the set I_X.
- For every $1 \leq i \leq n$, y_i is adjacent to all vertices of $(C \cup C_Y \cup I \cup I_X) \setminus \{x_i\}$.
- For every $1 \leq i \leq n$, x_i is adjacent to all vertices of $(C \cup C_Y) \setminus \{y_i\}$.

Thus w_i misses only the vertices of B_i from the clique. By construction it is clear that G' is a split graph with a split partition $(C \cup C_Y, I \cup I_X)$. Notice that the clique $C \cup C_Y$ has $2n$ vertices and $G = G'[I \cup C]$.

We next prove that G has a solution for MaxDisjointNN of size at least k if and only if G' has a strong triadic closure with at least $n(2n-1) + \lfloor \frac{n}{2} \rfloor + \lceil \frac{k}{2} \rceil$ strong edges.
($\Rightarrow$) Assume that $\{w_1, \ldots, w_k\} \subseteq I$ is a solution for MaxDisjointNN on G of size at least k. Since the sets $B_1, \ldots, B_k$ are pairwise disjoint, there are k distinct vertices $y_1, \ldots, y_k$ in C_Y such that $k \leq n$. We will give a strong-weak labeling for the edges of G' that fulfills the strong triadic closure and has at least the claimed number of strong edges. For simplicity,

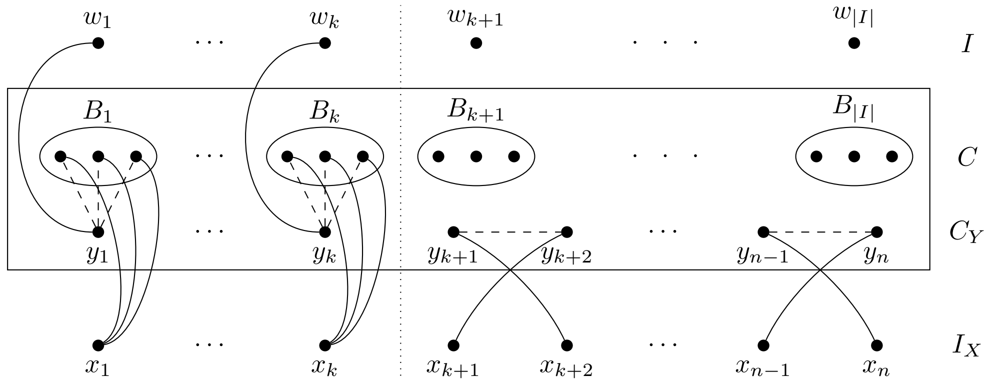

Figure 3: The split graph $(C \cup C_Y, I \cup I_X)$ given in the polynomial-time reduction. Every vertex w_i misses the vertices of B_i and sees the vertices of $(C \cup C_Y) \setminus B_i$. Every vertex x_i misses y_i and sees the vertices of $(C \cup C_Y) \setminus \{y_i\}$. The sets $B_1, \ldots, B_k$ are pairwise disjoint whereas for every set B_j, $k < j \leq |I|$, there is a set B_i, $1 \leq i \leq k$, such that $B_i \cap B_j \neq \emptyset$. The drawn edges correspond to the strong edges between the independent set and the clique, and the dashed edges are the only weak edges in the clique $C \cup C_Y$.

we describe only the strong edges; the edges of G' that are not given are all labeled weak. We label the edges incident to each vertex w_i, y_i, x_i and the three vertices of each set B_i, for $1 \leq i \leq k$ as follows:

- The edges of the form $\{y_i, v\}$ are labeled strong if $v \in (C \cup C_Y) \setminus B_i$ or $v = w_i$.
- The edges between x_i and the three vertices of B_i are labeled strong.

Next we label the edges incident to the rest of the vertices. No edge incident to a vertex of $I \setminus \{w_1, \ldots, w_k\}$ is labeled strong. For every vertex $u \in C \setminus (B_1 \cup \cdots \cup B_k)$ we label the edge $\{u, v\}$ strong if $v \in (C \cup C_Y)$. Let $C'_Y = \{y_{k+1}, \ldots, y_n\}$ and let $I'_X = \{x_{k+1}, \ldots, x_n\}$. Recall that every vertex x_{k+j} is adjacent to every vertex of $C'_Y \setminus \{y_{k+j}\}$. Let $\ell = \lfloor \frac{n-k}{2} \rfloor$. Let $M = \{e_1, \ldots, e_\ell\}$ be a maximal set of pairwise non-adjacent edges in $G'[C'_Y]$ where $e_j = \{y_{k+2j-1}, y_{k+2j}\}$, for $j \in \{1, \ldots, \ell\}$; note that M is a maximal matching of $G'[C'_Y]$. For every vertex $y \in C'_Y$, we label the edge $\{y, v\}$ strong if $v \in (C \cup C_Y) \setminus \{y'\}$ such that $\{y, y'\} \in M$. Moreover, for $j \in \{1, \ldots, \ell\}$, the edges $\{x_{k+2j-1}, y_{k+2j}\}$ and $\{x_{k+2j}, y_{k+2j-1}\}$ are labeled strong. Note that if $n - k$ is odd then no edge incident to the unique vertex y_n belongs to M and all edges between y_n and the vertices of $C \cup C_Y$ are labeled strong; in such a case also note that no edge incident to x_n is strong.

Let us show that such a labeling fulfills the strong triadic closure. Any labeling for the edges inside $G'[C \cup C_Y]$ is satisfied since $G'[C \cup C_Y]$ is a clique. Also note that there are no two adjacent strong edges that have a common endpoint in the clique $C \cup C_Y$ and the two other endpoints in the independent set $I \cup I_X$. If there are two strong edges incident to the same vertex v of the independent set then $v \in \{x_1, \ldots, x_k\}$ and $N_S[v] = B_i$ which is a clique. Assume that there are two adjacent strong edges $\{u, v\}$ and $\{v, z\}$ such that $u \in I \cup I_X$, and $v, z \in C \cup C_Y$.

- If $u \in \{w_1, \ldots, w_k\}$ then $\{u, z\} \in E(G')$ since every w_i misses only the vertices of B_i.
- If $u \in \{x_1, \ldots, x_k\}$ then $v \in B_i$ and $\{u, z\} \in E(G')$ since every vertex x_i misses only y_i.
- If $u \in I_X \setminus \{x_1, \ldots, x_k\}$ then the strong neighbors of v in $C \cup C_Y$ are adjacent to u in G' since for the only non-neighbor of u in $C \cup C_Y$ there is a weak edge incident to v.

Recall that there is no strong edge incident to the vertices of $I \setminus \{w_1, \ldots, w_k\}$. Therefore the given strong-weak labeling fulfills the strong triadic closure.

Observe that the number of vertices in $C \cup C_Y$ is $2n$. There are exactly $3k + \ell$ weak edges in $G'[C \cup C_Y]$. Thus the number of strong edges in $G'[C \cup C_Y]$ is $n(2n-1) - 3k - \ell$. There are k strong edges incident to $\{w_1, \ldots, w_k\}$, $3k$ strong edges incident to $\{x_1, \ldots, x_k\}$, and 2ℓ strong edges incident to $I_X \setminus \{x_1, \ldots, x_k\}$. Thus the total number of strong edges is $n(2n-1) - 3k - \ell + k + 3k + 2\ell = n(2n-1) + \ell + k$ and by substituting $\ell = \lfloor \frac{n-k}{2} \rfloor$ we get the claimed bound.

($\Leftarrow$) For the opposite direction, assume that G' has a strong triadic closure with at least $n(2n-1) + \lfloor \frac{n}{2} \rfloor + \lceil \frac{k}{2} \rceil$ strong edges. Let E_S be the set of strong edges in such a strong-weak labeling. Observe that the number of edges in $G'[C \cup C_Y]$ is $n(2n-1)$ which implies that E_S contains edges between the independent set $I \cup I_X$ and the clique $C \cup C_Y$. If no vertex of I_X is incident to an edge of E_S then the first statement of Lemma 3.2 implies that $|E_S| = |E(C \cup C_Y)| = n(2n-1)$. And if no vertex of I is incident to an edge of E_S then the second statement of Lemma 3.2 shows that $|E_S| \leq |E(C \cup C_Y)| + \lfloor \frac{n}{2} \rfloor$. Therefore, E_S contains edges that are incident to a vertex of I and edges that are incident to a vertex of I_X.

In the graph spanned by E_S we denote by S_W the set of vertices of I that have strong neighbors in $C \cup C_Y$. A *nice solution* is a strong-weak labeling that satisfies the strong triadic closure with a set of strong edges E'_S such that $|E_S| = |E'_S|$. We will show that the non-neighborhoods of the vertices of S_W in $C \cup C_Y$ are disjoint in G' and, since G is an induced subgraph of G', their non-neighborhoods are also disjoint in G.

Claim 3.5. *There exists a nice solution such that for every $w_i \in S_W$, (i) $N_S(w_i) \subseteq C_Y$ holds and (ii) there is a unique vertex $x \in I_X$ with $N_S(x) = B_i$.*

Proof: Let w_i be a vertex of S_W. We first show that $N_S(w_i) \subseteq C_Y$. Let W_i be the strong neighbors of w_i in C and let Y_i be the strong neighbors of w_i in C_Y. Observe that no other vertex of S_W has a strong neighbor in $W_i \cup Y_i$. Furthermore, notice that there are $(|W_i| + |Y_i|)|B_i|$ weak edges since w_i is non-adjacent to the vertices of B_i. We show that $W_i = \emptyset$, for every vertex $w_i \in S_W$. For all vertices w_i for which $W_i \neq \emptyset$ we replace in E_S the strong edges between w_i and the vertices of W_i by the edges between the vertices of B_i and W_i. Notice that making strong the edges between the vertices of B_i and W_i does not violate the strong triadic closure since no vertex from S_W has a strong neighbor in B_i and every vertex of I_X is adjacent to all the vertices of W_i. Let $E(W, B)$ be the set of edges that have one endpoint in W_i and the other endpoint in B_i, for every $w_i \in S_W$. Notice that the difference between the two described solutions is $|E(W, B)| - \sum |W_i|$. By Lemma 3.1 and $|B_i| = 3$, we know that $|E(W, B)| \geq 3/2 \sum |W_i|$. Thus, such a replacement is safe for the number of edges of E_S and every vertex $w_i \in S_W$ has strong neighbors only in C_Y.

Let X_i be the set of vertices of I_X that have at least one non-neighbor in Y_i. By construction every vertex of Y_i is non-adjacent to exactly one vertex of I_X, and thus $|X_i| = |Y_i|$. Since w_i has strong neighbors in Y_i, every edge between X_i and Y_i is weak. By the previous argument every vertex of S_W has strong neighbors only in C_Y so that $N_S(B_i) \cap I = \emptyset$. Also notice that no two vertices of the independent set have a common strong neighbor in the clique, which means that there are at most $|B_i|$ strong neighbors between the vertices of B_i and I_X. Choose an arbitrary vertex $x \in X_i$. We replace all strong edges in E_S between B_i and I_X by $|B_i|$ strong edges between x and the vertices of B_i. Notice that such a replacement is safe since the unique non-neighbor of x belongs to Y_i and there are weak edges already in the solution between B_i and Y_i because of the strong edges between w_i and Y_i. Thus $B_i \subseteq N_S(x)$. We focus on the edges between the vertices of $(C \cup C_Y) \setminus (B_i \cup Y_i)$ and x. If a vertex x of X_i has a strong neighbor u in $(C \cup C_Y) \setminus B_i$ then the edge $\{u, y\}$ is weak where $y \in Y_i$ is the unique non-neighbor of x. Also notice that $N_S(u) \cap (I \cup I_X) = \{x\}$, $N_S(y) \cap (I \cup I_X) = \{w_i\}$, and w_i is adjacent to u. Then we can safely replace the strong edge $\{x, u\}$ by the edge $\{u, y\}$ and keep the same size of E_S. Hence $N_S(x) = B_i$. $\diamondsuit$

Claim 3.6. *There exists a nice solution such that for every $w_i \in S_W$, $N_S(w_i) = \{y\}$ where $y \in C_Y$ is the non-neighbor of x with $N_S(x) = B_i$.*

Proof: Let $Y_i = N_S(w_i)$. By Claim 3.5 we know that $Y_i \subseteq C_Y$ and there exists $x \in I_X$ such that $N_S(x) = B_i$. Both w_i and x are vertices of the independent set and, thus, no other vertex of $I \cup I_X$ has strong neighbors in $B_i \cup Y_i$. This means that if we remove w_i from S_W by making weak the edges incident to w_i and the vertices of Y_i then the edges between the vertices of B_i and $Y_i \setminus \{y\}$ are safely turned into strong. Let E'_S be the set of strong edges in a nice solution such that all edges incident to w_i are weak. Then $|E_S| - |E'_S| = |Y_i| + |B_i| - |Y_i||B_i|$ and $|E_S| > |E'_S|$ only if $|Y_i| = 1$ because $|B_i| > 1$. Thus $N_S(w_i)$ contains exactly one vertex $y \in C_Y$. $\diamondsuit$

We claim that for every pair of vertices $w_i, w_j \in S_W$, $B_i \cap B_j = \emptyset$. Assume for contradiction that $B_i \cap B_j \neq \emptyset$. Applying Claim 3.5 for w_i shows that there exists $x \in I_X$ that has strong neighbors in every vertex of $B_i \cap B_j$. With a similar argument for w_j we deduce that there exists $x' \in I_X$ that has strong neighbors in every vertex of $B_i \cap B_j$. If $x \neq x'$ then a vertex from $B_i \cap B_j$ has two distinct strong neighbors in I_X which is not possible due to the strong triadic closure. Thus $x = x'$. Claim 3.6 implies that the unique non-neighbor y of x is strongly adjacent to both w_i and w_j. This however violates the strong triadic closure for the edges of E_S since w_i, w_j are non-adjacent and we reach a contradiction. Thus $B_i \cap B_j = \emptyset$. This means that the number of edges in E_S is at least $n(2n-1) + \lfloor \frac{n}{2} \rfloor + \lceil \frac{|S_W|}{2} \rceil$ which is maximized for $k = |S_W|$. Therefore there are at least $|S_W|$ vertices from the independent set which form a solution for MaxDisjointNN on G, since G is an induced subgraph of G'. $\square$

4 Computing MaxSTC on proper interval graphs

Due to the NP-completeness on split graphs given in Theorem 3.4, it is natural to consider interval graphs that form another well-studied subclass of chordal graphs. However besides few observations of this section that may be applied for interval graphs, we found several unresolved technical cases. Moreover we would like to note that, to the best of our knowledge, the complexity of the close-related Cluster Deletion problem still remains unresolved on interval graphs [3]. Therefore we further restrict the input to the class of proper interval graphs that form a proper subclass of interval graphs. Our polynomial solution for MaxSTC on proper interval graphs can be seen as a first step towards determining its complexity on interval graphs.

A graph is a *proper interval graph* if there is a bijection between its vertices and a family of closed intervals of the real line such that two vertices are adjacent if and only if the two corresponding intervals overlap and no interval is properly contained in another interval. A vertex ordering σ is a linear arrangement $\sigma = \langle v_1, \ldots, v_n \rangle$ of the vertices of G. For a vertex pair x, y we write $x \preceq y$ if $x = v_i$ and $y = v_j$ for some indices $i \leq j$; if $x \neq y$ which implies $i < j$ then we write $x \prec y$. We extend $\prec$ to support sets of vertices, as follows: for two sets of vertices A and B we write $A \prec B$ if for any $a \in A$ and $b \in B$, $a \prec b$ holds. The first position in σ will be referred to as the *left end* of σ, and the last position as the *right end*. We will use the expressions *to the left of*, *to the right of*, *leftmost*, and *rightmost* accordingly.

A vertex ordering σ for G is called a *proper interval ordering* if for every vertex triple x, y, z of G with $x \prec y \prec z$, $\{x, z\} \in E(G)$ implies $\{x, y\}, \{y, z\} \in E(G)$. Proper interval graphs are characterized as the graphs that admit such orderings, that is, a graph is a proper interval graph if and only if it has a proper interval ordering [25]. We only consider this vertex

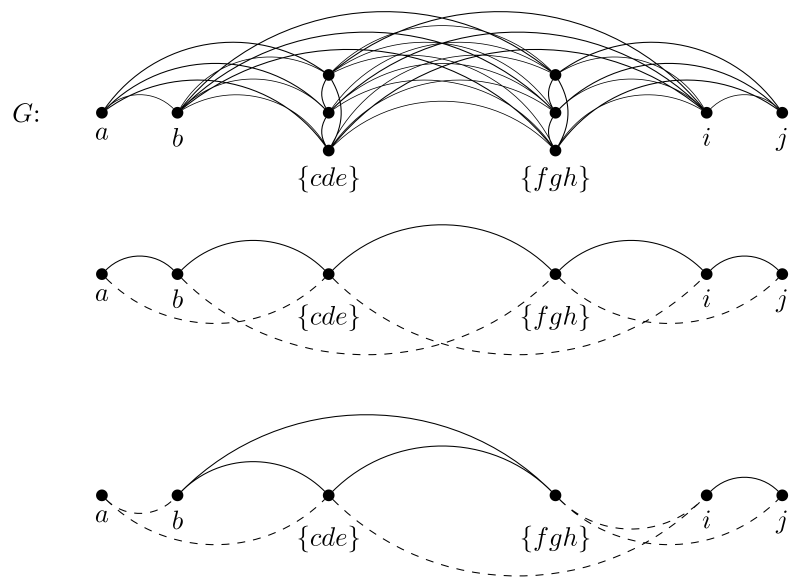

Figure 4: A proper interval graph G and its proper interval ordering. The vertices $\{c, d, e\}$ and $\{f, g, h\}$ form twin sets in G. The two lower orderings depict two solutions for MaxSTC on G. A solid edge corresponds to a strong edge, whereas a dashed edge corresponds to a weak edge. Observe that the upper solution contains larger number of strong edges than the lower one. Also note that the lower solution consists an optimal solution for the Cluster Deletion problem on G.

ordering characterization for proper interval graphs. Moreover it can be decided in linear time whether a given graph is a proper interval graph, and if so, a proper interval ordering can be generated in linear time [25]. It is clear that a vertex ordering σ for G is a proper interval ordering if and only if the reverse of σ is a proper interval ordering. A connected proper interval graph without twin vertices has a unique proper interval ordering σ up to reversal [8, 16]. Figure 4 shows a proper interval graph with its proper interval ordering.

Before reaching the details of our algorithm for proper interval graphs, let us highlight the difference between the optimal solution for MaxSTC and the optimal solution for the Cluster Deletion. As already explained in the Introduction a solution for Cluster Deletion satisfies the strong triadic closure, though the converse is not necessarily true. In fact such an observation carries out for the class of proper interval graphs as shown in the example given in Figure 4. For the Cluster Deletion problem twin vertices can be grouped together following a similar characterization with Lemma 2.4, as proved in [3]. This means that the P_3-free graph depicted in the lower part of Figure 4 that is obtained by removing its weak edges (i.e., the dashed drawn lines) is an optimal solution for Cluster Deletion problem on the given proper interval graph. Therefore when restricted to proper interval graphs the optimal solution for Cluster Deletion does not necessarily imply an optimal solution for MaxSTC.

Let G be a proper interval graph and let σ be a proper interval ordering for G. Hereafter we denote by E_S the set of strong edges in an optimal solution for MaxSTC on G. Moreover a labeled edge $\{x, y\}$ (weak or strong) is simple denoted by xy. By Proposition 2.2 and the strong triadic closure, in the forthcoming arguments we will apply the following observations:

- If $xy \in E_S$, then for every strong edge $yz \in E_S$, $\{x, z\} \in E(G)$.
- If $xy \notin E_S$, then there is a strong edge $yz \in E_S$ such that $\{x, z\} \notin E(G)$.

We say that a solution E_S has the *consecutive strong property* with respect to σ if for any three vertices x, y, z of G with $x \prec y \prec z$ the following holds: $xz \in E_S$ implies $xy, yz \in E_S$. Our task is to show that such an ordering exists for an optimal solution. We start by characterizing the optimal solution E_S with respect to the proper interval ordering σ.

Lemma 4.1. *Let x, y, z be three vertices of a proper interval graph G such that $x \prec y \prec z$. If $xz \in E_S$ then $xy \in E_S$ or $yz \in E_S$.*

Proof. We show that at least one of xy or yz belongs to E_S. Assume towards a contradiction that neither xy nor yz belong to E_S. Consider the edge xy in G. Since $xy \notin E_S$, there is a strong edge of E_S that share a common vertex with xy. Suppose that there is an edge $xx_\ell \in E_S$ such that $\{x_\ell, y\} \notin E(G)$. Then observe that $x_\ell \prec y$ because $x \prec y$ and $\{x_\ell, y\} \notin E(G)$. Since both xx_ℓ and xz belong to E_S, $\{x_\ell, z\} \in E(G)$. This, however, contradicts the proper interval ordering because $x_\ell \prec y \prec z$, $\{x_\ell, z\} \in E(G)$ and y is non-adjacent to x_ℓ. Thus, there is no strong edge xx_ℓ with $\{x_\ell, y\} \notin E(G)$ and, in analogous fashion, there is no strong edge zz_r with $\{y, z_r\} \notin E(G)$.

Now, if all strong neighbors of y are adjacent to x (resp., z) then the edge xy (resp., yz) can be made strong. So, let us assume that there are edges $yy_r \in E_S$ and $y_\ell y \in E_S$ such that $\{x, y_r\} \notin E(G)$ and $\{z, y_\ell\} \notin E(G)$. Since $\{x, z\} \in E(G)$, y_ℓ and y_r cannot be ordered between x and y in the proper interval ordering. In particular, by the facts $\{x, y_r\} \notin E(G)$ and $\{z, y_\ell\} \notin E(G)$ we have $y_\ell \prec x \prec y \prec z \prec y_r$. Then, notice that $\{y_\ell, y_r\} \in E(G)$, because both $yy_r, yy_\ell \in E_S$. By the proper interval ordering we know that both x and z are adjacent to y_ℓ, y_r, leading to a contradiction to the assumptions $\{x, y_r\} \notin E(G)$ and $\{z, y_\ell\} \notin E(G)$. Therefore, at least one of xy or yz belongs to E_S. □

Thus, by Lemma 4.1 we have two symmetric cases to consider. The next characterization suggests that there is a fourth vertex with important properties in each corresponding case.

Lemma 4.2. *Let x, y, z be three vertices of a proper interval graph G such that $x \prec y \prec z$ and $xz \in E_S$.*

- *If $xy \notin E_S$ and $yz \in E_S$ then for every edge $x_\ell x \in E_S$, $\{x_\ell, y\} \in E(G)$ holds and there is a vertex w such that $yw \in E_S$, $\{x, w\} \notin E(G)$, and $z \prec w$.*
- *If $xy \in E_S$ and $yz \notin E_S$ then for every edge $zz_r \in E_S$, $\{y, z_r\} \in E(G)$ holds and there is a vertex w such that $wy \in E_S$, $\{w, z\} \notin E(G)$ and $w \prec x$.*

Proof. Let $xy \notin E_S$ and $yz \in E_S$. The case for $xy \in E_S$ and $yz \notin E_S$ is completely symmetric. First, we prove that there is no strong edge $x_\ell x$ such that $\{x_\ell, y\} \notin E(G)$. Suppose that there is an edge $x_\ell x \in E_S$ with y being non-adjacent to x_ℓ. Notice that $x_\ell \prec x$ because y is adjacent to x and $x \prec y$. Due to the fact that $x_\ell x, xz \in E_S$, we have $\{x_\ell, z\} \in E(G)$. Since $x_\ell \prec x \prec y \prec z$ and $\{x_\ell, z\} \in E(G)$, by the proper interval ordering we get $\{x_\ell, y\} \in E(G)$ leading to a contradiction. Thus there is no strong edge $x_\ell x \in E_S$ with $\{x_\ell, y\} \notin E(G)$.

Now to prove the statement, assume towards a contradiction that there is no vertex w such that $yw \in E_S$, $\{x, w\} \notin E(G)$, and $z \prec w$. Thus, for all edges $yy_r \in E_S$ either $\{x, y_r\} \notin E(G)$ and $y_r \prec z$, or $\{x, y_r\} \in E(G)$. If $\{x, y_r\} \notin E(G)$ we know that $y_r \prec x$ or $x \prec y_r \prec z$ by the proper interval ordering. However, both cases lead to a contradiction to $\{x, y_r\} \notin E(G)$ since in the former case we have $\{y_r, y\} \in E(G)$ and $y_r \prec x \prec y$, and in the latter case we know that $\{x, z\} \in E(G)$. If $\{x, y_r\} \in E(G)$ then putting xy in E_S does not violate the strong triadic closure, contradicting the assumption $xy \notin E_S$ of the statement. Therefore, the corresponding statement holds. □

Now we are ready to show that there is an optimal solution that has the described properties with respect to the given proper interval ordering.

Lemma 4.3. *There exists an optimal solution E_S that has the consecutive strong property with respect to σ.*

Proof. Let σ be a proper interval ordering for G. Assume towards a contradiction that E_S does not have the consecutive strong property. Then there exists a *conflict* with respect to σ, that is, there are three vertices x, y, z with $x \prec y \prec z$ and $xz \in E_S$ such that $xy \notin E_S$ or $yz \notin E_S$. We will show that as long as there are conflicts in σ, we can reduce the number of conflicts in σ without affecting the value of the optimal solution E_S. Consider such a conflict formed by the three vertices $x \prec y \prec z$ with $xz \in E_S$. By Lemma 4.1 we know that $xy \in E_S$ or $yz \in E_S$. Without loss of generality, assume that $yz \in E_S$. Then, clearly $xy \notin E_S$, for otherwise there is no conflict. Moreover, by Lemma 4.2 there is a vertex w such that $yw \in E_S$, $\{x, w\} \notin E(G)$, and $x \prec y \prec z \prec w$. Notice that both triples x, y, z and y, z, w create conflicts in σ.

We start by choosing an appropriate such conflict that is formed by four vertices x, y, z, w so that $x \prec y \prec z \prec w$, $xz, yz, yw \in E_S$, and $\{x, w\} \notin E(G)$. Fix y and z in σ with y, z being the leftmost and the rightmost vertices, respectively, such that for every vertex v with $y \prec v \prec z$, $yv, vz \in E_S$ holds. Recall that $yz \in E_S$. We choose x as the leftmost vertex such that $xz \in E_S$, $xy \notin E_S$ and we choose w as the rightmost vertex such that $yw \in E_S$, $zw \notin E_S$. Observe that $\{x, w\} \notin E(G)$, since y and z participate in a conflict. Due to the properties of the considered conflicts, all such vertices exist (see for e.g., Figure 5). Our task is to create a new set of strong edges E'_S from E_S such that (i) E'_S does not violate the strong triadic closure, (ii) $|E'_S| = |E_S|$, and (iii) E'_S has strictly less conflicts than E_S. We do so, with a series of claims that we prove next.

We define the following sets of vertices with respect to the strong neighbors of z and y:

- Let $X(z)$ be the set of vertices x_j such that $x_j z \in E_S$ and $\{x_j, w\} \notin E(G)$.
- Let $W(y)$ be the set of vertices w_i such that $yw_i \in E_S$ and $\{x, w_i\} \notin E(G)$.

Claim 4.4. *$X(z) \prec y \prec z \prec W(y)$ holds. Moreover, for any vertex of $x_j \in X(z)$, $x_j y \notin E_S$ holds, and for any vertex of $w_i \in W(y)$, $zw_i \notin E_S$ holds.*

Proof: Let $x_j \in X(z)$. If $w \prec x_j$ then $\{x_j, w\} \in E(G)$ because $z \prec w$ and $\{x_j, z\} \in E(G)$, and if $y \prec x_j \prec w$ then $\{x_j, w\} \in E(G)$ because $\{y, w\} \in E(G)$. By the definition of $X(z)$ we know that $\{x_j, w\} \notin E(G)$ which means $x_j \prec y$. Thus, $\{x_j, y\} \in E(G)$ holds, since $\{x_j, z\} \in E(G)$. With symmetric arguments for the vertices of $W(y)$, we conclude that $X(z) \prec y \prec z \prec W(y)$. Moreover, if $x_j y \in E_S$ then by the fact that $yw \in E_S$, we have $\{x_j, w\} \in E(G)$, contradicting the definition of $X(z)$. $\diamondsuit$

Notice that the previous claim and the choices for x and w imply that x is the leftmost vertex of $X(z)$ and w is the rightmost vertex of $W(y)$. We further partition the strong neighbors of z that lie to the left of y, and the strong neighbors of y that lie to the right of z, as follows (see Figure 5):

- Let $A(z)$ be the set of vertices a_j such that $a_j \prec y$, $a_j z \in E_S$, $\{a_j, w\} \in E(G)$, and $a_j y \notin E_S$.
- Let $B(y)$ be the set of vertices b_i such that $z \prec b_i$, $yb_i \in E_S$, $\{x, b_i\} \in E(G)$, and $zb_i \notin E_S$.

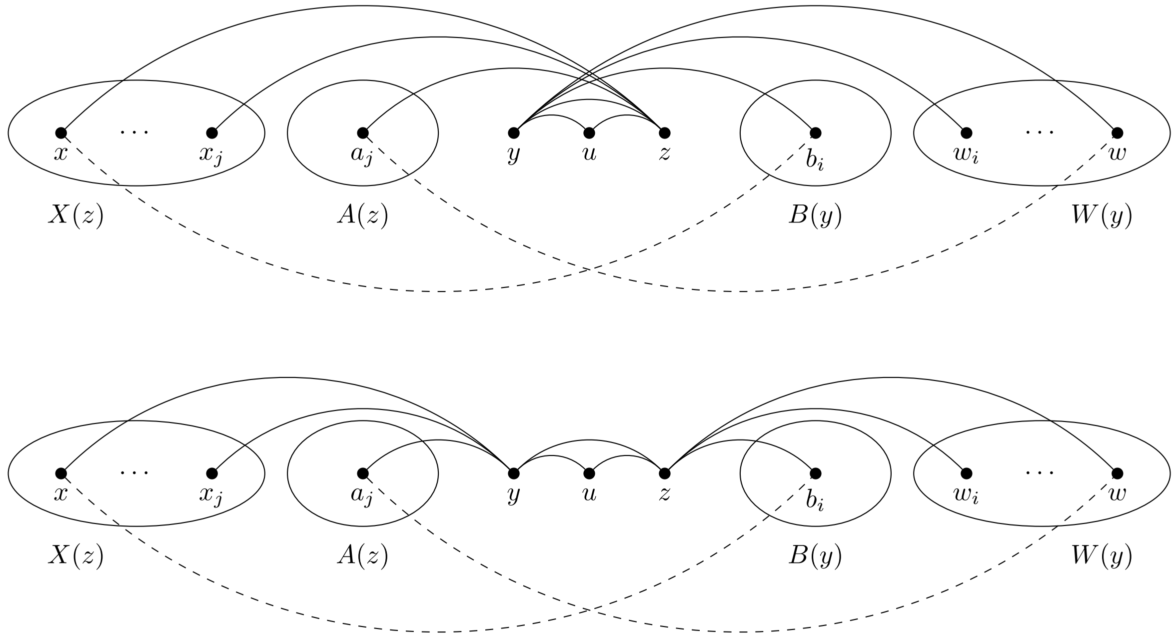

Figure 5: A proper interval ordering for a graph G with two different solutions considered in the proof of Lemma 4.3. A solid edge corresponds to a strong edge in an optimal strong-weak labeling, whereas a dashed edge corresponds to a weak edge in an optimal strong-weak labeling. Observe that the lowest ordering contains less *conflicts* than the topmost, that is, triple of vertices that violate the consecutive strong property.

Claim 4.5. *$X(z) \prec A(z) \prec y \prec z \prec B(y) \prec W(y)$ holds. Moreover, every vertex of $A(z)$ is adjacent to all the vertices of $W(y)$ and every vertex of $B(y)$ is adjacent to all the vertices of $X(z)$.*

Proof: By Claim 4.4 and the definition of $A(z)$, we know that $X(z) \prec y$ and $A(z) \prec y$. Let $x_j \in X(z)$ and $a_j \in A(z)$. Since $\{x_j, w\} \notin E(G)$, $\{a_j, w\} \in E(G)$, and σ is a proper interval ordering, we deduce that $X(z) \prec A(z) \prec y$. Moreover, a_j is adjacent to w by definition, and w is the rightmost vertex of $W(y)$. Thus, a_j is adjacent to every vertex of $W(y)$. Symmetric arguments for the vertices of $B(y)$ and $W(y)$ show the rest of the claims. $\diamondsuit$

Claim 4.6. *Let v be a vertex of G. For any vertex $x_j \in X(z)$, we have $vx_j \in E_S$ implies $\{v, y\} \in E(G)$. Moreover, for any vertex $a_j \in A(z)$, we have $va_j \in E_S$ implies $\{v, y\} \in E(G)$.*

Proof: For the first claim, notice that $v \prec w$, since $\{x_j, w\} \notin E(G)$. If $x_j \prec v \prec w$ then $\{v, y\} \in E(G)$, since $x_j \prec y \prec w$ and y is adjacent to x_j and w; and if $v \prec x_j$ then, due to the fact that $x_j z, vx_j \in E_S$ and $\{z, v\} \in E(G)$, we have again $\{v, y\} \in E(G)$, because $v \prec y \prec z$. For the latter, if $y \prec v$ then $\{v, y\} \in E(G)$ because $a_j \prec y \prec v$ and $\{a_j, v\} \in E(G)$; and if $v \prec y$ then $\{v, z\} \in E(G)$ because $a_j z, va_j \in E_S$, so that $\{v, y\} \in E(G)$ by the proper interval ordering $v \prec y \prec z$. $\diamondsuit$

With symmetric arguments, we get the following for the vertices of $W(y)$ and $B(y)$:

Claim 4.7. *Let v be a vertex of G. For any vertex $w_i \in W(y)$, we have $vw_i \in E_S$ implies $\{v, z\} \in E(G)$. Moreover, for any vertex $b_i \in B(y)$, we have $vb_i \in E_S$ implies $\{v, z\} \in E(G)$.*

The next claim shows that for every vertex v that is strongly adjacent to y and lies to the right of y, either v belongs to $B(y) \cup W(y)$ or v is strongly adjacent to z, as well.

Claim 4.8. *Let $yv \in E_S$ such that $y \prec v$ and $v \notin B(y) \cup W(y)$. Then, $zv \in E_S$.*

Proof: If $y \prec v \prec z$ then according to the choice of y and z we have $yv, zv \in E_S$. Assume for contradiction that $z \prec v$ and $zv \notin E_S$. Then, v is either adjacent to x or non-adjacent to x. In the former case we have $v \in B(y)$ and in the latter case we have $v \in W(y)$. Thus, in all cases we get $zv \in E_S$. $\diamondsuit$

Symmetrically, for the left strong neighbors of z we have the following:

Claim 4.9. *Let $zv \in E_S$ such that $v \prec z$ and $v \notin X(z) \cup A(z)$. Then, $yv \in E_S$.*

We define the following set of weak and strong edges:

- Let $S(y)$ be the set of weak edges of the form yv where $v \in X(z) \cup A(z)$.
- Let $E(y)$ be the set of strong edges of the form yv where $v \in B(y) \cup W(y)$.
- Let $S(z)$ be the set of weak edges of the form zv where $v \in B(y) \cup W(y)$.
- Let $E(z)$ be the set of strong edges of the form zv where $v \in X(z) \cup A(z)$.

Now we are ready to modify E_S as follows:

$$E'_S = (E_S \setminus (E(y) \cup E(z))) \cup (S(y) \cup S(z)) \,.$$

We say that a labeled edge e is *removed* from E_S if $e \in E_S \setminus E'_S$, whereas e is *added* in E'_S if $e \in E'_S \setminus E_S$. Under these terms, observe that an edge uy is added only if $u \prec y$, and uy is removed only if $y \prec u$; symmetrically, uz is added only if $z \prec u$, and uz is removed only if $u \prec z$.

Claim 4.10. *Labeling strong the edges of E'_S does not violate the strong triadic closure.*

Proof: Observe that we only replace labeled edges incident to y and z. Consider the edges incident to y. Those new strong edges are formed by the set $S(y)$. Since E_S satisfies the strong triadic closure, the edges of $E_S \setminus E(y)$ do not violate the strong triadic closure. Let $uy \in E'_S$. By the definition of $S(y)$, we know that $u \in X(z) \cup A(z)$ and $u \prec y$. There are three cases to consider:

(i) there is an edge $vy \in E'_S$ and $v \in X(z) \cup A(z)$.

(ii) there is an edge $vy \in E'_S$ and $v \notin X(z) \cup A(z)$.

(iii) there is an edge $uv \in E'_S$.

In the former case (i), observe that $uz, vz \in E_S$ according to the definition of $X(z)$ and $A(z)$. Thus, $\{u, v\} \in E(G)$ holds. For the second case (ii), if $y \prec v$ then $vz \in E_S$ by Claim 4.8, which means that $\{u, v\} \in E(G)$ holds, since $uz \in E_S$. If $v \prec y$ then $x \prec v$ because $yw, vy \in E_S$ and $\{x, w\} \notin E(G)$. Thus, $x \prec \{u, v\} \prec y \prec z$ and, since $\{x, z\} \in E(G)$, we get $\{u, v\} \in E(G)$. For the third case (iii), Claim 4.6 shows that $\{y, v\} \in E(G)$. Therefore, the edges incident to y in E'_S do not violate the strong triadic closure.

Completely symmetric arguments hold for the edges incident to z in E'_S. Hence, labeling strong the edges of E'_S does not violate the strong triadic closure. $\diamondsuit$

Claim 4.11. *$|E'_S| = |E_S|$ holds.*

Proof: Only edges incident to y and z are modified from E_S with respect to E'_S. By definition, the number of edges incident to y that are removed from E_S is $|B(y)| + |W(y)|$, whereas the number of edges incident to z that are added in E'_S is $|B(y)| + |W(y)|$. Symmetrically, the number of edges incident to z that are removed from E_S is $|X(z)| + |A(z)|$, whereas the number of edges incident to y that are added in E'_S is $|X(z)| + |A(z)|$. Since the rest of the edges in E_S and E'_S remain the same, we have $|E'_S| = |E_S|$. $\diamondsuit$

Claim 4.12. *The number of conflicts in E'_S is strictly smaller than the number of conflicts in E_S.*

Proof: Observe that conflicts of the form $X(z) \prec y \prec z$ and $y \prec z \prec W(y)$ in E_S do not create conflicts in E'_S by the construction of E'_S. Since $X(z)$ and $W(y)$ are non-empty sets, such already existed conflicts in E_S do not appear in E'_S. We show that for every other conflict in E'_S, there is a unique conflict in E_S. First we show that both y and z do not participate in a conflict in E'_S. Assume for contradiction that y and z form a conflict in E'_S with another vertex v. If $v \prec y \prec z$ then, since $yz \in E_S \cap E'_S$, we have $vy \notin E'_S$ and $vz \in E'_S$ in order to create a conflict. Then, observe that $vz \in E_S$, because no added edge is incident to the left of z in E'_S. Moreover, no edge incident to the left of y is removed from E_S which means that $vy \notin E_S$. This, however, contradicts Claim 4.9, since $vz \in E_S$, $v \notin X(z) \cup A(z)$, and $vy \notin E_S$. If $y \prec z \prec v$ then symmetric arguments show that there is no such conflict. Moreover, if $y \prec v \prec z$ holds, then by the choice of y and z, we know that $yv, vz \in E_S \cap E'_S$. Thus, there is no conflict in E'_S that is formed by y and z.

Consider the edges incident to y. Note that with respect to the proper interval ordering, we only add edges incident to y and a vertex to the left of y, and we only remove edges incident to y and a vertex to the right of y. This means that for any vertex $v \prec y$ with $vy \notin E'_S$, we have $vy \notin E_S$, and for any vertex $y \prec v$ with $yv \in E'_S$, we have $yv \in E_S$. Observe also that the (added or removal) edges of $E'_S \setminus E_S$ and $E_S \setminus E'_S$ are incident to y or z. Thus, for any two vertices u, v different than y and z, we have $uv \in E'_S$ if and only if $uv \in E_S$.

Assume that there is a conflict in E'_S formed by the vertices u, v, y. Note that $u, v \neq z$, since y and z do not participate in a conflict. If both edges of the conflict belong to $E_S \cap E'_S$ and the non-edge of the conflict does not belong to $E_S \cup E'_S$, then such a conflict formed by the vertices u, v, y was already a conflict in E_S. So, let us assume next that at least one of the two edges in the conflict belongs to $E'_S \setminus E_S$ and is incident to y (i.e., $uy \in E'_S \setminus E_S$ or $vy \in E'_S \setminus E_S$) or the non-edge of the conflict belongs to $E_S \setminus E'_S$ and is incident to y (i.e., $uy \in E_S \setminus E'_S$).

- Suppose that $u \prec v \prec y$. Then, we have either $uy, uv \in E'_S$ and $vy \notin E'_S$, or $uy, vy \in E'_S$ and $uv \notin E'_S$. In the former case, we have the following:

  - Let $uy \in E'_S$, $uv \in E'_S$, and $vy \notin E'_S$. Notice that $uv \in E_S$ and $vy \notin E_S$, which means that $uy \in E'_S \setminus E_S$. Then, by the definition of E'_S, we have $u \in X(z) \cup A(z)$ and $v \notin X(z) \cup A(z)$. This means that $vz \notin E_S$ by Claim 4.9. Thus, $u \prec v \prec z$ form a conflict in E_S, because $uz \in E_S$ and $vz \notin E_S$. Moreover, observe that u, v, z is not a conflict in E'_S, because $uz \notin E'_S$ by the definition of E'_S.

  In the latter case, we distinguish between uy being an added edge:

  - Let $uy \in E'_S \setminus E_S$, $uv \notin E'_S$, and $vy \in E'_S$. Notice that $uv \notin E_S$. If $vy \in E'_S \setminus E_S$ then $u \prec v \prec z$ form a conflict in E_S, because $u, v \in X(z) \cup A(z)$ and $uz, vz \in E_S$. Moreover, by the definition of E'_S observe that $uz, vz \notin E'_S$, so that u, v, z is not a conflict in E'_S. If $vy \in E_S$ then, since $uz \in E_S$, v is strongly adjacent to u or z in E_S by Lemma 4.1. By the fact that $uv \notin E_S$, we get $vz \in E_S$. This shows that

$u \prec v \prec z$ form a conflict in E_S. Moreover, observe that u, v, z is not a conflict in E'_S, because $uz \notin E'_S$ by the definition of E'_S.

  – Let $uy \in E'_S \cap E_S$, $uv \notin E'_S$, and $vy \in E'_S \setminus E_S$. Notice that $uy \in E_S$, $uv \notin E_S$, $vy \notin E_S$. This, however, is not possible due to Lemma 4.1, since $u \prec v \prec y$ and v is not strongly adjacent to u and y.

- Suppose that $u \prec y \prec v$. Then, we have either $uv, uy \in E'_S$ and $yv \notin E'_S$, or $uv, yv \in E'_S$ and $uy \notin E'_S$. In the former case, we distinguish between uy being an added edge:

  – Let $uy \in E'_S \setminus E_S$, $uv \in E'_S$, and $yv \notin E_S \cup E'_S$. Notice that $uy \notin E_S$ and $uv \in E_S$. This, however, is not possible due to Lemma 4.1, since $uy, yv \notin E_S$.

  – Let $uy \in E'_S \setminus E_S$, $uv \in E'_S$, and $yv \in E_S \setminus E'_S$. Notice that $uy \notin E_S$, $uv \in E_S$, and $yv \in E_S$. Thus $u \prec y \prec v$ form a conflict in E_S.

  – Let $uy \in E_S \cap E'_S$, $uv \in E'_S$, and $yv \in E_S \setminus E'_S$. Notice that $uv \in E_S$ and $yv \in E_S$. Then, by the definition of E'_S, we have $v \in B(y) \cup W(y)$ which implies $zv \notin E_S$ and $y \prec z \prec v$. Moreover, by Lemma 4.1 we have $uz \in E_S$, since $u \prec z \prec v$ and $uv \in E_S$. Thus, $u \prec z \prec v$ form a conflict in E_S, because $uv, uz \in E_S$, and $zv \notin E_S$. Further, notice that u, z, v is not a conflict in E'_S, because $zv \in E'_S$ by $v \in B(y) \cup W(y)$ and $uz \in E'_S$ by $u \notin X(z) \cup A(z)$.

  In the latter case, we have the following:

  – Let $uy \notin E'_S$, $uv \in E'_S$, and $yv \in E'_S$. Notice that $uy \notin E_S$ because $u \prec y$, $uv \in E_S$, and $yv \in E_S$ because $y \prec v$. Thus, $u \prec y \prec v$ form a conflict in E_S.

- Suppose that $y \prec u \prec v$. Then, we have either $yv, yu \in E'_S$ and $uv \notin E'_S$, or $yv, uv \in E'_S$ and $yu \notin E'_S$. We distinguish between the two cases:

  – Let $yv \in E'_S$, $yu \in E'_S$ and $uv \notin E'_S$. Notice that $yv \in E_S$, $yu \in E_S$, because $y \prec \{u, v\}$, and $uv \notin E_S$. Thus, $y \prec u \prec v$ form a conflict in E_S.

  – Let $yu \notin E'_S$, $yv \in E'_S$, and $uv \in E'_S$. Notice that $yv \in E_S$ because $y \prec v$, and $uv \in E_S$. Thus, yu is a removal edge, so that $yu \in E_S \setminus E'_S$. Then, by the definition of E'_S, we have $u \in B(y) \cup W(y)$ and $v \notin B(y) \cup W(y)$. This means that $zu \notin E_S$ and by Claim 4.8 we have $zv \in E_S$. Thus, $z \prec u \prec v$ form a conflict in E_S, because $zu \notin E_S$, $zv \in E_S$, and $uv \in E_S$. Moreover, observe that z, u, v is not a conflict in E'_S, because $zu, zv \in E'_S$ by the definition of E'_S.

Putting together, for all edges incident to y that create new conflict in E'_S there is a unique conflict in E_S. With completely symmetric arguments, we get that edges incident to z that create new conflict in E'_S there is a unique conflict in E_S. Therefore, E'_S has strictly smaller number of conflicts than E_S, since the conflicts formed by y and z in E_S are not conflicts in E'_S. $\diamondsuit$

Thus, we replace appropriate set of strong edges in E_S and obtain an optimal solution by Claims 4.10 and 4.11 having a smaller number of conflicts by Claim 4.12. Therefore, by applying such a replacement in every possible conflict, we get an optimal solution that has no conflicts and, thus, it satisfies the consecutive strong property. $\square$

Lemma 4.3 suggests to find an optimal solution that has the consecutive strong property with respect to σ. In fact by the proper interval ordering, this reduces to computing the largest proper interval subgraph H of G such that the vertices of every P_3 of H induce a clique in G.

Let G be a proper interval graph and let $\sigma = \langle v_1, \ldots, v_n \rangle$ be its proper interval ordering. For a vertex v_i we denote by $\ell(i)$ and $r(i)$ the positions of its leftmost and rightmost neighbors, respectively, in σ. Observe that for any two vertices $v_i \prec v_j$ in σ, $v_{\ell(i)} \preceq v_{\ell(j)}$ and $v_{r(i)} \preceq v_{r(j)}$ [8]. For $1 \leq i \leq n$, let $V_i = \{v_i, \ldots, v_n\}$ and let $G_i = G[V_i]$. In what follows, we assume that E_S is a solution of G that has the consecutive strong property with respect to σ. Given E_S, we denote by $r_S(i)$ the position of the rightmost strong neighbor of v_i, that is,

$$r_S(i) = \begin{cases} n & \text{if } i = n \text{ or } v_i v_k \in E_S \text{ for every vertex } v_k \in V_{i+1}, \\ i & \text{if } v_i v_k \notin E_S \text{ for every vertex } v_k \in V_{i+1}, \\ k & \text{if } v_i v_k \in E_S \text{ and } v_i v_{k+1} \notin E_S \text{ for a vertex } v_k \in V_{i+1}. \end{cases}$$

Observe that if $r_S(i) = i$, then v_i has no strong neighbor in V_i. Moreover, notice that $i \leq r_S(i) \leq r(i)$. It is not difficult to see that given $r_S(i)$ for every vertex v_i, we can describe all strong edges of a solution E_S, since, by the consecutive strong property, for $i < j$, we have $v_i v_j \in E_S$ if and only if $i < j \leq r_S(i)$ holds.

We are now ready to define our subproblems with respect to σ. Let A_i be the value of an optimal solution E_S of G_i. Clearly A_1 corresponds to the value of an optimal solution of G.

Definition 4.13. *For $i \leq r \leq r(i)$, we denote by $A[i, r]$ the value of an optimal solution of G_i such that $r_S(i) = r$.*

As a base case, observe that $A_n = A[n, n] = 0$, because G_n contains exactly one vertex. By the consecutive strong property, we get the following equation, for any $1 \leq i \leq n$:

$$A_i = \max_{i \leq r \leq r(i)} A[i, r]. \tag{1}$$

For $i \leq r$, we denote by $r_{S(i,r)}(j)$ the position of the rightmost strong neighbor of every vertex $v_j \in V_i$ that is described by the value $A[i, r]$. For $1 \leq i < r \leq r' \leq n$, let $B_{i,r,r'}$ be the number of strong edges incident to $v_{i+1}, \ldots, v_{r-1}$ (i.e., to the vertices of $V_{i+1} \setminus V_r$) that belong to G_{i+1} in an optimal solution of G_i such that

- $r_S(i) = r$ and
- $r_S(j) = r_{S(r,r')}(j)$, for every $r \leq j \leq n$.

Observe that if $r = i + 1$ then $B_{i,r,r'} = 0$, as there is no vertex between v_i and v_r (i.e., the set $V_{i+1} \setminus V_r$ is empty). Moreover, notice that $r_{S(r,r')}(j)$ corresponds to the position of the rightmost strong neighbor of v_j described by $A[r, r']$.

In order to recursively compute $A[i, r]$, the key idea is that we try every allowed position r' of the rightmost strong neighbor of r. This is achieved through the consecutive strong property. Then, we split the solution into two parts among the vertices of $V_i \setminus V_r$ and V_r, respectively.

Lemma 4.14. *Let $1 \leq i \leq r \leq n$. Then $A_n = A[n, n] = 0$, $A[i, i] = A_{i+1}$, and*

$$A[i, r] = \max_{r \leq r' \leq r(i)} \left(A[r, r'] + B_{i,r,r'} + r - i \right).$$

Proof. The base cases follow by definition. Assume that $v_i v_r \in E_S$ with $i < r$ and v_r being the rightmost strong neighbor of v_i. It is clear that $r \leq r(i)$, as $\{v_i, v_r\} \in E(G)$. Moreover, for the rightmost strong neighbor $v_{r'}$ of v_r, we know that $\{v_i, v_{r'}\} \in E(G)$ since both strong edges $v_i v_r$ and $v_r v_{r'}$ satisfy the strong triadic closure. Thus we have $r \leq r' \leq r(i)$.

Furthermore, v_i is strongly adjacent to every vertex $v_{i+1}, \ldots, v_r$ by the consecutive strong property, implying that there are $r - i$ strong edges incident to v_i in G_i.

Let $S_{i,r}$ and $S_{r,r'}$ be the strong edges described by the values $A[i, r]$ and $A[r, r']$, respectively. Suppose that $\{x, y\}$ is an edge of G_r. We claim that $xy \in S_{i,r}$ if and only if $xy \in S_{r,r'}$. If $xy \in S_{i,r}$ then $xy \in S_{r,r'}$, since $V_r \subseteq V_i$. Assume for contradiction that $xy \in S_{r,r'}$ and $xy \notin S_{i,r}$. Then there is a vertex $w \in V_i \setminus V_r$ such that $wx \in S_{i,r}$ and $\{w, y\} \notin E(G)$. If $w \neq v_i$ then wx is not a strong edge described by the definition of $B_{i,r,r'}$. Thus $w = v_i$. Since v_r is the only strong neighbor of v_i in G_r, we know that $x = v_r$. For any strong neighbor y of v_r in $S_{r,r'}$ we know that $v_r \preceq y \preceq v_{r'}$ by the consecutive strong property. Then, however, by the fact that $r' \leq r(i)$ we reach a contradiction to $\{v_i, y\} \notin E(G)$. Therefore the given formula describes the considered solutions. □

Next we focus on computing $B_{i,r,r'}$. For doing so, we define a recursive formulation on subsolutions that take into account the position $r_S(i + 1)$ of the rightmost strong neighbor of v_{i+1}. Observe that $r \leq r_S(i + 1) \leq r'$ holds, since $v_i v_r \in E_S$ and $v_r v_{r'} \in E_S$ by the consecutive strong property. In fact, it is not difficult to verify that, for every $i < k < r$, $r \leq r_S(k) \leq r'$ holds, because $r_S(i) = r$ and $r_S(r) = r'$.

Definition 4.15. *For $i < r \leq t \leq r'$, we denote by $B[i, r, t, r']$ the number of strong edges incident to $v_{i+1}, \ldots, v_{r-1}$ (i.e., to the vertices of $V_{i+1} \setminus V_r$) that belong to G_{i+1} in an optimal solution of G_i such that*

- $r_S(i) = r$,
- $t \leq r_S(k)$, *for every $i < k < r$, and*
- $r_S(j) = r_{S(r,r')}(j)$, *for every $r \leq j \leq n$.*

By the definition of the describes values and the previous discussion on the consecutive strong property, we get the following equation:

$$B_{i,r,r'} = \max_{r \leq t \leq r'} B[i, r, t, r'] \tag{2}$$

For every $i < k < r$, let $x_{r'}(k)$ be the rightmost position j such that $j \leq r'$ and $r_{S(r,r')}(j) \leq r(k)$. Observe that $r \leq x_{r'}(k)$, because $r' \leq r(i) \leq r(k)$ for every $v_i \prec v_k$.

Lemma 4.16. *Let $1 \leq i < r \leq t \leq r' \leq n$. Then $B[r - 1, r, t, r'] = 0$ and*

$$B[i, r, t, r'] = \max_{t \leq j \leq x_{r'}(i+1)} \big(B[i + 1, r, j, r'] + j - (i + 1)\big).$$

Proof. Let v_j be the rightmost strong neighbor of v_{i+1} in G_{i+1}. Then $t \leq j$ holds, by the restriction described in $B[i, r, t, r']$. By the fact that $r_{S(r,r')}(r) = r'$ and $v_{i+1} \prec v_r$ we deduce that $j \leq r'$. Suppose that $x_{r'}(i+1) < j$. By the choice of $x_{r'}(i+1)$ we know that either $r' < j$ which is not possible, or $j \leq r'$ and there is a strong edge $v_j v \in S(r, r')$ with $\{v_{i+1}, v\} \notin E(G)$ which violates the strong triadic closure. Thus we conclude $t \leq j \leq x_{r'}(i + 1)$ and the consecutive strong property implies that there are $j - i - 1$ strong edges incident to v_{i+1} in G_{i+1}.

Let S_i and S_{i+1} be the strong edges described by $B[i, r, t, r']$ and $B[i + 1, r, j, r']$, respectively. Suppose that $\{v_{i+2}, v\}$ is an edge of G_{i+2}. We claim that $v_{i+2}v \in S_i$ if and only if $v_{i+2}v \in S_{i+1}$. If $v_{i+2}v \in S_i$ then it is clear that $v_{i+2}v \in S_{i+1}$, since $V_{i+1} = V_{i+2} \cup \{v_{i+1}\}$. Assume for contradiction that $v_{i+2}v \in S_{i+1}$ and $v_{i+2}v \notin S_i$. Observe that $v_r \preceq v \preceq v_{r'}$. Let z be the rightmost strong neighbor of v in S_{i+1}. Then, by definition, we have $vz \in S_i$. This means that there is a strong edge wv_{i+2} in $S_i \setminus S_{i+1}$ such that $\{w, v\} \notin E(G)$. By the definition of $B[i, r, t, r']$, we have that $w = v_{i+1}$. Then, however, we reach a contradiction to $\{v_{i+1}, v\} \notin E(G)$, since $v \preceq v_{r'}$ and $r' \leq r(i) \leq r(i+1)$. Therefore we conclude the described formula. $\square$

Now we are equipped with our necessary tools in order to obtain our main result, namely a polynomial-time algorithm that solves the MaxSTC problem on proper interval graphs.

Theorem 4.17. *There is a polynomial-time algorithm that computes the* MaxSTC *of a proper interval graph.*

Proof. Let G be a proper interval graph on n vertices and m edges. We first compute its proper interval ordering σ in linear time [25]. In order to compute an optimal solution A_1 for G described in Equations 1 and 2, we use a dynamic programming approach based on their recursive formulation given in Lemmas 4.14 and 4.16, respectively. For doing so, we store two tables $\mathcal{T}_A$ and $\mathcal{T}_B$ that correspond to the values $A[i, r]$ and $B[i, r, t, r']$. Whenever we compute the value $\mathcal{T}_A[i, r]$, we perform a second sweep to backtrack the actual rightmost strong neighbor of each vertex in V_i. More precisely, it suffices to backtrack the rightmost strong neighbors of each vertex of $V_i \setminus V_r$. Correctness follows from Lemmas 4.3, 4.14, and 4.16.

Regarding the running time, notice that all instances of $\mathcal{T}_A[i, r]$ and $\mathcal{T}_B[i, r, t, r']$ can be computed as follows. At each vertex v_i, we compute all possible vertex pairs $v_r, v_{r'}$ with $i \leq r \leq r' \leq r(i)$ which are bounded by n^2. For each vertex v_k with $i < k < r$, computing $x_{r'}(k)$ takes $O(n)$ time by scanning the rightmost strong neighbors of the vertices of $V_r \setminus V_{r'}$. Moreover, for each index t with $r \leq t \leq r'$, there are at most n choices. Thus the number of instances $\mathcal{T}_A[i, r]$ and $\mathcal{T}_B[i, r, t, r']$ generated by v_i is $O(n^3)$. Because we visit n vertices, the total running time of the algorithm is $O(n^4)$. $\square$

5 Concluding remarks

Given the first study with positive and negative results for the MaxSTC problem on restricted input, there are some interesting open problems. As we pointed out MaxSTC is more difficult than Cluster Deletion in the following sense: a solution for Cluster Deletion forms a solution for MaxSTC but the converse is not necessarily true. We have given examples showing that such an observation carries out for split graphs as well as for proper interval graphs. Despite the structural difference of both problems, our result on split graphs points out an important and interesting complexity difference between the two problems: on split graphs Cluster Deletion has already been shown to be polynomially solvable [3] whereas we prove that MaxSTC remains NP-complete. It is interesting to explore other graph classes that exhibit the same behavior.

Apart from the structural properties that we proved for the solution on proper interval graphs, the complexity of MaxSTC on interval graphs is still open. Determining the complexity of MaxSTC for other graph classes towards AT-free graphs seems to be an interesting direction for future work. More precisely, by Proposition 2.2 it is interesting to consider the line-incompatibility graph of comparability graphs since they admit a well-known similar characterization [20]. Moreover it is natural to characterize the graphs for which their line-incompatibility graph is perfect. As already mentioned, such a characterization will lead to further polynomial cases of MaxSTC [13].

More general there are extensions and variations of the MaxSTC problem that are interesting to consider as proposed in [32]. An interesting and realistic problem is to allow multiple types of strong edges $S_0, S_1, \ldots, S_k$ that do not allow violating "ordered" P_3's. More precisely the objective is to partition the edges of G into $S_0, S_1, \ldots, S_k$ with $k \geq 1$

so that there is no pair of edges $\{u, v\} \in S_i$ and $\{v, w\} \in S_i$ such that $\{u, w\} \notin E(G)$ and $|S_1| + \cdots + |S_k|$ is as large as possible. Another realistic scenario is to restrict the freedom of the choice of the strong edges [31]. Assume that a subset F of edges is required to be strong. Then it is natural to ask for a suitable set of edges $E' \subseteq E \setminus F$ with $|E'|$ as large as possible such that labeling the edges of $E' \cup F$ as strong satisfy the strong triadic closure. Clarifying the complexity of such generalized problem is interesting on graphs for which MaxSTC is solved in polynomial time.

Acknowledgements

We thank the reviewers for their valuable comments that helped improve the presentation of the paper.

References

[1] A. B. Adcock, B. D. Sullivan, and M. W. Mahoney. Tree decompositions and social graphs. *Internet Mathematics*, 12:315–361, 2016.

[2] L. Backstrom and J. Kleinberg. Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook. In *Proceedings of CSCW 2014*, pages 831–841, 2014.

[3] F. Bonomo, G. Durán, and M. Valencia-Pabon. Complexity of the cluster deletion problem on subclasses of chordal graphs. *Theoretical Computer Science*, 600:59–69, 2015.

[4] A. Brandstädt, V. B. Le, and J. Spinrad. *Graph Classes: A Survey*. Society for Industrial and Applied Mathematics, 1999.

[5] M. Cochefert, J.-F. Couturier, P. A. Golovach, D. Kratsch, and D. Paulusma. Parameterized algorithms for finding square roots. *Algorithmica*, 74:602–629, 2016.

[6] D.G. Corneil, H. Lerchs, and L.K. Stewart. Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981.

[7] B. Courcelle. The monadic second-order logic of graphs i: Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.

[8] X. Deng, P. Hell, and J. Huang. Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. *SIAM J. Comput.*, 25:390–403, 1996.

[9] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[10] J. Edmonds. Paths, trees and flowers. *Canad. J. Math.*, 17:449–467, 1965.

[11] M.C. Golumbic. Trivially perfect graphs. *Discrete Mathematics*, 24:105–107, 1978.

[12] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78:1360–1380, 1973.

[13] M. Grötschel. Polynomial algorithms for perfect graphs. *North-Holland Mathematics Studies*, 21:325–356, 1984.

[14] P. Heggernes, D. Lokshtanov, J. Nederlof, C. Paul, and J. A. Telle. Generalized graph clustering: recognizing (p, q)-cluster graphs. In *Proceedings of WG 2010*, pages 171–183, 2010.

[15] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2:225–231, 1973.

[16] L. Ibarra. The clique-separator graph for chordal graphs. *Discrete Applied Mathematics*, 157:1737–1749, 2009.

[17] M. O. Jackson. *Social and economic networks.* Princeton University press, vol. 3, 2008.

[18] R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.

[19] D. J. Kleitman and R. V. Vohra. Computing the bandwidth of interval graphs. *SIAM J. Disc. Math.*, 3:373–375, 1990.

[20] D. Kratsch, R. M. McConnell, K. Mehlhorn, and J. P. Spinrad. Certifying algorithms for recognizing interval graphs and permutation graphs. *SIAM J. Comput.*, 36:326–353, 2006.

[21] L. C. Lau. Bipartite roots of graphs. *ACM Transactions on Algorithms*, 2:178–208, 2006.

[22] L. C. Lau and D. G. Corneil. Recognizing powers of proper interval, split, and chordal graphs. *SIAM J. Disc. Math.*, 18:83–102, 2004.

[23] V. B. Le. Gallai graphs and anti-gallai graphs. *Discrete Mathematics*, 159:179–189, 1996.

[24] V. B. Le, A. Oversberg, and O. Schaudt. Polynomial time recognition of squares of ptolemaic graphs and 3-sun-free split graphs. *Theoretical Computer Science*, 602:39–49, 2015.

[25] P. J. Looges and S. Olariu. Optimal greedy algorithms for indifference graphs. *Computers & Mathematics with Applications*, 25:15–25, 1993.

[26] M. Milanič and O. Schaudt. Computing square roots of trivially perfect and threshold graphs. *Discrete Applied Mathematics*, 161:1538–1545, 2013.

[27] J. L. Pfaltz. Chordless cycles in networks. In *Proceedings of ICDE Workshops 2013*, pages 223–228, 2013.

[28] F. Protti, M. D. da Silva, and J. L. Szwarcfiter. Applying modular decomposition to parameterized cluster editing problems. *Theory Comput. Syst.*, 44:91–104, 2009.

[29] F. S. Roberts. Indifference graphs. In *Proof Techniques in Graph Theory, Academic Press, New York*, pages 139–146, 1969.

[30] R. Rotabi, K. Kamath, J. Kleinberg, and A. Sharma. Detecting strong ties using network motifs. In *Proceedings of WWW 2017*, pages 983–992, 2017.

[31] P. Rozenshtein, N. Tatti, and A. Gionis. Inferring the strength of social ties: a community-driven approach. In *Proceedings of KDD 2017*, pages 1017–1025, 2017.

[32] S. Sintos and P. Tsaparas. Using strong triadic closure to characterize ties in social networks. In *Proceedings of KDD 2014*, pages 1466–1475, 2014.

[33] J. Ugander, L. Backstrom, and J. Kleinberg. Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections. In *Proceedings of WWW 2013*, pages 1307–1318, 2013.