A constant approximation algorithm for the densest k-subgraph problem on chordal graphs

Maria Liazi [a,*,1], Ioannis Milis [b], Vassilis Zissimopoulos [a]

[a] *Department of Informatics and Telecommunications, University of Athens, 157 84 Athens, Greece*

[b] *Department of Informatics, Athens University of Economics and Business, 104 34 Athens, Greece*

Abstract

The Densest k-Subgraph (DkS) problem asks for a k-vertex subgraph of a given graph with the maximum number of edges. The DkS problem is NP-hard even for special graph classes including bipartite, planar, comparability and chordal graphs, while no constant approximation algorithm is known for any of these classes. In this paper we present a 3-approximation algorithm for the class of chordal graphs. The analysis of our algorithm is based on a graph theoretic lemma of independent interest.

Key words: Densest k-subgraph, Chordal graphs, Approximation algorithm

1 Introduction

In the *Densest k-subgraph* (DkS) problem we are given a graph $G = (V, E)$, $|V| = n$, and an integer $k \leq n$, and we ask for a subgraph of G induced by exactly k of its vertices such that the number of edges of this subgraph is maximized. The problem is directly NP-hard as a generalization of the well known *Maximum Clique* problem. In the weighted version of the DkS we are

* Corresponding author.
Email addresses: mliazi@di.uoa.gr (Maria Liazi), milis@aueb.gr (Ioannis Milis), vassilis@di.uoa.gr (Vassilis Zissimopoulos).

[1] The project is co-financed within Op. Education by the ESF (European Social Fund) and National Resources.

also given non negative weights on the edges of G and the goal is to find a k-vertex induced subgraph of maximum total edge weight.

During last years a large body of work has been concentrated on the design of approximation algorithms for both the DkS problem and its weighted version, based on a variety of techniques including greedy algorithms, LP relaxations and semidefinite programming. For a brief presentation of this body of work the reader is referred to [3] and the references therein. The best known approximation ratio for the DkS problem, which performs well for all values of k, is $O(n^{\delta})$, for some $\delta < \frac{1}{3}$ [6], while a simple greedy algorithm in [2] achieves an approximation ratio of $O(\frac{n}{k})$ even for the weighted version of the problem. On the other hand, it has been shown that the DkS problem does not admit a polynomial time approximation scheme (PTAS) [13]. However, there is not a negative result that achieving an approximation ratio of $O(n^{\epsilon})$, for some $\epsilon > 0$, is NP-hard. Concerning approximation algorithms for special cases of the problem it is known that the DkS problem admits a PTAS for graphs of minimum degree $\Omega(n)$ as well as for dense graphs (of $\Omega(n^2)$ edges) when k is $\Omega(n)$ [1]. Moreover, algorithms achieving approximation factors of 4 [17] and 2 [11] have been proposed for the weighted DkS problem on complete graphs where the weights satisfy the triangle inequality.

The DkS problem is trivial on trees (any subtree of k vertices contains exactly $k-1$ edges). It is also known that DkS is polynomial for graphs of maximal degree two [7] as well as for cographs, split graphs and k-trees [4]. On the other hand the DkS problem remains NP-hard for bipartite graphs [4], even of maximum degree three [7], as well as for comparability graphs, chordal graphs [4] and planar graphs [12]. The weighted version of the DkS problem is polynomial on trees either if we ask for a connected solution [10,14,15] or for a disconnected one [16]. In fact, the result for the latter case is implied by a result for the solution of the quadratic 0-1 knapsack problem on edge series-parallel graphs in [16].

In the next section we introduce the reader to the class of chordal graphs and their properties and we give our notation. In Section 3 we present the approximation algorithm and the lemmas yielding our approximation ratio. We conclude in Section 4.

2 Definitions and Notation

A *clique* of an undirected graph, $G = (V, E)$, is a subset of its vertices, $C \subseteq V$, inducing a complete subgraph in G. The size $|C|$ of a clique is the number of its vertices. A *maximal clique* is a clique, which is not contained in any larger clique. A largest maximal clique is called *maximum clique*. A vertex of a graph

G is called *simplicial* if its adjacent vertices induce a complete subgraph in G. An order $\langle u_1, u_2, \ldots, u_n \rangle$ of the vertices of G, is called *perfect elimination order* if each u_i is a simplicial vertex of the subgraph of G induced by the vertices $\{u_i, u_{i+1}, \ldots, u_n\}$.

A graph is called *chordal* if every cycle of length strictly greater than three possesses a chord, that is, an edge joining two nonconsecutive vertices of the cycle. In the rest of this paper by $G = (V, E)$ we denote a chordal graph. It is well known that for a chordal graph, $G = (V, E)$, the following hold:
(i) G has at most $m \leq |V|$ maximal cliques, $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$, which can be found in polynomial time [9].
(ii) G has a simplicial vertex. Actually, if G is not a clique, then it has two nonadjacent simplicial vertices [5].
(iii) G has a perfect elimination order. Moreover, any simplicial vertex can start such an order [8].

By G_A we denote the subgraph of G induced by a subset $A \subseteq V$ of its vertices and by G^F we denote the subgraph of G induced by a subset $F \subseteq E$ of its edges. A direct consequence of the definition of the class of chordal graphs is that being chordal is a hereditary property inherited by every vertex-induced subgraph G_A of G, but not by every edge-induced subgraph G^F of G. It is also obvious that for every maximal clique C_i of a vertex-induced or an edge-induced subgraph of G, there is at least one maximal clique C_j of G such that $C_i \subseteq C_j$.

By $E(A)$ we denote the set of edges in a subgraph G_A of G, while by $E(A, B)$ we denote the set of edges between two disjoint subsets $A, B \subseteq V$ of vertices of G i.e., the set of edges with one of their endpoints in A and the other in B.

By S we denote a solution to the DkS problem, that is a subset $S \subseteq V$ such that $|S| = k$, while by S^* we denote an optimal solution, that is a solution S for which $|E(S)|$ is maximized.

Finally, we assume that $k > |C_i|,\ 1 \leq i \leq m$, for otherwise S^* consists of any subset of k vertices of some clique for which $|C_i| \geq k$.

3 An approximation algorithm for the DkS problem on chordal graphs

Since all the maximal cliques of a chordal graph $G = (V, E)$ can be found in polynomial time it is natural to study the DkS problem on those maximal cliques instead on G itself. In this section we analyze the following simple greedy algorithm for finding an approximate solution to the DkS problem on a chordal graph G.

Greedy Algorithm:
1. Let $C_1, C_2, \ldots C_m$ be the maximal cliques of G, sorted in non-increasing order of their sizes.
2. Find the largest integer t such that $k > |\bigcup_{i=1}^{t-1} C_i| = k'$.
3. Return the solution S consisting of all the vertices of the cliques $C_1, C_2, \ldots, C_{t-1}$ plus $k - k' > 0$ vertices of clique C_t.

The size of the maximal clique C_t plays a crucial role in our analysis and it will be denoted by $L = |C_t|$.

We first obtain a lower bound on the number of edges $|E(S)|$ in the solution S derived by the *Greedy Algorithm* for a chordal graph G. This bound is obtained by relating the solution S to the solution that the *Greedy Algorithm* returns for a graph consisting of independent cliques of size L. Formally, for a chordal graph G and the parameter L defined above we consider the chordal graph $\widetilde{G}$ consisting of at least $\lceil k/L \rceil$ *independent* cliques all of size L.

Lemma 1 *Let S and $\widetilde{S}$ be the solutions that the Greedy Algorithm returns for the DkS problem on graphs G and $\widetilde{G}$, respectively. It holds that $|E(S)| \geq |E(\widetilde{S})| = \frac{k(L-1)-b(L-b)}{2}$, where $b = k \bmod L$.*

Proof: To prove the bound of the lemma we consider also the solutions S' and $\widetilde{S}'$ that the Greedy Algorithm returns for the Densest k'-Subgraph (Dk'S) problem on graphs G and $\widetilde{G}$, respectively. Recall that $k' = |\bigcup_{i=1}^{t-1} C_i|$.

Consider first the solutions S and S'. The solution S consists of the $t-1$ largest maximal cliques of G, of k' vertices, plus a set $A \subseteq C_t$, of $a = k - k'$ vertices, such that $\bigcup_{i=1}^{t-1} C_i \cap A = \emptyset$. Obviously, $0 < a \leq |C_t| = L$. Moreover, $S \setminus S' = A$ and therefore,

$$|E(S)| \geq |E(S')| + \frac{a(a-1)}{2}.$$

Consider next the solutions S' and $\widetilde{S}'$. The solution S' consists of exactly the $t-1$ largest maximal cliques of G, of k' vertices. As all the maximal cliques of S' are of size at least L, it follows that all the vertices in S' have degree at least $L-1$. The solution $\widetilde{S}'$ consists of $q' = \lfloor k'/L \rfloor$ independent cliques of size L and one more independent clique, B', of size $b' = k' \bmod L$, that is $|\widetilde{S}'| = q'L + b'$. Each one of the b' vertices in the clique B' of $\widetilde{S}'$ has degree $b'-1$. Hence, the solution S' contains at least $b'(L-1-(b'-1)) = b'(L-b')$ more edges than the solution $\widetilde{S}'$, that is $|E(S')| \geq |E(\widetilde{S}')| + b'(L-b')$. Therefore,

$$|E(S)| \geq |E(\widetilde{S}')| + b'(L-b') + \frac{a(a-1)}{2}.$$

Consider finally the solutions $\widetilde{S}'$ and $\widetilde{S}$. The solution $\widetilde{S}$ consists of $q = \lfloor k/L \rfloor$

independent cliques of size L and one more independent clique, B, of size $b = k \bmod L$, that is $|\widetilde{S}| = qL + b$. Moreover, $|\widetilde{S}| - |\widetilde{S}'| = a$.
If $a \leq b$, then the solution $\widetilde{S}'$ can be obtained from the $\widetilde{S}$, by removing a vertices of the clique B of $\widetilde{S}$. In this case $|\widetilde{S}'| = qL + b'$, where $b' = b - a$, and $|E(\widetilde{S})| = |E(\widetilde{S}')| + \frac{a(a-1)}{2} + b'a$. Therefore,

$$|E(S)| \geq |E(\widetilde{S})| - b'a + b'(L - b') = |E(\widetilde{S})| + b'(L - b),$$

and since $L > b$, the inequality of the lemma follows.
If $a > b$, then the solution $\widetilde{S}'$ can be obtained from the $\widetilde{S}$, by removing all the b vertices of the clique B and $x = a - b$ vertices of the $q - th$ clique of size L of $\widetilde{S}$. In this case $|\widetilde{S}'| = (q-1)L + b'$, where $a - b = L - b'$, and $|E(\widetilde{S})| = |E(\widetilde{S}')| + \frac{b(b-1)}{2} + \frac{x(x-1)}{2} + xb' = |E(\widetilde{S}')| + \frac{a(a-1)}{2} - xb + xb' = |E(\widetilde{S}')| + \frac{a(a-1)}{2} + (a-b)(b'-b)$. Therefore,

$$|E(S)| \geq |E(\widetilde{S})| - (a-b)(b'-b) + b'(L-b') = |E(\widetilde{S})| + (L-b')b,$$

and since $L > b'$ the inequality of the lemma follows.

The solution $\widetilde{S}$ consists of $\frac{k-b}{L}$ full cliques plus $b < L$ vertices from the clique B. Hence, the number of edges in $\widetilde{S}$ is

$$|E(\widetilde{S})| = \frac{k-b}{L} \cdot \frac{L(L-1)}{2} + \frac{b(b-1)}{2} = \frac{k(L-1) - b(L-b)}{2}. \quad \square$$

Next lemma, which is of independent interest, gives un upper bound on the number of edges of a chordal graph as a function of the size of its maximum clique.

Lemma 2 *Let $c \geq 2$ be the size of a maximum clique of a chordal graph $G = (V, E)$. It holds that $|E| \leq (c-1)(|V| - \frac{c}{2})$ and this bound is the best possible.*

Proof: The graph G, as a chordal one, has a perfect elimination order. We remove from G vertices (and their incident edges) in a perfect elimination order until the remaining number of its vertices is c.

Since the size of a maximum clique of G is c, each removed vertex has degree at most $c-1$. Thus, the number of the edges removed during this process, let $|E_1|$, is at most $(|V| - c)(c-1)$. Moreover, the remaining number of edges, let $|E_2|$, is at most $\frac{c(c-1)}{2}$ (when the remaining c vertices form a clique).

Therefore, it follows that

$$|E| = |E_1| + |E_2| \leq (|V| - c)(c-1) + \frac{c(c-1)}{2} = (c-1)(|V| - \frac{c}{2}).$$

To prove that this bound is the best possible consider the chordal graph $G = (V, E)$ consisting of a clique, C, of size $c-1$ and $|V| - c + 1$ independent vertices each one of them adjacent to all vertices of C. Observe that a maximum clique of G consists of the clique C plus one of the independent vertices, and it is of size c. For this graph G it holds that $|E| = \frac{(c-1)(c-2)}{2} + (|V|-c+1)(c-1) = (c-1)(|V| - \frac{c}{2})$. Note that if $c = |V|$, then G becomes a complete graph. □

Let us now relate the solution S of the *Greedy Algorithm* to an optimal solution S^* to the DkS problem on a chordal graph G. Let $S^* = A \cup B$, where $A = S^* \cap S$ is the subset of vertices of S^* that belong to S and $B = S^* \setminus A$ is the subset of vertices of S^* that do not belong to S. Let also $\Gamma \subseteq A$ be the subset of vertices in A that have adjacent vertices in B and $F = E(B) \cup E(\Gamma, B)$. Obviously, $\Gamma \cap B = \emptyset$ and $|E(S^*)| = |E(A)| + |E(B)| + |E(\Gamma, B)| = |E(A)| + |F|$.

In order to bound the number of edges in an optimal solution S^* we shall consider the edge-induced subgraph $G^F = (\Gamma \cup B, F)$ as well as the vertex-induced subgraph $G_{B\cup\Gamma}$ of G. Note that $G_{B\cup\Gamma}$, as a vertex-induced subgraph of G, is a chordal graph, while G^F, as an edge-induced subgraph of G, is in general a non chordal one. Next proposition gives a useful structural property of the subgraph G^F.

Proposition 1 *All the maximal cliques of the graph $G^F = (\Gamma \cup B, F)$ are of size at most L.*

Proof: The solution S of the *Greedy Algorithm* contains vertices from maximal cliques of G of size at least L and the vertices in B do not belong to S. Therefore, the vertices in B belong to maximal cliques of G of size at most L.

Consider first the set B of vertices of the subgraph G^F. Since they belong to maximal cliques of G of size at most L, they also form in G^F maximal cliques of size at most L.
Consider next the set Γ of vertices of the subgraph G^F. By the definition of the subgraph G^F, it follows that: a) they are independent in G^F, and b) every one of them has at least one adjacent vertex in B. Assume that such vertex belongs to a maximal clique, K, of G^F of size greater than L. As the vertices of Γ are independent in G^F, it follows that at least one vertex of B belongs also in such a clique K of G^F, and, therefore, to a maximal clique of G of size greater than L, a contradiction to the definition of the set B. Hence the vertices of Γ belong to maximal cliques of G^F of size at most L.
Therefore, all maximal cliques of the graph G^F are of size at most L. □

Using Proposition 1 we can now prove that the bound of Lemma 2 holds also for the graph G^F.

Lemma 3 *For the edge-induced graph $G^F = (\Gamma \cup B, F)$ of G it holds that $|F| \leq (L-1)(|\Gamma \cup B| - \frac{L}{2})$.*

Proof:

We work analogously as in Lemma 2. The graph $G_{B\cup\Gamma}$, as a vertex-induced subgraph of G, is a chordal one and it has a perfect elimination order. Following this order (of the graph $G_{B\cup\Gamma}$), we remove vertices from the graph G^F until the remaining number of its vertices is L.

By Proposition 1 the size of a maximum clique of G^F is L and, hence, the degree, in G^F, of each removed vertex is at most $L-1$. Thus, the number of the removed edges of G^F during this process, let $|E_1|$, is at most $(|B\cup\Gamma|-L)(L-1)$. Moreover, the remaining edges of G^F, let $|E_2|$, is at most $\frac{L(L-1)}{2}$ (when the remaining L vertices form a clique).

Therefore, it follows that

$$|E| = |E_1| + |E_2| \leq (|B \cup \Gamma| - L)(L-1) + \frac{L(L-1)}{2} = (L-1)(|B \cup \Gamma| - \frac{L}{2}).$$ □

Applying Lemma 3 on G^F, (with $|\Gamma \cup B| \leq |S^*| = k$) we obtain

$$|F| = |E(B)| + |E(\Gamma, B)| \leq (L-1)(k - \frac{L}{2}).$$

For the edges $E(A)$ it holds that $|E(A)| \leq |E(S)|$, since $A \subseteq S$. We also know, by Lemma 1, that

$$|E(S)| \geq \frac{k(L-1) - b(L-b)}{2}.$$

Therefore,

$$\frac{|E(S^*)|}{|E(S)|} = \frac{|E(A)| + |F|}{|E(S)|} \leq 1 + \frac{|F|}{|E(S)|} \leq 1 + \frac{(L-1)(2k-L)}{k(L-1) - b(L-b)}.$$

By recalling that $b = k \bmod L \leq L-1$ and by distinguishing between two cases for b ($b \leq L/2$ and $b > L/2$) it is easy to prove that $\frac{(L-1)(2k-L)}{k(L-1)-b(L-b)} \leq 2$. Thus the next theorem follows.

Theorem 1 *There is a 3-approximation algorithm for the DkS problem on chordal graphs.*

4 Concluding remarks

We have shown a 3-approximation algorithm for the DkS problem on chordal graphs, which, up to our knowledge, is the first constant approximation algorithm for an NP-hard variant of the problem on non-dense graphs. Concerning the tightness of our analysis we succeeded to construct counterexamples for which our algorithm gives a solution of at least half of the edges of an optimal one.

On the other hand many questions concerning the frontier between hard and polynomial solvable or approximable, within a constant ratio, variants of the DkS problem, remain open. Such an outstanding open question concerns the complexity of the DkS problem on interval graphs or even on proper interval graphs. The existence of a constant approximation algorithm for the NP-hard variant of the DkS problem on planar graphs is another interesting open question.

Acknowledgment. The crucial suggestions of an anonymous referee for the simplification of the proofs of Lemmas 2 and 3 are gratefully acknowledged.

References

[1] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proceedings of the 27th annual ACM symposium on Theory of Computing*, pages 284–293, 1995.

[2] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.

[3] A. Billionnet and F. Roupin. A deterministic algorithm for the densest k-subgraph problem using linear programming. Technical Report No486, CEDRIC, CNAM-IIE, Paris, 2004.

[4] D. G. Corneil and Y. Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9:27–39, 1984.

[5] G. A. Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25:71–76, 1961.

[6] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[7] U. Feige and M. Seltser. On the densest k-subgraph problem. Technical Report CS97-16, Weizmann Institute, 1997.

[8] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[9] F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.

[10] O. Goldschmidt and D. Hochbaum. k-edge subgraph problems. *Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 74(2):159–169, 1997.

[11] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Operations Research Letters*, 21(3):133–137, 1997.

[12] J. M. Keil and T. B. Brecht. The complexity of clustering in planar graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 9:155–159, 1991.

[13] S. Khot. Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 136–145, 2004.

[14] F. Maffioli. Finding a best subtree of a tree. Technical Report 91.041, Politechnico di Milano, Dipartimento di Elektronica, 1991.

[15] Y. Perl and Y. Shiloach. Efficient optimization of monotonic functions on trees. *SIAM Journal on Algebraic and Discrete Methods*, 4(4):512–516, 1983.

[16] D. J. Jr. Rader and G. J. Woeginger. The quadratic 0-1 knapsack problem with series-parallel support. *Operation Research Letters*, 30(3):159–166, 2002.

[17] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994.