Strong Triadic Closure in Cographs and Graphs of Low Maximum Degree

Athanasios L. Konstantinidis* Stavros D. Nikolopoulos†
Charis Papadopoulos‡

Abstract

The MaxSTC problem is an assignment of the edges with two types of labels, namely, strong and weak, that maximizes the number of strong edges such that any two vertices that have a common neighbor with a strong edge are adjacent. The Cluster Deletion problem seeks for the minimum number of edge removals of a given graph such that the remaining graph is a disjoint union of cliques. Both problems are known to be NP-hard and an optimal solution for the Cluster Deletion problem provides a feasible solution for the MaxSTC problem, however not necessarily an optimal one. In this work we conduct the first systematic study that reveals graph families for which the optimal solutions for MaxSTC and Cluster Deletion coincide. We first show that MaxSTC coincides with Cluster Deletion on cographs and, thus, MaxSTC is solvable in polynomial time on cographs. As a side result, we give an interesting computational characterization of the maximum independent set on the cartesian product of two cographs. Furthermore, we address the influence of the low degree bounds to the complexity of the MaxSTC problem. We show that this problem is polynomial-time solvable on graphs of maximum degree three, whereas MaxSTC becomes NP-complete on graphs of maximum degree four. The proof of the latter result implies that there is no subexponential-time algorithm for MaxSTC unless the Exponential-Time Hypothesis fails.

1 Introduction

The principle of strong triadic closure is an important concept in social networks [7]. It states that it is not possible for two individuals to have a strong relationship with a common friend and not know each other [10]. The strong triadic closure is satisfied if the edges of the underlying graph are characterized into weak and strong such that any two vertices that have a strong neighbor in common are adjacent. Towards the investigation of the behavior of a network, such a principle has been recently proposed as a maximization problem, called MaxSTC, in which the goal is to assign each edge as strong or weak so that to maximize the number of strong edges of the underlying graph that satisfy the strong triadic closure [22]. Closely related to the MaxSTC problem is the Cluster Deletion problem which finds important applications in areas involving clustering [1]. In the latter problem the goal is to remove the minimum number of edges such that the resulting graph consists of vertex-disjoint union of cliques.

The relation between MaxSTC and Cluster Deletion arises from the fact that the edges inside the cliques in the resulting graph for Cluster Deletion can be seen as strong

*Department of Mathematics, University of Ioannina, Greece. E-mail: skonstan@cc.uoi.gr
†Department of Computer Science & Engineering, University of Ioannina, Greece. E-mail: stavros@cs.uoi.gr
‡Department of Mathematics, University of Ioannina, Greece. E-mail: charis@cs.uoi.gr

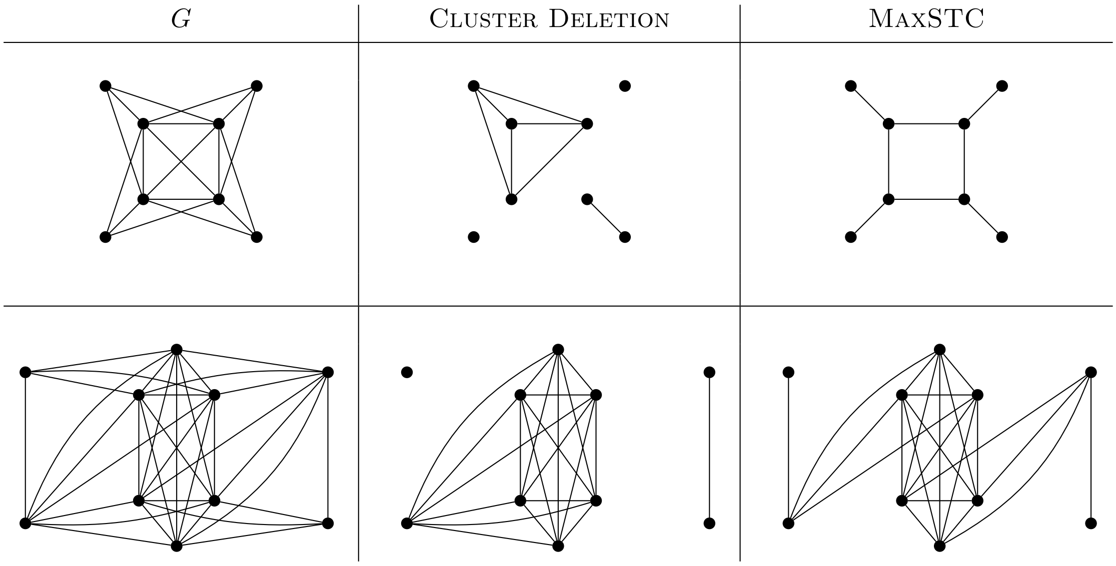


Figure 1: Two examples of graphs with their corresponding optimal solutions for Cluster Deletion and MaxSTC, respectively. For the MaxSTC problem the edges of G that are not drawn in the solution correspond to the weak edges.

edges for MaxSTC which satisfy the strong triadic closure. Thus, the number of edges in an optimal solution for Cluster Deletion consists a lower bound for the number of strong edges in an optimal solution for MaxSTC. However there are graphs (see for e.g., Figure 1) showing that an optimal solution for MaxSTC contains larger number of edges than an optimal solution for Cluster Deletion. Interestingly, there are also families of graphs in which their optimal value for MaxSTC matches such a lower bound. For instance, any maximum matching on graphs that do not contain triangles constitutes a solution for both problems. Here we initiate a systematic study on other non-trivial classes of graphs for which the optimal solutions for both problems have exactly the same value.

Our main motivation is to further explore the complexity of the MaxSTC problem when restricted to graph classes. As MaxSTC has been recently introduced, there are few results concerning its complexity. The problem has been shown to be NP-complete for general graphs [22] and split graphs [17] whereas it becomes polynomial-time tractable on proper interval graphs and trivially perfect graphs [17]. The NP-completeness on split graphs shows an interesting algorithmic difference between the two problems, since Cluster Deletion on such graphs can be solved in polynomial time [2]. It is known that Cluster Deletion is NP-complete on general graphs [21] and remains NP-complete on chordal graphs and, also, on graphs of maximum degree four [2, 15]. On the positive side Cluster Deletion admits polynomial-time algorithms on proper interval graphs [2], graphs of maximum degree three [15], and cographs [9]. In fact for cographs a greedy algorithm that finds iteratively maximum cliques gives an optimal solution, although no running time was explicitly given in [9].

Such a greedily approach is also proposed for computing a maximal independent set of the cartesian product of general graphs. Summing the partial products between iteratively maximum independent sets consists a lower bound for the cardinality of the maximum independent set of the cartesian product [13, 14]. Here we prove that a maximum independent set of the cartesian product of two cographs matches such a lower bound. We would like to note that a polynomial-time algorithm for computing such a maximum independent set is already claimed [11]. However neither a characterization is given, nor an explicit running time of the algorithm is reported.

Our results. In this work we further explore the complexity of the MaxSTC problem. We consider two unrelated families of graphs, namely, cographs and graphs of bounded degree. Cographs are characterized by the absence of a chordless path on four vertices. For such graphs we prove that the optimal value for MaxSTC matches the optimal value for Cluster Deletion. For doing so, we reveal an interesting vertex partitioning with respect to their maximum clique and maximum independent set. This result enables us to give an $O(n^2)$-time algorithm for MaxSTC on cographs. As a byproduct we characterize a maximum independent set of the cartesian product of two cographs which implies a polynomial-time algorithm for computing such a maximum independent set. Moreover we study the influence of low maximum degree for the MaxSTC problem. We show an interesting complexity dichotomy result: for graphs of maximum degree four MaxSTC remains NP-complete, whereas for graphs of maximum degree three the problem is solved in polynomial time. Our reduction for the NP-completeness on graphs of maximum degree four implies that, under the Exponential-Time Hypothesis, there is no subexponential time algorithm for MaxSTC. A preliminary version of this work appeared as an extended abstract in the proceedings of COCOON 2017 [16].

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 $N[v] = N(v) \cup \{v\}$. The *degree* of v is $d(v) = |N(v)|$. 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, denoted by $G[X]$, has vertex set X, and for each vertex pair u, v from X, $\{u, v\}$ is an edge of $G[X]$ if and only if $u \neq v$ and $\{u, v\}$ 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.

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 $k \geq 3$, the chordless path on k vertices is denoted by P_k and the chordless cycle on k vertices is denoted by C_k. We denote by K_k a clique on k vertices whereas the special case for $k = 3$ is called *triangle*.

Given a graph $G = (V, E)$, a *strong-weak labeling* on the edges of G is a bijection $\lambda : E(G) \rightarrow \{\text{strong}, \text{weak}\}$; i.e., λ assigns to each edge of $E(G)$ a *strong* or *weak* label. An edge that is labeled strong (resp., weak) is simple called *strong* (resp. *weak*). The *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 $\{u, w\}$. We say that such a labeling *satisfies* the strong triadic closure. The problem of computing the maximum strong triadic closure, denoted by MaxSTC, is to find a strong-weak labeling on the edges of $E(G)$ that satisfies the strong triadic closure and has the maximum number of strong edges.

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 $G \setminus E_W$; notice that the graph spanned by E_S consists of the whole vertex set $V(G)$ and it may contain vertices with degree equal to zero. For a strong edge $\{u, v\}$, we say that u (resp., v) is a strong neighbor of v (resp., u). We denote by $N_S(v) \subseteq N(v)$ the strong neighbors of v. Given an optimal solution for MaxSTC that consists of the strong edges E_S, the graph spanned by the edges of E_S is denoted by $E_S(G)$. Whenever we write $|E_S(G)|$ we refer to its number of edges, that is $|E_S(G)| = |E_S|$.

In the Cluster Deletion problem the goal is to partition the vertices of a given graph G into vertex-disjoint cliques with the minimum number of edges outside the cliques, or, equivalently, with the maximum number of edges inside the cliques. A *cluster graph* is a graph in which every connected component is a clique. Cluster graphs are characterized as exactly the graphs that do not contain a P_3 as an induced subgraph. Given an optimal solution for Cluster Deletion, the cluster graph spanned by the edges that are inside the cliques is denoted by $E_C(G)$. We write $|E_C(G)|$ to denote the number of edges in the cluster graph. Notice that if we assign strong labels to all the edges of a cluster graph then such a labeling satisfies the strong triadic closure of the given graph. Thus $|E_C(G)| \leq |E_S(G)|$ holds for any graph G.

Figure 1 shows two graphs in which the optimal solution of Cluster Deletion contains strictly less edges than the optimal solution for MaxSTC. In terms of $E_C(G)$ and $E_S(G)$ notice that in such cases we have $|E_C(G)| < |E_S(G)|$, though in general $|E_C(G)| \leq |E_S(G)|$ holds. In the first example, from top to bottom, an optimal solution of Cluster Deletion consists of 7 edges whereas there is a solution of MaxSTC that contains 8 edges. The second example shows an optimal solution of Cluster Deletion with 22 edges, whereas there is a solution of MaxSTC with 23 edges. Notice that in the second example the 6 vertices drawn in the middle induce a clique on 6 vertices.

3 Computing MaxSTC on Cographs

Let $G = (V, E)$ and $H = (W, F)$ be two undirected graphs with $V \cap W = \emptyset$. The *disjoint union* of G and H is the graph obtained from the union of G and H, denoted by $G \oplus H = (V \cup W, E \cup F)$. The *complete join* of G and H is the graph obtained from the union of G and H and adding edges between every pair of vertices that belong to different graphs, denoted by $G \otimes H = (V \cup W, E \cup F \cup \{vw \mid v \in V, w \in W\})$. A graph is a *cograph* if it can be generated from single-vertex graphs and recursively applying the disjoint union and complete join operations. The complement of a cograph is also a cograph. Cographs are exactly the graphs that do not contain any chordless path on four vertices [4], and they can be recognized in linear time [5].

Let G be the given cograph. Our main goal is to show that there is an optimal solution for MaxSTC on G that coincides with an optimal solution for Cluster Deletion. The strong edges that belong to an optimal solution for MaxSTC span the graph $E_S(G)$. An optimal solution for Cluster Deletion consists of a cluster graph $E_C(G)$ by removing a minimum number of edges of G. Labeling all edges of a cluster graph as strong, results in a strong-weak labeled graph that satisfy the strong triadic closure. Thus, our goal is to show that there is an optimal solution $E_S(G)$ for MaxSTC that is a cluster graph.

A clique (resp. independent set) of G having the maximum number of vertices is denoted by $C_{\max}(G)$ (resp., $I_{\max}(G)$). A *greedy clique partition* of G, denoted by $\mathcal{C}$, is the ordering of cliques $(C_1, C_2, \ldots, C_p)$ in G such that

- $C_1 = C_{\max}(G)$ and
- $C_i = C_{\max}\left(G - \bigcup_{j=1}^{i-1} C_j\right)$ for $i = 2, 3, \ldots, p$.

Similarly, a *greedy independent set partition* of G, denoted by $\mathcal{I}$, is the ordering of independent sets $(I_1, I_2, \ldots, I_q)$ in G such that

- $I_1 = I_{\max}(G)$ and
- $I_i = I_{\max}\left(G - \bigcup_{j=1}^{i-1} I_j\right)$ for $i = 2, 3, \ldots, q$.

Observe that the subgraph spanned by the edges of $\mathcal{C}$ does not contain any P_3 and, thus, consists a solution for CLUSTER DELETION. Although in general a greedy clique partition does not necessarily imply an optimal solution for CLUSTER DELETION, when restricted to cographs the optimal solution is characterized by the greedy clique partition.

Lemma 3.1 ([9])**.** *Let G be a cograph with a greedy clique partition $\mathcal{C}$. Then the edges of $\mathcal{C}$ span an optimal solution $E_C(G)$ for* CLUSTER DELETION.

We will use such a characterization of CLUSTER DELETION in order to give its equivalence with the MAXSTC problem. Notice, however, that due to the freedom of the adjacencies between the cliques of a greedy clique partition, it is not sufficient to consider such a partition of the vertices. For doing so, we will further decompose the cliques of a greedy clique partition. It is known that a graph G is a cograph if and only if for any maximal clique C and any maximal independent set I of every induced subgraph of G, $|C \cap I| = 1$ holds (also known as the *clique-kernel intersection* property) [4]. Thus, we state the following lemma.

Lemma 3.2 ([4])**.** *Let G be a cograph. Then $C_{\max}(G) \cap I_{\max}(G) = \{v\}$ for some vertex v.*

We recursively apply Lemma 3.2 to obtain the following result.

Lemma 3.3. *Let G be a cograph with a greedy clique partition $\mathcal{C} = (C_1, \ldots, C_p)$ and a greedy independent set partition $\mathcal{I} = (I_1, \ldots, I_q)$. For every i, j with $1 \leq i \leq p$ and $1 \leq j \leq q$, if $|C_i| \geq j$ or $|I_j| \geq i$ then $C_i \cap I_j \neq \emptyset$.*

Proof. We prove that if $|C_i| \geq j$ or $|I_j| \geq i$ then $C_i \cap I_j \neq \emptyset$. Assume for contradiction that there exist C_i and I_j such that $C_i \cap I_j = \emptyset$. Let i and j be the smallest integers for which $C_i \cap I_j = \emptyset$. By the choice of j we know that for every $j' < j$, $C_i \cap I_{j'} \neq \emptyset$ holds because $|C_i| \geq j > j'$. This means that there are $j-1$ vertices $u_1, \ldots, u_{j-1}$ such that $C_i \cap I_1 = \{u_1\}, \ldots, C_i \cap I_{j-1} = \{u_{j-1}\}$. Similarly, for every $i' < i$ we have $|C_{i'}| \geq |C_i| \geq j$, by the greedy choice of $C_1, \ldots, C_i$. Thus there are $i-1$ vertices $v_1, \ldots, v_{i-1}$ such that $C_1 \cap I_j = \{v_1\}, \ldots, C_{i-1} \cap I_j = \{v_{i-1}\}$. Let $G_{i,j}$ be the graph obtained from G by removing the sets of vertices $C_1, \ldots, C_{i-1}$ and $I_1, \ldots, I_{j-1}$. Notice that $G_{i,j}$ contains at least one vertex because $|C_i| \geq j$ or $|I_j| \geq i$. We will prove that $C_{\max}(G_{i,j}) = C_i \setminus \{u_1, \ldots, u_{j-1}\}$ and $I_{\max}(G_{i,j}) = I_j \setminus \{v_1, \ldots, v_{i-1}\}$.

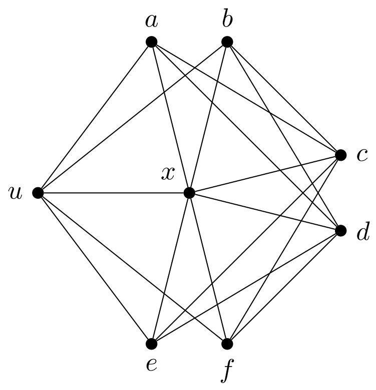
Let C' be the vertices of $C_1, \ldots, C_{i-1}$ and let I' be the vertices of $I_1, \ldots, I_{j-1}$. By the greedy independent set partition, the vertices of $G_{i,j}$ can be partitioned into $|C_i| - j + 1$ independent sets $I_j \setminus C', \ldots, I_{|C_i|} \setminus C'$. This implies that a maximum clique of $G_{i,j}$ has size at most $|C_i| - j + 1$. As $G_{i,j}$ is an induced subgraph of G, $C_i \setminus \{u_1, \ldots, u_{j-1}\}$ is a clique of size $|C_i| - j + 1$ of $G_{i,j}$. Thus, we have $C_{\max}(G_{i,j}) = C_i \setminus \{u_1, \ldots, u_{j-1}\}$. Following symmetric arguments, the vertices of $G_{i,j}$ can be partitioned into $|I_j| - i + 1$ cliques $C_i \setminus I', \ldots, C_{|I_j|} \setminus I'$. This implies that a maximum independent set of $G_{i,j}$ has size at most $|I_j| - i + 1$. Thus $I_{\max}(G_{i,j}) = I_j \setminus \{v_1, \ldots, v_{i-1}\}$.

Notice that $\{u_1, \ldots, u_{j-1}\} \cap \{v_1, \ldots, v_{i-1}\} = \emptyset$, due to the choice of i and j. Then Lemma 3.2 applies to $G_{i,j}$, which shows that

$$(C_i \setminus \{u_1, \ldots, u_{j-1}\}) \cap (I_j \setminus \{v_1, \ldots, v_{i-1}\}) \neq \emptyset.$$

Therefore $C_i \cap I_j \neq \emptyset$, leading to a contradiction that proves the desired statement. □

Lemma 3.3 suggests a partition of the vertices of G with respect to $\mathcal{C}$ and $\mathcal{I}$ as follows. We call *greedy canonical partition* a pair $(\mathcal{C}, \mathcal{I})$ with elements $\langle v_{i,j} \rangle$, where $1 \leq i \leq p$ and $1 \leq j \leq |C_i|$, such that $V(G) = \left\{v_{1,1}, \ldots, v_{p,|C_p|}\right\}$ and $v_{i,j} \in C_i \cap I_j$. Figure 2 shows such a greedy canonical partition of a given cograph. Observe that such a partition corresponds



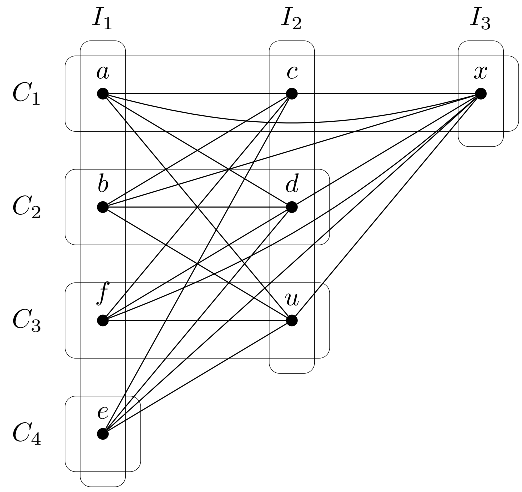


Figure 2: A cograph and its greedy canonical partition $(\mathcal{C}, \mathcal{I})$ where $\mathcal{C} = (C_1, C_2, C_3, C_4)$ and $\mathcal{I} = (I_1, I_2, I_3)$.

to a 2-dimensional representation of G. By Lemma 3.3 it follows that a cograph admits a greedy canonical partition.

Let us turn our attention back to the initial MaxSTC problem. We first consider the disjoint union of cographs.

Lemma 3.4. *Let G and H be vertex-disjoint cographs. Then $E_S(G \oplus H) = E_S(G) \oplus E_S(H)$ and $E_C(G \oplus H) = E_C(G) \oplus E_C(H)$.*

Proof. There are no edges between G and H so that a strong edge of G and a strong edge of H have no common endpoint. Thus the union of the solutions for G and H satisfy the strong triadic closure. By Lemma 3.1, $E_C(G \oplus H)$ contains the edges of a greedy clique partition which is obtained from the corresponding cliques of G and H. □

We next consider the complete join of cographs. Given two vertex-disjoint cographs G and H with greedy clique partitions $\mathcal{C} = (C_1, \ldots, C_p)$ and $\mathcal{C}' = (C'_1, \ldots, C'_{p'})$, respectively, we denote by $C_i(G, H)$ the edges that have one endpoint in C_i and the other endpoint in C'_i, for every $1 \leq i \leq \min\{p, p'\}$.

Lemma 3.5. *Let G and H be vertex-disjoint cographs with greedy clique partitions $\mathcal{C} = (C_1, \ldots, C_p)$ and $\mathcal{C}' = (C'_1, \ldots, C'_{p'})$, respectively. Then,*

- $E_S(G \otimes H) = (E_S(G) \oplus E_S(H)) \cup E(G, H)$ *and*
- $E_C(G \otimes H) = (E_C(G) \oplus E_C(H)) \cup E(G, H)$,

where $E(G, H) = C_1(G, H) \cup \cdots \cup C_k(G, H)$ and $k = \min\{p, p'\}$.

Proof. For the edges of $E_C(G \otimes H)$ we know that a greedy clique partition of $G \otimes H$ forms an optimal solution by Lemma 3.1. A greedy clique partition of $G \otimes H$ is obtained from the cliques $C_i \cup C'_i$, for every $1 \leq i \leq k$, since all the vertices of G are adjacent to all the vertices of H. The edges of $C_i \cup C'_i$ can be partitioned into the sets $E(C_i), E(C'_i)$, and $C_i(G, H)$ giving the desired formulation for $E_C(G \otimes H)$.

We consider the optimal solution for MaxSTC described by the edges of $E_S(G \otimes H)$. Let us show that any solution on the edges of G satisfy the strong triadic closure in the graph $G \otimes H$. Consider a strong edge $\{x, y\}$ of G. If the resulting labeling does not satisfy the strong triadic closure then there is a strong edge $\{x, w\}$ such that y and w are non-adjacent.

As G and H are vertex-disjoint graphs, $w \in V(G)$ or $w \in V(H)$. If $w \in V(G)$ then we already know that the labeling of $E_S(G)$ satisfies the strong triadic closure so that y and w are adjacent. If $w \in V(H)$ then by the complete join operation w is adjacent to y. Thus maximizing the number of strong edges that belong in G and H results in an optimal solution for $G \otimes H$.

We next consider the edges that have one endpoint in G and the other in H, denoted by $E(G, H)$. Our goal is to show that edges of $C_1(G, H) \cup \cdots \cup C_k(G, H)$ belong to an optimal solution. Let $(\mathcal{C}, \mathcal{I})$ and $(\mathcal{C}', \mathcal{I}')$ be the greedy canonical partitions of G and H, respectively, where

- $\mathcal{C} = (C_1, \ldots, C_p)$, $\mathcal{I} = (I_1, \ldots, I_q)$, and
- $\mathcal{C}' = (C'_1, \ldots, C'_{p'})$, $\mathcal{I}' = (I'_1, \ldots, I'_{q'})$.

Now observe that $|C_1(G, H) \cup \cdots \cup C_k(G, H)| = \sum_{i=1}^{k} |C_i||C'_i|$. Notice that the edges of $C_1(G, H) \cup \cdots \cup C_k(G, H)$ satisfy the strong triadic closure, since every two strong edges incident to a vertex of G belong to $C_i(G, H)$ which implies that the endpoints of H belong to a clique C'_i and, thus, are adjacent in $G \otimes H$. Therefore, we have $|E_S(G \otimes H)| \geq |E_C(G \otimes H)|$ and

$$|E(G, H)| \geq \sum_{i=1}^{k} |C_i||C'_i|.$$

In the forthcoming arguments we prove that $|E(G, H)| \leq \sum_{i=1}^{k} |C_i||C'_i|$.

We consider the vertices of I_j, $1 \leq j \leq q$, and count the number of strong edges that have one endpoint in I_j and the other endpoint on a vertex of H. Without loss of generality assume that $|I_1| \leq |I'_1|$. Then, $k = |I_1|$ since $p = |I_1|$ and $p' = |I'_1|$ by Lemma 3.3. For a subset W of vertices of G, we denote by $s(W)$ the number of strong edges of $E(G, H)$ that are incident to the vertices of W. By the strong triadic closure principle, any vertex of H has at most one strong neighbor in I_j and any vertex of G has at most one strong neighbor in $I'_{j'}$, $1 \leq j' \leq q'$. Thus, for any $I'_{j'}$ of H there are at most $\min\{|I_j|, |I'_{j'}|\}$ strong edges between the vertices of I_j and $I'_{j'}$. Let r_j be the largest index of $\{1, \ldots, q'\}$ for which $|I'_{r_j}| \geq |I_j|$; notice that r_j exists, since $|I_j| \leq |I_1| \leq |I'_1|$. Then, since $|I'_1| \geq \cdots \geq |I'_{q'}|$, it is clear that $|I_j|$ is smaller than or equal to any of $|I'_1|, \ldots, |I'_{r_j}|$ and greater than to any of $|I'_{r_j+1}|, \ldots, |I'_{q'}|$. Thus, we get the following inequality:

$$s(I_j) \leq \sum_{j'=1}^{q'} \min\{|I_j|, |I'_{j'}|\} = \sum_{j'=1}^{r_j} |I_j| + \sum_{j'=r_j+1}^{q'} |I'_{j'}|.$$

We next describe the vertices of $I'_1, \ldots, I'_{r_j}, I'_{r_j+1}, \ldots, I'_{q'}$ by the cliques of H. In particular, for every $1 \leq i \leq |I_j|$, we consider a clique C'_i of H. By Lemma 3.3 we know that C'_i contains exactly one vertex from each of $I'_1, \ldots, I'_{r_j}, I'_{r_j+1}, \ldots, I'_{|C'_i|}$. This means that all previously described vertices are contained in the disjoint union of cliques $C'_1, \ldots, C'_{|I_j|}$. Thus, the previous inequality can be written as follows.

$$s(I_j) \leq \sum_{j'=1}^{r_j} |I_j| + \sum_{j'=r_j+1}^{q'} |I'_{j'}| = \sum_{i=1}^{|I_j|} |C'_i|.$$

Summing up each of $s(I_j)$ for every I_j, $1 \leq j \leq q$, we obtain:

$$|E(G, H)| = \sum_{j=1}^{q} s(I_j) \leq \sum_{j=1}^{q} \sum_{i=1}^{|I_j|} |C'_i|.$$

Observe that, in the described sum, each $|C'_i|$ is counted for all $1 \leq j \leq q$ for which $|I_j| \geq i$. For such $|I_j|$ and i, by Lemma 3.3 we have $C_i \cap I_j \neq \emptyset$. Thus, the number that $|C'_i|$ appears in the formula is exactly $|C_i|$. Moreover, by the greedy canonical partition we know that $\sum_{j=1}^{q} |I_j| = \sum_{i=1}^{p} |C_i|$ and $p = |I_1|$. Hence, we get the desired upper bound for the number of strong edges in $E(G, H)$:

$$|E(G,H)| \leq \sum_{j=1}^{q} \sum_{i=1}^{|I_j|} |C'_i| = \sum_{i=1}^{|I_1|} |C_i||C'_i|.$$

Therefore, the claimed formula holds for the strong edges of $E_S(G \otimes H)$ and this concludes the proof. □

We are now ready to state our claimed result, namely that the solutions for MaxSTC and Cluster Deletion coincide for the class of cographs.

Theorem 3.6. *Let G be a cograph. There is an optimal solution for* MaxSTC *on G that is a cluster graph. Moreover* MaxSTC *on G can be solved in $O(n^2)$ time.*

Proof. An optimal solution for MaxSTC coincides with an optimal solution for Cluster Deletion trivially for graphs that consist of a single vertex. If G is a non-trivial cograph then it is constructed by the disjoint union or the complete join operation. In the former case Lemma 3.4 applies, whereas in the later Lemma 3.5 applies showing that in all cases $E_S(G) = E_C(G)$.

Regarding the running time, a maximum clique C_1 of G can be found in $O(n)$ time [4], due to a suitable data structure called *cotree*. We first construct the cotree of G which takes time $O(n + m)$ [5]. Removing a vertex v from a cograph G and updating the cotree takes $O(d(v))$ time, where $d(v)$ is the degree of v in G [20]. Thus, after removing all vertices from G we can mantain the cotree in an overall $O(n + m)$ time. In every intermediate step, we first remove the set of vertices C_i in $O(d(C_i))$ time where $d(C_i)$ is the sum of the degree of the vertices of C_i, and then spend $O(n)$ time to compute a maximum clique by using the resulting cotree. Therefore, since there are at most n such cliques in $\mathcal{C}$, a greedy clique partition of G can be found in total $O(n^2)$ time. □

3.1 Maximum independent set of the cartesian product of cographs

In this section we apply the characterization of Theorem 3.6 in order to show an interesting computational characterization of the cartesian product of cographs. Towards such a characterization we take advantage of an equivalent transformation of an optimal solution for MaxSTC in terms of a maximum independent set of an auxiliary graph that is called the *line-incompatibility* graph. The line-incompatibility graph (also known under the term *Gallai graph* [3, 18]), denoted by $\Gamma(G)$, has a node uv in $\Gamma(G)$ for every edge $\{u, v\}$ of G, and two nodes uv, vw of $\Gamma(G)$ are adjacent if and only if the vertices u, v, w induce a P_3 in G. The connection between a maximum independent set in $\Gamma(G)$ and a solution for MaxSTC in G is given in the following result.

Proposition 3.7 ([17])**.** *For any graph G, a subset E_S of edges span $E_S(G)$ if and only if the nodes corresponding to E_S form $I_{\max}(\Gamma(G))$.*

Let G and H be two vertex-disjoint graphs. The *cartesian product* of G and H, denoted by $G \times H$, is the graph with the vertex set $V(G) \times V(H)$ and any two vertices (u, u') and (v, v') are adjacent in $G \times H$ if and only if either $u = v$ and u' is adjacent to v' in H, or $u' = v'$ and u is adjacent to v in G. We are interested in computing a maximum independent set of $G \times H$ whenever G and H are cographs. We first characterize the graph $\Gamma(G \otimes H)$ in terms of $G \times H$.

Lemma 3.8. *Let G and H be two vertex-disjoint cographs. Then, $\Gamma(G \otimes H) = \Gamma(G) \oplus \Gamma(H) \oplus \left(\overline{G} \times \overline{H}\right)$.*

Proof. Notice that $G \otimes H$ is a connected cograph, as every vertex of G is adjacent to every vertex of H. The edges of $G \otimes H$ can be partitioned into the following sets of edges: $E(G)$, $E(H)$, and $E(G, H)$ where $E(G, H)$ is the set of edges between G and H in $G \otimes H$. By definition the nodes of $\Gamma(G)$ and $\Gamma(H)$ correspond to the sets $E(G)$ and $E(H)$. Moreover since G and H are vertex-disjoint graphs, $\Gamma(G)$ and $\Gamma(H)$ are also node-disjoint. This means that there are no common endpoints in the edges inside G and H. Hence every node of $\Gamma(G)$ is non-adjacent to all nodes of $\Gamma(H)$.

Next we show that every node of $\Gamma(G \otimes H)$ that corresponds to an edge of $E(G, H)$ is non-adjacent to the nodes of $\Gamma(G)$ and $\Gamma(H)$. If a node xy of $\Gamma(G)$ is adjacent to a node xa of $E(G, H)$ then a is a vertex of H and $\{y, a\}$ is not an edge of $G \otimes H$ contradicting the adjacency between the vertices of G and H. Symmetric arguments show that any node of $\Gamma(H)$ is non-adjacent to any node of $E(G, H)$. Thus no node that corresponds to an edge of $E(G, H)$ is adjacent to any node of $\Gamma(G) \oplus \Gamma(H)$.

To complete the proof we need to show that graph of $\Gamma(G \otimes H)$ induced by the nodes of $E(G, H)$ is exactly the graph $\overline{G} \times \overline{H}$. Let x, y be two vertices of G and let w, z be two vertices of H. By the definition of $\Gamma(G \otimes H)$, two nodes xw, yz are adjacent if and only if either $x = y$ and w is non-adjacent to z in H (so that w is adjacent to z in $\overline{H}$), or $w = z$ and x is non-adjacent to y in G (so that x is adjacent to y in $\overline{G}$). Such an adjacency corresponds exactly to the definition of the cartesian product of $\overline{G}$ and $\overline{H}$. Therefore the graph of $\Gamma(G \otimes H)$ induced by the nodes of $E(G, H)$ is exactly the graph $\overline{G} \times \overline{H}$. □

Now we are ready to give the characterization of a maximum independent set of the cartesian product of cographs, in terms of their greedy independent set partition. Although a polynomial-time algorithm for computing such a maximum independent set has already been claimed earlier [11], no characterization is proposed nor an explicit bound on the running time is reported.

Theorem 3.9. *Let G and H be two vertex-disjoint cographs with greedy independent set partitions $\mathcal{I} = (I_1, \ldots, I_q)$ and $\mathcal{I}' = (I'_1, \ldots, I'_{q'})$, respectively. Then the vertices of $(I_1 \times I'_1) \oplus \cdots \oplus (I_\ell \times I'_\ell)$ form a maximum independent set of $G \times H$, where $\ell = \min\{q, q'\}$. Moreover $I_{\max}(G \times H)$ can be computed in $O(n^2)$ time, where $n = \max\{|V(G)|, |V(H)|\}$.*

Proof. Let $(C_1, \ldots, C_p)$ and $(C'_1, \ldots, C'_{p'})$ be greedy clique partitions of G and H, respectively. By Lemma 3.5, we know that $E_S(G \otimes H) = E_S(G) \oplus E_S(H) \cup E(G, H)$, where $E(G, H) = C_1(G, H) \cup \cdots \cup C_k(G, H)$ and $k = \min\{p, p'\}$. Notice that if $(C_1, \ldots, C_p)$ is a greedy clique partition for G then $(C_1, \ldots, C_p)$ is a greedy independent set partition for $\overline{G}$. Moreover, by Proposition 3.7, we know that the edges of $E_S(G \otimes H)$ correspond to the nodes of $I_{\max}(\Gamma(G \otimes H))$. Since $\Gamma(G \otimes H) = \Gamma(G) \oplus \Gamma(H) \oplus \left(\overline{G} \times \overline{H}\right)$ by Lemma 3.8, we get $E(G, H) = I_{\max}(\overline{G} \times \overline{H})$. Therefore, the vertices of $(I_1 \times I'_1) \oplus \cdots \oplus (I_\ell \times I'_\ell)$ consist a $I_{\max}(\Gamma(G \otimes H))$.

For the running time, we need to compute two greedy independent set partitions $(I_1, \ldots, I_q)$ and $(I'_1, \ldots, I'_{q'})$ for G and H, respectively, and then combine each of I_j with I'_j, for $1 \leq j \leq \ell$. Computing a greedy independent set partition for a cograph G can be done in $O(n^2)$ time by applying the algorithm on $\overline{G}$ given in the proof of Theorem 3.6. Therefore, the total running time is bounded by $O(|V(G)|^2 + |V(H)|^2)$. □

4 Graphs of Low Maximum Degree

Here we study the influence of the bounded degree in a graph for the MaxSTC problem. We show an interesting complexity dichotomy result: for graphs of maximum degree four MaxSTC remains NP-complete, whereas for graphs of maximum degree three the problem has a polynomial solution.

We prove the hardness result even on a proper subclass of graphs with maximum degree four. A graph G is a *4-regular K_4-free graph*, if every vertex of G has degree four and there is no K_4 in G. The decision version of MaxSTC takes as input a graph G and an integer k and asks whether there is a strong-weak labeling of G that satisfies the strong triadic closure with at least k strong edges. Similarly the decision version of Cluster Deletion takes as input a graph G and an integer k and asks whether G has a spanning cluster subgraph by removing at most k edges. It is known that the decision version of Cluster Deletion on connected 4-regular K_4-free graphs is NP-complete [15].

Theorem 4.1. *The decision version of* MaxSTC *is NP-complete on connected 4-regular K_4-free graphs.*

Proof. We give a polynomial-time reduction to MaxSTC from the Cluster Deletion problem on connected 4-regular K_4-free graphs which is already known to be NP-complete [15]. Let $G = (V, E)$ be a connected 4-regular K_4-free graph with $n = 3q$ and $2n$ edges. Let $E_C(G)$ be a solution for the Cluster Deletion with $k = n$ edges. It is not difficult to see that every connected component of $E_C(G)$ is a triangle, since the graph is 4-regular and K_4 is a forbidden graph [15]. Then $E_C(G)$ is a solution for MaxSTC with at least n strong edges.

For the opposite direction, assume that $E_S(G)$ is a solution for MaxSTC with at least n strong edges. We show that the graph spanned by the strong edges of $E_S(G)$ is a two-regular graph. That is, every vertex of G has exactly two strong neighbors. Assume that there is a vertex v that has at least three strong neighbors. By the strong triadic closure all its strong neighbors must induce a clique in G. Then $N[v]$ induces a K_4 which is a forbidden subgraph. Thus every vertex has at most two strong neighbors. Furthermore if there is a vertex having only one strong neighbor then $|E_S(G)| < n$ which contradicts the assumption of n strong edges. Hence every vertex has exactly two strong neighbors in $E_S(G)$.

Since $E_S(G)$ is a 2-regular graph we know that the graph spanned by the strong edges is the disjoint union of triangles or chordless cycles C_p, with $4 \leq p \leq n$. Let us also rule out that a connected component of $E_S(G)$ is a chordless cycle on four vertices C_4. To see this, observe that if there is a C_4 in $E_S(G)$ then the four vertices of the C_4 induce a K_4 in G. Now assume that there is a connected component of $E_S(G)$ that is a chordless cycle C_p with $4 < p < n$. In such a connected component, every vertex belongs to two distinct P_3's as an endpoint. More precisely, let $v_1, \ldots, v_p$ be the vertices of C_p such that $\{v_i, v_{i+1}\}$ and $\{v_p, v_1\}$ are strong edges with $1 \leq i < p$. Then, for every vertex v_i of C_p there two P_3's v_{i-2}, v_{i-1}, v_i and v_i, v_{i+1}, v_{i+2} such that $v_{i-2} \neq v_{i+2}$. By the strong triadic closure, we know that v_i is adjacent to both v_{i-2} and v_{i+2} in G. Since G is a 4-regular graph, there are no more edges incident to any vertex of C_p. Thus, every vertex of C_p is non-adjacent to any other vertex of $G - C_p$ which contradicts the original connectivity of G. Therefore, either every connected component of $E_S(G)$ is a triangle or $E_S(G)$ is connected and $E_S(G) = C_n$.

If every connected component of $E_S(G)$ is a triangle then clearly $E_S(G)$ spans a cluster graph. Suppose that $E_S(G) = C_n$. Since $n = 3q$, we can partition the vertices of C_n into q triangles with the same number of strong edges as follows. For every triplet of vertices v_i, v_{i+1}, v_{i+2}, $1 \leq i \leq n-2$, we further label the edge $\{v_i, v_{i+2}\}$ strong and we label both edges $\{v_{i+2}, v_{i+3}\}$ and $\{v_n, v_1\}$ weak. Observe that $\{v_i, v_{i+2}\}$ is an edge of G, since both

$\{v_i, v_{i+1}\}, \{v_{i+1}, v_{i+2}\}$ are strong edges. Such a labeling satisfies the strong triadic closure property and maintain the same number of strong edges. Therefore in every case a solution for MaxSTC with n edges can be equivalently transformed into a solution for Cluster Deletion with n edges. □

We can also obtain lower bounds for the running time of MaxSTC with respect to the integer k (size of the solution) or the number of vertices n. For that purpose, we make use of the *exponential-time hypothesis*: it states that k-SAT, $k \geq 3$, cannot be solved in time $2^{o(n)}$ or $2^{o(m)}$ where n is the number of variables and m is the number of clauses in the given k-CNF formula (see for e.g., [12, 19, 23]). In this context, algorithms with running time $2^{o(p)}$ for some parameter p are called *subexponential-time* algorithms.

A subexponential-time algorithm for MaxSTC would imply an algorithm for solving Cluster Deletion that has running time subexponential in the size of the solution k or the number of vertices n. However, Cluster Deletion does not admit such subexponential-time algorithms even if we restrict to graphs of maximum degree four [15]. Since we can reduce Cluster Deletion to MaxSTC instances on the same graph with $k = n$, we arrive at the following.

Corollary 4.2. MaxSTC *cannot be solved in $2^{o(k)} \cdot poly(n)$ time or in $O(2^{o(n)})$ time unless the exponential-time hypothesis fails.*

Due to Proposition 3.7, we stress that MaxSTC reduces to finding a minimum vertex cover of $\Gamma(G)$ corresponding to the weak edges in an optimal solution. Thus MaxSTC admits algorithms with running times $2^{\Omega(k)}$ poly(n) or $O^*(c^n)$[1] where k is the minimum number of weak edges and $c < 2$ is a constant [6, 8].

Now let us show that if we restrict to graphs of maximum degree three then MaxSTC becomes polynomial-time solvable. Our goal is to show that there is an optimal solution for MaxSTC that is a cluster graph, since Cluster Deletion is solved in polynomial time on such graphs [15].

Theorem 4.3. *Let G be a graph with maximum degree three. Then, there is an optimal solution for* MaxSTC *on G that is a cluster graph.*

Proof. Observe that if there is a K_4 in G then the vertices of the K_4 form a connected component in G since no vertex can have degree more than three. Let $E_S(G)$ be the graph spanned by the strong edges in an optimal solution for MaxSTC. For a vertex v, we denote by $N_S(v)$ the strong neighbors of v. Clearly $|N_S(v)| \leq 3$. If $|N_S(v)| = 3$ then the vertices of $N[v]$ form a K_4 since the strong neighbors of v are adjacent in G, which implies that all edges of $G[N[v]]$ are strong. In what follows we assume that for every vertex v, $|N_S(v)| \leq 2$ holds.

If every connected component of $E_S(G)$ is a clique then $E_S(G)$ is a cluster graph. Assume that there is a connected component of $E_S(G)$ that is not a clique. Then, there is a $P_3 = x, y, z$ in $E_S(G)$ so that $\{x, y\}$ and $\{y, z\}$ are strong edges. Notice that y has no other strong neighbor in $E_S(G)$. We distinguish cases according to the strong neighbors of x and z.

- Let $N_S(x) = \{y\}$ and $N_S(z) = \{y\}$. Observe that $\{x, z\}$ is an edge of G by the strong triadic closure. Then, we reach a contradiction to the optimality of $E_S(G)$ since labeling the edge $\{x, z\}$ as strong does not violate the strong triadic closure.

- Let $N_S(x) = \{x', y\}$ and $N_S(z) = \{y\}$. Then, observe that the edge $\{y, x'\}$ is weak. We show that we can label the edge $\{y, x'\}$ as strong and the edge $\{y, z\}$ as weak without

[1] The O^* notation suppresses polynomial factors of n.

violating the strong triadic closure. Assume for contradiction that labeling the edge $\{y, x'\}$ as strong violates the strong triadic closure. Then, since there is no other strong edge incident to y, there is a strong edge $\{x', a\}$. Since $N_S(z) = \{y\}$, $a \neq z$. Then, however, we reach a contradiction to the degree of x since x is adjacent to a, x', y, and z in G. Hence, we can safely label the $\{y, x'\}$ as strong so that x, y, x' does not induce a P_3 in $E_S(G)$.

- Let $N_S(x) = \{x', y\}$ and $N_S(z) = \{z', y\}$. If $x' \neq z'$ then y must be adjacent in G to all four vertices x, z, x', z' which contradicts its degree in G. Thus $x' = z'$ and the vertices x, y, z, x' induce a K_4 in G by the strong triadic closure. This means that all edges of the K_4 are strong.

Since $|N_S(x)| \leq 2$ and $|N_S(z)| \leq 2$, we have considered all cases for the strong neighbors of x and z. Thus, we can reform the solution $E_S(G)$ for MaxSTC into a union of cliques and keep the same size. Therefore, there is a solution for Cluster Deletion having the same size with an optimal solution for MaxSTC. □

By combining Theorem 4.3 with the fact that Cluster Deletion can be solved in $O(n^{1.5} \cdot \log^2 n)$ on graphs with maximum degree three [15], we get the following result.

Corollary 4.4. MaxSTC *can be solved in* $O(n^{1.5} \cdot \log^2 n)$ *time when the input graph has maximum degree three.*

5 Concluding Remarks

We have performed a systematic study on families of graphs for which optimal solutions for both MaxSTC and Cluster Deletion problems coincide. As an important outcome, we have complemented previous results regarding the complexity of MaxSTC when restricted to cographs or graphs of bounded degree. Some open questions arise from our work. It is interesting to completely characterize graphs by forbidden subgraphs for which MaxSTC and Cluster Deletion solutions coincide. Towards such an approach, Proposition 3.7 seems to be a useful tool. Moreover, despite the fact that the optimal solutions for MaxSTC and Cluster Deletion coincide on cographs, there exist superclasses of cographs, namely, permutation graphs (the lowest example of Figure 1 is a permutation graph), for which both problems do not coincide and both problems restricted to such graphs have unresolved complexity status. Settling the complexity of both problems on superclasses of cographs consists an interesting research area for future work.

A more general and realistic scenario for both problems is to restrict the choice of the considered edges. Assume that a subset F of edges is required to be included in the same clusters for Cluster Deletion or those edges are required to be strong for MaxSTC. 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 the edges of $E' \cup F$ span a cluster graph or satisfy the strong triadic closure. Clarifying the complexity of such generalized problems is interesting on graphs for which Cluster Deletion or MaxSTC are solved in polynomial time.

References

[1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

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

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

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

[5] D.G. Corneil, Y. Perl, and L.K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14:926–934, 1985.

[6] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

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

[8] F.V. Fomin and D. Kratsch. *Exact Exponential Algorithms*. Springer, 2010.

[9] Y. Gao, D.R. Hare, and J. Nastos. The cluster deletion problem for cographs. *Discrete Mathematics*, 313:2763–2771, 2013.

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

[11] W.-K. Hon, T. Kloks, H.H. Liu, S.-H. Poon, and Y.-L. Wang. On independence domination. In *Proceedings of FCT 2013*, pages 183–194, 2013.

[12] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63:512–530, 2001.

[13] W. Imrich, S. Klavzar, and D.F. Rall. *Topics in Graph Theory: Graphs and Their Cartesian Product*. AK Peters Ltd, 2008.

[14] P.K. Jha and G. Slutzki. Independence numbers of product graphs. *Applied Mathematics Letters*, 7:91–94, 1994.

[15] C. Komusiewicz and J. Uhlmann. Cluster editing with locally bounded modifications. *Discrete Applied Mathematics*, 160:2259–2270, 2012.

[16] A.L. Konstantinidis, S.D. Nikolopoulos, and C. Papadopoulos. Strong triadic closure in cographs and graphs of low maximum degree. In *COCOON 2017*, pages 346–358, 2017.

[17] A.L. Konstantinidis and C. Papadopoulos. Maximizing the strong triadic closure in split graphs and proper interval graphs. In *ISAAC 2017*, pages 53:1–53:12, 2017.

[18] V.B. Le. Gallai graphs and anti-gallai graphs. *SIAM J. Disc. Math.*, 159:179–189, 1996.

[19] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the European Association for Theoretical Computer Science*, 105:41–72, 2011.

[20] R. Shamir and R. Sharan. A fully dynamic algorithm for modular decomposition and recognition of cographs. *Discrete Applied Mathematics*, 136:329–340, 2004.

[21] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144:173–182, 2004.

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

[23] G.J. Woeginger. *Exact algorithms for NP-hard problems: a survey*, volume 2570. Lecture Notes in Computer Science, Springer, 2003.