Theoretical Computer Science 740 (2018) 76–84

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Strong triadic closure in cographs and graphs of low maximum degree

Athanasios L. Konstantinidis [a], Stavros D. Nikolopoulos [b], Charis Papadopoulos [a,*]

[a] Department of Mathematics, University of Ioannina, Greece
[b] Department of Computer Science & Engineering, University of Ioannina, Greece

A R T I C L E   I N F O

Article history:
Received 30 July 2017
Received in revised form 26 April 2018
Accepted 10 May 2018
Available online 18 May 2018
Communicated by V.Th. Paschos

Keywords:
Strong triadic closure
Cluster deletion
Cographs
Bounded degree graphs
NP-completeness
Polynomial-time algorithm

A B S T R A C T

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.

* Corresponding author.
E-mail addresses: skonstan@cc.uoi.gr (A.L. Konstantinidis), stavros@cs.uoi.gr (S.D. Nikolopoulos), charis@cs.uoi.gr (C. Papadopoulos).

https://doi.org/10.1016/j.tcs.2018.05.012
0304-3975/

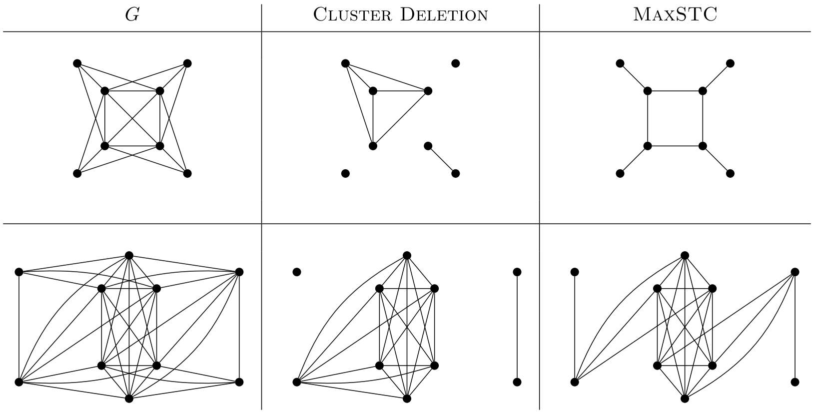

Fig. 1. Two examples of graphs with their corresponding optimal solutions for Cluster Deletion and MaxSTC, respectively. For the MaxSTC problem the edges of G that are not drawn in the solution correspond to the weak edges.

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 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., Fig. 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*