The clique problem on inductive k-independent graphs

George Manoussakis

gomanous@gmail.com

[a]*LIAFA, University Paris 7, 75205 Paris Cedex 13, France.*

Abstract

A graph is inductive k-independent if there exists and ordering of its vertices $v_1, ..., v_n$ such that $\alpha(G[N(v_i) \cap V_i]) \leq k$ where $N(v_i)$ is the neighbourhood of v_i, $V_i = \{v_i, ..., v_n\}$ and α is the independence number. In this article, by answering to a question of [Y.Ye, A.Borodin, Elimination graphs, ACM Trans. Algorithms 8 (2) (2012) 14:1-14:23], we design a polynomial time approximation algorithm with ratio $\overline{\Delta}/log(log(\overline{\Delta})/k)$ for the maximum clique and also show that the decision version of this problem is fixed parameter tractable for this particular family of graphs with complexity $O(1.2127^{(p+k-1)^k} n)$. Then we study a subclass of inductive k-independent graphs, namely k-degenerate graphs. A graph is k-degenerate if there exists an ordering of its vertices $v_1, ..., v_n$ such that $|N(v_i) \cap V_i| \leq k$. Our contribution is an algorithm computing a maximum clique for this class of graphs in time $O(1.2127^k (n - k + 1))$, thus improving previous best results. We also prove some structural properties for inductive k-independent graphs.

1. Introduction and notations

The main contribution of this work is an algorithmic approach of the clique problem for inductive k-independent graphs and k-degenerate graphs.

Inductive k-independent graphs, introduced in [1], are a natural generalisation of chordal graphs and have been studied more extensively in [2]. In these two papers, the authors studied algorithmic and structural properties of this new family of graphs, and showed that several natural classes of graphs are in-

Preprint submitted to Elsevier *December 10, 2014*

ductive k-independent for a small constant k. For instance, chordal graphs are inductive 1-independent, planar graphs are inductive 3-independent and claw-free graphs are inductive 2-independent. They also studied k-approximation algorithms for the maximum independent set and minimum vertex colouring. For the minimum vertex cover they showed that one can achieve a $(2 - \frac{1}{k})$-approximation. Here we will also consider degenerate graphs [3], a subclass of inductive k-independent ones. Roughly speaking, degeneracy is a common measure of the sparseness of a graph. Many real-life graphs are sparse, have low degeneracy, and follow a power-law degree distribution [4]. A graph will be said to be k-degenerate if every induced subgraph has a vertex of degree at most k.

Formally, let $G = (V, E)$ be a graph of n vertices and m edges. If $X \subset V$, the subgraph of G induced by X is denoted by $G[X]$. The set $N(x)$ is called the open neighbourhood of the vertex x. The closed neighbourhood of x is defined as $N[x] = N(x) \cup x$. The order of a maximum independent set in G is denoted by $\alpha(G)$. For a graph G, the graph $\overline{G}$ will be its complement. Given an order $v_1, ..., v_n$ of the vertices of G, V_i is the set of vertices following v_i including itself in this order, that is, the set $\{v_i, ..., v_n\}$. Let G_i^+ denote the induced subgraph $G[N[v_i] \cap V_i]$. By $\Delta(G)$ we will denote the maximum degree of a graph. For a given ordering σ of the vertices of a graph G, let Δ_i be the maximum degree of G_i and let $\Delta_\sigma = max_i(\Delta_i)$. $\overline{\Delta_i}$ is the maximum degree of $\overline{G_i}$ and $\overline{\Delta_\sigma} = max_i(\overline{\Delta_i})$. A k-independence ordering is an ordering of vertices $v_1, v_2, ..., v_n$ such that for any v_i, $1 \leq i \leq n$, $\alpha(G[N(v_i) \cap V_i] \leq k$. The minimum of such k over all orderings is called the inductive independence number, which we denote as $\lambda(G)$. A graph G is inductive k-independent if $\lambda(G) \leq k$. A graph is k-degenerate if there is an ordering $v_1, ..., v_n$ of its vertices such that v_i, $1 \leq i \leq n$, $|N(v_i) \cap V_i| \leq k$.

The paper is organised as follows. In Section 2 we establish some properties of cliques in general graphs with any orderings of vertices. In Section 3 we present the FPT and approximation algorithms for inductive k-independent graphs. In Section 4 we prove our algorithm for the maximum clique of k-degenerate graphs. Finally, in Section 5 we will present some miscellaneous results for inductive k-independent graphs.

2. General results for the clique problem

In this section we give some general results concerning maximal cliques related to orderings of vertices in general graphs.

Lemma 1. *Let G be a graph and let $v_1, ..., v_n$ be any ordering of its vertices. Then every maximal clique of G belongs to exactly one subgraph G_i^+.*

PROOF. Let K be any maximal clique of G. Consider the vertex v_i of K that appears first in the ordering. It is clear that K belongs to G_i^+. Now we claim that K can not belong to another subgraph G_j^+ for $j \neq i$. To show that assume by contradiction that K appears in two induced subgraphs, say G_i^+ and G_j^+, with $i < j$. By the maximality property of K we observe that necessarily $v_i, v_j \in K$. Therefore we have a contradiction since $v_i \notin V(G_j^+)$.

Lemma 2. *Let G be a graph. If G has an ordering of its vertices such that for each i, the maximum clique of G_i^+ can be found in time $O(T_i)$, then the maximum clique of G can be found in time $O(\sum_{k=1}^{n} T_k)$.*

PROOF. Let G be a graph. Suppose it has an ordering of its vertices $v_1, ...v_n$ such that, for all i, the maximum clique of G_i^+ can be found in time $O(T_i)$. For each vertex of the order compute the maximum clique in its neighbourhood. This can be done in time $O(\sum_{k=1}^{n} T_k)$. Then, among these cliques return the largest one. By Lemma 1, this clique is the maximum one of G and the algorithm runs in time $O(\sum_{k=1}^{n} T_k)$.

Lemma 3. *Let G be a graph and $\beta_1, ..., \beta_n \in \mathbb{R}$. Let $\beta = max_i(\beta_i)$. Assume that G has an ordering of its vertices such that for all i, there is an β_i-approximation algorithm for the maximum clique of G_i^+ running in time $O(T_i)$. Then there is an β-approximation algorithm for the maximum clique of G running in time $O(\sum_{k=1}^{n} (T_k))$.*

Proof. For every graph G_i^+ compute an β_i-approximation of its maximum clique K_i, call it K_i'. This will take $O(\sum_{k=1}^{n}(T_k))$ time. Among these approximated cliques return the largest one, say K'. By Lemma 1, there exists $j \in \{1, ..., n\}$ such that K_j is the maximum clique of G. By definition $|K'| \geq |K_j'|$. Therefore K' is indeed an approximation of the maximum clique of G. Concerning the ratio of this approximation we must consider the worst case which arises when the maximum clique of G appears in a graph G_l^+ where the existing approximation algorithm achieves the largest ratio β. Since $|K'| \geq |K_l'|$ then we achieve an β-approximation in the worst case. In conclusion the algorithm achieves a β-approximation in time $O(\sum_{k=1}^{n}(T_k))$.

Corollary 4. *Let G be a graph. Assume G has an ordering of its vertices such that for all i, there is an β-approximation algorithm for the maximum clique of G_i^+ running in time $O(T)$. Then there is an β-approximation algorithm for the maximum clique of G running in time $O(nT)$.*

3. The clique problem for inductive k-independent graphs

In [2], the authors ask if there is any approximation for the maximum clique problem for inductive independent graphs. Using the general results of Section 2 in connection with Theorem 5 below, we prove an approximation algorithm of ratio $\overline{\Delta}/log(log(\overline{\Delta})/k)$. (see Corollary 7).

Theorem 5. *[5] Given a K_k-free graph G, there is a polynomial time algorithm that achieves a $\Delta/log(log(\Delta)/(k+1))$ approximation for maximum independent set.*

Theorem 6. *Given an inductive k-independent graph and a k-inductive ordering of its vertices, there is a polynomial time algorithm that achieves a $\overline{\Delta_\sigma}/log(log(\overline{\Delta_\sigma})/(k+1))$ approximation for maximum clique.*

Proof. By definition we have that for each i, $\alpha(G_i^+) \leq k$. Therefore every graph $\overline{G_i}$ is $(k+1)$-clique free. By Theorem 5 we can approximate the maximum independent set of every $\overline{G_i^+}$ with ratio $\overline{\Delta_\sigma}/log(log(\overline{\Delta_\sigma})/(k+1))$ in polynomial time. Thus, this gives an approximation for the maximum clique of every G_i^+ with same ratio. Now by Corollary 4 we have the claimed result.

Corollary 7. *Given an inductive k-independent graph and a k-inductive ordering of its vertices, there is a polynomial time algorithm that achieves a $\overline{\Delta}/log(log(\overline{\Delta})/k)$ approximation for the maximum clique.*

Proof. Straightforward from the fact that $\overline{\Delta_\sigma} \leq \Delta(\overline{G})$.

Observe that, since the complementary of a K_k-free graph is inductive k-independent, if we had a better approximation ratio, we would automatically improve Theorem 5. Now for inductive 2-independent graphs we can achieve a better approximation ratio thanks to Theorem 8.

Theorem 8. *[6] Given a K_3-free graph G, there is a polynomial time algorithm that achieves a $(\Delta-1)^2/(\Delta ln(\Delta)-\Delta+1)$ approximation for the maximum independent set.*

Theorem 9. *Given an inductive 2-independent graph and a 2-inductive ordering of its vertices, there is a polynomial time algorithm that achieves a $(\overline{\Delta}-1)^2/(\overline{\Delta} ln(\overline{\Delta})-\overline{\Delta}+1)$ approximation for the maximum clique.*

Proof. Same as for Theorem 5.

Now we show that the decision version of the CLIQUE problem is FPT when parametrized by the size p of the solution on inductive k-independent graphs:

Theorem 10. *Given an inductive k-independent graph and a k-inductive ordering of its vertices, there is a fixed-parameter tractable algorithm solving the CLIQUE problem in $O(1.2127^{(p+k-1)^k} n)$ time where p is the size of the problem.*

Proof. Ramsey's theorem states that for any two positive integers i, c there exists a positive integer $R(i, c)$ such that any graph with at least $R(i, c)$ vertices contains either an independent set on i vertices or a clique on c vertices or both. It is shown in [7] that $R(i,c) \leq \binom{i+c-2}{c-1}$. Thus if $i = k+1$ and $c = p$ then if a graph with at least $\binom{p+k-1}{p-1} = \binom{p+k-1}{k} \leq (p+k-1)^k$ vertices does not contain an independent set of size $k+1$, then it must have a clique on p vertices. Let G be our inductive k-independent graph and let $v_1, ..., v_n$ be its k-independent ordering. We know by definition that for each i, $\alpha(G[N(v_i) \cap V_i]) < k+1$. Therefore, if for some i, $|N(v_i) \cap V_i| \geq (p+k-1)^k$, the neighbourhood of v_i contains a clique on p vertices by Ramsey's theorem. This can be checked in time $O(n)$. Now suppose that for each i, $|N(v_i) \cap V_i| < (p+k-1)^k$. We iterate over the vertices in the k-independence order and check for every vertex if the graph $G[N(v_i) \cap V_i]$ contains a clique of size $p-1$. By Lemma 1, this procedure will return a clique of size p, if any, in G. To achieve this, we can use the algorithm from [8] that will find a maximum clique in a graph of size $(p+k-1)^k$ in time $O(1.2127^{(p+k-1)^k})$. Thus by Lemma 2 we can conclude that the algorithm runs in time $O(1.2127^{(p+k-1)^k} n)$.

4. The clique problem for k-degenerate graphs

There has been some work done on the clique problem (and similar versions) for k-degenerate graphs. In [9], Eppstein and al. design an algorithm to list all maximal cliques of these graphs in $O(kn3^{k/3})$ time. In [10] the authors prove an algorithm finding the maximum clique of a k-degenerate graph with complexity $O(nm+n2^{k/4})$. Here we improve that result. To achieve this, using a degeneracy ordering, we first construct an algorithm to compute $n-k$ subgraphs G_i^+ for $i = 1, ..., n-k$ and the graph G^*_{n-k+1} which is the subgraph induced on the last k vertices of the ordering. Using these subgraphs, we then show how to find the maximum clique of G.

Lemma 11. *Given a k-degenerate graph G, there is an algorithm constructing the induced subgraphs G_i for $i = 1, ..., (n-k)$ and the graph G^*_{n-k+1} in $O((n-k+1)k^3)$ time, using $O(m)$ memory space.*

Proof. Assume that G is represented by its adjacency lists, using therefore $O(m)$ memory space. Degeneracy, along with a degeneracy ordering, can be computed by greedily removing a vertex with smallest degree (and its edges) from the graph until it is empty. The degeneracy ordering is the order in which vertices are removed from the graph and this algorithm can be implemented in $O(m)$ time [11].

Using this degeneracy ordering we construct below the vertex sets of the graphs G_i for $i = 1, ..., (n-k)$ and of the graph G^*_{n-k+1} as follows. Assume that initially all the vertices of G are coloured blue. Consider iteratively, one by one, the first $n-k$ vertices $v_1, v_2, ...v_{n-k}$ of the ordering. At Step i, we start by colouring vertex v_i red. Then, we scan its neighbourhood (using an adjacency list), we skip its red neighbours and put its blue neighbours in $V(G_i)$. This is because if one of its neighbour is red, it means that it appears before it in the ordering and thus should not be put in $V(G_i)$. At the end the $(n-k)$ first iterations put the remaining k vertices in the vertex set $V(G^*_{n-k+1})$. This construction can be done in $O(m)$ time since each iteration takes time proportional to the degree of the vertex we are considering in the order.

Now we construct the edge sets of the graphs (G_i) for $i = 1, ..., n-k$ and of the graph G^*_{n-k+1} as follows. For the vertex sets $V(G_i)$ for $i = 1, ..., (n-k)$ and for $V(G^*_{n-k+1})$ we start by sorting their vertices following the degeneracy ordering in time $O(klog(k))$ for each such set. This takes total time $O((n-k+1)klog(k))$. This will give us, for each vertex $v_1, ..., v_n$, a sorted array $D_i = d_1, ..., d_k$ containing its neighbours coming later in the degeneracy ordering. This takes space $O(nk) = O(m)$ since every such array is at most of size k. Using this structure, we now show how to build the edge sets. Assume that we want to build the edge set of the graph G_i^+. Since v_i is connected to all the vertices of G_i^+, we go through D_i and add all the corresponding edges to the adjacency list of G_i^+, that is, we add v_i to the adjacency lists of every vertex in D_i and add every vertex of D_i in the adjacency list of v_i. Then, for each element d_j for $j = 1, ..., k$ of D_i, we check for every element $d_{j'}$ of D_i with $j' > j$ if it appears in D_{d_j}. If it is the case, we add the corresponding edge. This is done in $O(k^3)$ for all the elements of D_i. Therefore, to build all the graphs (G_i) for $i = 1, ..., n-k$ and of the graph G^*_{n-k+1} we need, overall,

$O((n-k+1)k^3+m) = O((n-k+1)k^3+nk) = O((n-k+1)k^3)$, and $O(m)$ space, as claimed.

Theorem 12. *Given a k-degenerate graph G, there is an algorithm finding a maximum clique in $O(1.2127^k(n-k+1))$ time.*

PROOF. Using Lemma 11, we start by constructing the subgraphs G_i for $i = 1, ..., (n-k)$ and G^*_{n-k+1} in time $O((n-k+1)k^3)$. Then for each of these graphs, since they are of size k, we find their maximum clique K_i in time $O(1.2127^k)$ using for instance the exact algorithm of [8]. Among these computed cliques find the one of largest size, call it K_j. If K_j is an induced subgraph of some graph G_i for i in $\{1, ..., (n-k-1)\}$, then, by Lemma 1, $K_j \cap \{v_i\}$ is a maximum clique of G and we output it. In the other case, assume that K_j is an induced subgraph of G^*_{n-k+1}. There are two cases. First case there is no other maximum clique $K_{j'}$, of same largest size, and which is an induced subgraph of a $G_{j'}$ for j' in $\{1, ..., (n-k)\}$. Then we simply output K_j using again Lemma 1. In the second case, assume that there is a maximum clique $K_{j'}$ of same largest size induced in some graph $G_{j'}$ for a j' in $\{1, ..., (n-k)\}$. We have that $K_{j'} \cap \{j'\}$ is a maximum clique of G (using again Lemma 1) and such that $|K_{j'} \cap \{j'\}| = |K_j| + 1$. Therefore in this case we output $K_{j'} \cap \{j'\}$.

Since the graphs G^+_i for i in $\{n-k+1, ..., n\}$ are induced subgraphs of G^*_{n-k+1}, this procedure will output a correct maximum clique of the graph. Overall this will take time $O(1.2127^k(n-k+1)+k^3(n-k+1)) = O(1.2127^k(n-k+1))$, as claimed.

5. Miscellaneous results

In this section we give some results of general interest. We start by proving that any induced subgraph of an inductive k-independent graph is inductive k-independent. This result was proved in the original paper [2] but there was some mistake in the proof. This is the reason we include a complete proof here.

Lemma 13. [2] *Any induced subgraph of an inductive k-independent graph is an inductive k-independent graph.*

PROOF. By contradiction suppose that $G = (V, E)$ is an inductive k-independent graph but some induced subgraph G_1 of G is not an inductive k-independent graph. This means that for any ordering of the vertices of G_1 there is always a vertex v such that $\alpha(G_v^+) > k$. Now let $v_1, ...v_n$ be the k-independence ordering that G admits and let X be the vertex set of G_1. Let v_i be the first vertex of X that appears in the ordering. Let $G_2 = G[V_i]$, observe that G_1 is an induced subgraph of G_2. Therefore an independent set of G_1 is an independent set of G_2. Since for any order of the vertices of G_1 there is a vertex v such that $\alpha(G_v^+) > k$ we have that $\alpha(G_2[N(v) \cap V_v]) > k$, which contradicts the fact that $v_1, ..., v_n$ is a k-independence ordering. Therefore any induced subgraph of G is also an inductive k-independent graph.

In the rest of the section we give some structural properties of inductive independent graphs which can be of general interest.

Theorem 14. *[12] Let G be a k-connected graph with $n \geq 3$ vertices. If $\alpha(G) \leq k$, then G is Hamiltonian.*

Proposition 15. *Let G be an inductive 2-independent graph and $\sigma = v_1, ..., v_n$ a 2-inductive ordering of its vertices. For every i, either the induced subgraph G_i^+ is Hamiltonian or G_i^+ is covered by at most two cliques.*

PROOF. If for some j the graph G_j^+ has strictly less than 3 vertices then observe that G_j^+ can be covered by at most two cliques and the theorem is true. Therefore consider now only the induced subgraphs with more than three vertices. Let G_i^+ be such an induced subgraph. We have two cases:

1. G_i^+ is k-connected with $k \geq 2$. Since by definition $\alpha(G_i^+) \leq 2$ then by Theorem 15, G_i^+ is Hamiltonian.

2. G_i^+ is k-connected with $k = 1$. By definition we know that v_i is adjacent to every vertex in $V(G_i)$. Therefore removing a vertex in $V(G_i)$ can not disconnect the graph. Thus since the graph G_i^+ is 1-connected and no vertex in $V(G_i)$ can be a cut vertex then necessarily v_i is a cut vertex. This implies that the graph G_i is disconnected and that it has at most two connected components (otherwise it would have an independent set of size greater than two). Call these components G_{i1} and G_{i2}. Since both of these graphs can have an independent set of size at most one, (otherwise G_i would have an independent set of size greater than two) then G_{i1} and G_{i2} are cliques. Thus the theorem holds in that case.

6. Acknowledgement

I am grateful to professor Michel Habib for all his valuable suggestions and comments.

References

[1] K. Akcoglu, J. Aspnes, B. DasGupta, M.-Y. Kao, Opportunity cost algorithms for combinatorial auctions, CoRR cs.CE/0010031.

[2] Y. Ye, A. Borodin, Elimination graphs, ACM Trans. Algorithms 8 (2) (2012) 14:1–14:23.

[3] A. T. W. D. R. Lick, *d*-degenerate graphs, Canad. J. Math. 22 (1970) 1082–1096.

[4] F. Chung, Graph theory in the information age (2009).

[5] M. M. Halldórsson, J. Radhakrishnan, Improved approximations of independent sets in bounded-degree graphs via subgraph removal, Nordic J. of Computing 1 (4) (1994) 475–492.

[6] J. B. Shearer, A note on the independence number of triangle-free graphs, ii, J. Comb. Theory, Ser. B 53 (2) (1991) 300–307.

[7] S. Jukna, Extremal combinatorics - with applications in computer science, Texts in theoretical computer science, Springer, 2001.

[8] N. Bourgeois, B. Escoffier, V. T. Paschos, J. M. M. van Rooij, A bottom-up method and fast algorithms for max independent set, in: Proceedings of the 12th Scandinavian Conference on Algorithm Theory, SWAT'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 62–73.

[9] D. Eppstein, M. Löffler, D. Strash, Listing all maximal cliques in sparse graphs in near-optimal time, CoRR abs/1006.5440.

[10] A. Buchanan, J. L. Walteros, S. Butenko, P. Pardalos, Solving maximum clique in sparse graphs: an $O(nm + n2^{d/4})$ algorithm for d-degenerate graphs, Optimization Letters 8 (5) (2014) 1611–1617.

[11] V. Batagelj, M. Zaversnik, An $O(m)$ algorithm for cores decomposition of networks, CoRR cs.DS/0310049.

[12] V. Chvátal, P. Erdös, A note on hamiltonian circuits, Discrete Mathematics 2 (2) (1972) 111 – 113.