Finding Independent Sets in Unions of Perfect Graphs

Venkatesan T. Chakaravarthy[1], Vinayaka Pandit[1], Sambuddha Roy[1], Yogish Sabharwal[1]

IBM India Research Lab, New Delhi
{vechakra, pvinayak, sambuddha, ysabharwal}@in.ibm.com

ABSTRACT. The maximum independent set problem (MAXIS) on general graphs is known to be NP-hard to approximate within a factor of $n^{1-\epsilon}$, for any $\epsilon > 0$. However, there are many "easy" classes of graphs on which the problem can be solved in polynomial time. In this context, an interesting question is that of computing the maximum independent set in a graph that can be expressed as the union of a small number of graphs from an easy class. The (MAXIS) problem has been studied on unions of interval graphs and chordal graphs. We study the (MAXIS) problem on unions of perfect graphs (which generalize the above two classes). We present an $O(\sqrt{n})$-approximation algorithm when the input graph is the union of two perfect graphs. We also show that the (MAXIS) problem on unions of two comparability graphs (a subclass of perfect graphs) cannot be approximated within any constant factor.

1 Introduction

It is well known that the classical maximum independent set problem (MAXIS) on general graphs is computationally hard, even to approximate. Zuckerman [13] showed that for any $\epsilon > 0$, it is NP-hard to approximate the MAXIS problem within a factor of $n^{1-\epsilon}$, where n is the number of vertices in the input graph. However, there are *easy* families of graphs for which the MAXIS problem can be solved optimally in polynomial time, for example interval graphs. In this context, it is interesting to consider the MAXIS problem on graphs which are unions of a small number graphs from an easy family. In this paper, we study this problem on perfect graphs and its subclass comparability graphs.

Formally, the input consists of a sequence of graphs $G_1 = (V, E_1), G_2 = (V, E_2), \ldots, G_t = (V, E_t)$ defined over the same vertex set V. A subset of vertices $X \subseteq V$ is called a *common independent set* (CIS), if X is an independent set in each graph G_i. Alternatively, a CIS is an independent set in the *union* graph given by $\widehat{G} = (V, \widehat{E})$, where $\widehat{E} = \cup_{i=1}^{t} E_i$. The goal is to find the maximum cardinality CIS. We call this the *maximum common independent set problem* (MAXCIS). For a fixed constant k, the k-MAXCIS is the special case where the number of input graphs is $t = k$. We consider restricted versions of the MAXCIS problem wherein all the input graphs belong to a particular class (or family) of graphs $\mathcal{C}$. This paper deals with the MAXCIS problem on easy classes $\mathcal{C}$, such as interval, chordal, comparability and perfect graphs. We shall also consider the weighted versions of MAXIS and MAXCIS problem, wherein the input includes a function $p : V \rightarrow \mathbf{R}$ that assigns a profit (or weight) $p(u)$ to each vertex u and the goal is to find the maximum profit independent set and CIS, respectively.

Motivated by applications in bioinformatics, scheduling and computational geometry, the MAXCIS problem on interval graphs has been well-studied (see [1]). Bar-Yehuda et al. [1] presented a $2t$-approximation algorithm for the weighted MAXIS problem on the class of t-interval graphs, which includes unions of t intervals graphs. Hermelin and Rawitz [9] generalized their result by presenting a $2t$-approximation algorithm for class of t-subtrees, which includes both $2t$-interval graphs and chordal graphs. Regarding unions of comparability and perfect graphs, prior work deals with certain combinatorial aspects. It is well known that any perfect graph G on n vertices has either an independent set of size $\sqrt{n}$ or a clique of size $\sqrt{n}$. Motivated by applications in computational geometry, Dumitrescu and Tóth [5] studied the same issue on unions of comparability graphs and perfect graphs, from a combinatorial perspective. They showed that if G is the union of two perfect graphs, then G has an independent set of size $n^{1/3}$ or a clique of size $n^{1/3}$. They also provided counter-examples to show that the above bound cannot be improved beyond $n^{0.42}$ (see Theorem 5).

The above discussion shows that approximation algorithms are known for the weigted MAXCIS problem on interval graphs and chordal graphs; both these classes are easy classes for which the weighted MAXIS problem can be solved optimally in polynomial time [7, 6]. The goal of this paper study the MAXCIS problem on comparability graphs and perfect graphs, two other important easy classes for which the MAXIS problem can be solved optimally in polynomial time [12, 8]. Perfect graphs generalize the other classes mentioned above and can be thought of as the pinnacle among easy classes.

Our main result presents an $O(\sqrt{n})$-approximation algorithm for the weighted 2-MAXCIS problem on perfect graphs. The algorithm is obtained by considering a suitable LP formulation. The LP is of exponential size. We solve it using a separation oracle and find the optimal LP solution. We categorize the vertices into multiple groups based on the values assigned to them by the above LP solution. Next, we find a suitably large independent set within each group. The best among these independent sets is the output. We then argue that the output independent set is a $O(\sqrt{n})$ approximation to the optimal solution.

Our next set of results provide some evidence that it may be difficult to significantly improve the $O(\sqrt{n})$-approximation ratio. This includes proving integrality gaps and hardness results.

Let us first briefly discuss the integrality gap results. We show that the above LP has an integrality gap of $\sqrt{n}$, even for the unweighted case and when both the input graphs are comparability graphs. We then consider a powerful strengthening of the LP, wherein a variable $x(u)$ is added for each variable u, which indicates whether $x(u)$ is selected in the independent set or not. Then, for each clique C of $\widehat{G}$, we add the constraint $\sum_{u \in C} x(u) \leq 1$. Such an LP is also considered while designing the polynomial time algorithm for the MAXIS problem on perfect graphs. In the case of MAXCIS, this LP is of exponential size and it is not clear whether it can be solved in polynomial time; it seems difficult to construct separation oracles (even approximate ones). Nevertheless, we show that even this strong LP has an integrality gap of $n^{0.16}$ (even for the unweighted case over comparability graphs). This result is derived as a consequence of a combinatorial result regarding unions of comparability graphs, due to Dumitrescu and Tóth [5] (discussed earlier).

On the hardness front, it is known that the 2-MAXCIS problem on linear forests is APX-hard [1] (a linear forest is a graph in which each component is a path). Linear forests are

bipartite graphs, which are in turn comparability graphs. It follows that 2-MAXCIS problem on comparability graphs cannot be approximated within a factor of $(1+\epsilon)$, for some $\epsilon > 0$. We show that the hardness gap produced by the above APX-hardness can be amplified (for the case of comparability graphs). We do this via the well-known approach of considering powers of graphs. The same graph powering technique is used to amplify hardness gap for the MAXIS (or the maximum clique problem). However, the standard graph product typically used for this purpose (see [10]) does not serve in our scenario. The reason is that, in our scenario, we need a graph product under which, if G is the union of two comparability graphs, then G^r is also the union of two comparability graphs. The standard graph product mentioned above may not satisfy this property. So, we consider a different graph product and obtain the amplification. (Dumitrescu and Tóth [5] also use a similar graph product for proving a combinatorial result on comparability graphs (see Theorem 5). Using this approach, we prove that if NP $\neq$ P, then the 2-MAXCIS problem on comparability graphs cannot be approximated within any constant factor. Via a simple extension of this result, we also show that if NP $\not\subseteq$ DTIME$[n^{O(\log n)}]$, then the above problem cannot be approximated within a factor of $2^{\sqrt{\log n}}$. Here, a challenging open problem is to show that it is NP-hard to approximate within a factor of n^{ϵ}, for some $\epsilon > 0$.

2 Preliminaries

In this section, we discuss some concepts and notations used in the paper. We also briefly discuss some known algorithmic results about perfect graphs and their special cases.

Notation: For two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, let $G_1 \oplus G_2$ denote their union graph. Let $G = (V, E)$ be a graph. For a subset of vertices $X \subseteq V$, let $G[X]$ denote the subgraph induced by X in G. Let $p : V \to \mathbf{R}$ be a weight or profit function that assigns a profit or weight $p(u)$ to each vertex $u \in V$. For a subset of vertices $X \subseteq V$, we use $p(X)$ to denote the sum of profits over X ($p(X) = \sum_{u \in X} p(u)$).

Basic Graph Parameters: Let $G = (V, E)$ be a graph. We shall use the following notations to refer to some basic graph properties: (i) Independence number $\alpha(G)$: cardinality of the maximum independent set in G; (ii) Clique number $\omega(G)$: cardinality of the maximum clique in G; (iii) Chromatic number $\chi(G)$: minimum number of colors needed to color the vertices of G so that no two adjacent vertices of G receive the same color.

Comparability Graphs: Comparability graphs capture the comparability relationships among elements of a partial ordered set. In this paper, we shall use directed acyclic graphs (DAG) to define them. Consider a DAG $D = (V, E')$. We say that a vertex u is an *ancestor* of a vertex v, if there is a path from u to v. A pair of vertices u and v are said to be *comparable*, if either u is an ancestor of v or v is an ancestor of u; otherwise, they are said to be *incomparable*. Construct a graph $G = (V, E)$; add an edge between every pair of comparable vertices. The graph G is said to be the comparability graph of D. Alternatively, G can be obtained by taking the transitive closure of D and ignoring the directionality of the edges. A graph is said to be a comparability graph, if it is the comparability graph of some DAG. McConnell and Spinrad [11] present a linear time algorithm for testing whether an input graph is a comparability graph; moreover, if the graph is a comparability graph, then their algorithm also outputs a DAG representation (or transitive orientation) in linear time.

Perfect Graphs: Notice that for any graph G, $\chi(G) \geq \omega(G)$. A graph $G = (V, E)$ is said to be *perfect*, if for every induced subgraph H, $\chi(H) = \omega(H)$. Chudnovsky et al. [4, 3] proved that whether an input graph is perfect can be tested in polynomial time.

3 Weighted 2-MAXCIS Problem on Perfect Graphs

In this section, we present a $O(\sqrt{n})$ approximation algorithm for the 2-MAXCIS problem on perfect graphs.

In our approximation algorithm, we will need a polynomial procedure for the tasks of finding maximum weight independent sets and cliques in perfect graphs. Grötschel et al. [8] present polynomial time algorithms for performing both the tasks.

THEOREM 1.*[[8]] The weighted* MAXIS *and weighted* MAXCLIQUE *problems can be solved in polynomial time for perfect graphs.*

Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be the two input perfect graphs over a vertex set V of size n and let $\widehat{G} = (V, \widehat{E}) = G_1 \oplus G_2$ be their union. Let the input profit function be $p : V \to \mathbf{R}$, where $p(u)$ is the profit of a vertex $u \in V$.

Our approximation algorithm is based on a crucial lemma discussed next. Consider a subset $X \subseteq V$. Let $\omega_1(X) = \omega(G_1[X])$ and $\omega_2(X) = \omega(G_2[X])$, be the size of the maximum cliques in the graph induced by X in G_1 and G_2, respectively. The lemma below presents a polynomial time algorithm for finding an independent set I contained in X and having reasonably large profit. The idea behind the lemma is as follows. Suppose $\widehat{H}$ is the union of two perfect graphs H_1 and H_2 over a vertex set of size m. Then, H_1 and H_2 can be colored in $\omega(H_1)$ colors and $\omega(H_2)$ colors, respectively. Therefore, their union $\widehat{H}$ can be colored using $\omega(H_1)\omega(H_2)$ colors. Hence, $\widehat{H}$ must have an independent set of size $m/(\omega(H_1)\omega(H_2))$. The lemma generalizes this idea to the weighted case and applies it to subgraphs. More importantly, it gives a polynomial time procedure for finding such a good independent set. Dumitrescu and Tóth [5] use a similar argument in the context of proving a certain combinatorial property about unions of perfect graphs.

LEMMA 2. *Let $X \subseteq V$. There exists a subset $I \subseteq X$ such that I is an independent set of $\widehat{G}$ and*

$$p(I) \geq \frac{p(X)}{\omega_1(X)\omega_2(X)}.$$

Moreover, such a set can be found in polynomial time.

PROOF. Since G_1 is perfect, $G_1[X]$ is also perfect (by definition). Thus, $G_1[X]$ can be colored using $\omega_1(X)$ colors. This means that X can be partitioned into $\omega_1(X)$ many subsets (i.e., color classes), which are all independent in $G_1[X]$. One of these independent sets must have profit at least $p(X)/\omega_1(X)$. Apply the algorithm given in Theorem 1 and find the maximum profit independent set of $G_1[X]$; let this be I'. We have $p(I') \geq p(X)/\omega_1(X)$. We shall now focus on $G_2[I']$. The graph $G_2[X]$ can be colored with $\omega_2(X)$ colors and so, $G_2[I']$ can also be colored using $\omega_2(X)$ colors. It follows that $G_2[I']$ has an independent set having profit at least $p(I')/\omega_2(X)$. Apply the algorithm given in Theorem 1 and find the maximum profit independent set of $G_2[I']$; let this be I. We have $p(I) \geq p(I')/\omega_2(X)$. It follows that

$p(I) \geq p(X)/(\omega_1(X)\omega_2(X))$. Notice that I is an independent set in both G_1 and G_2, and hence, it is an independent set in $\widehat{G}$. ∎

Our $O(\sqrt{n})$-approximation uses the LP rounding approach. We start the description of the algorithm by first discussing our LP formulation.

Linear Program: Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be the set of all cliques of G_1 and G_2, respectively. Notice that any independent set of $\widehat{G}$ can pick at most one vertex from any clique $C \in \mathcal{C}_1$ (or any $C \in \mathcal{C}_2$). We shall express these constraints in the linear program. For each vertex $u \in V$, add a variable $x(u)$ that indicates whether u is included in the independent set or not. The LP is shown next:

$$\begin{aligned} \max \sum_{u \in V} p(u)x(u) \qquad & \text{subject to} \\ \sum_{u \in C} x(u) \leq 1 \qquad & \text{for all } C \in \mathcal{C}_1 \\ \sum_{u \in C} x(u) \leq 1 \qquad & \text{for all } C \in \mathcal{C}_2 \\ 0 \leq x(u) \leq 1 \qquad & \text{for all } u \in V \end{aligned}$$

The above LP may have exponentially many constraints, since $|\mathcal{C}_1|$ and $|\mathcal{C}_2|$ could be exponential in n. Nevertheless, we can solve it using a separation oracle. Recall that a separation oracle is a procedure which takes as input a fractional assignment $\mathbf{x} : V \to [0,1]$ and decides whether $\mathbf{x}$ is a feasible solution to the LP; moreover, if $\mathbf{x}$ is not a feasible solution, the procedure should output a constraint violated by $\mathbf{x}$. In our case, we can perform the above task in polynomial time using the algorithm given in Theorem 1. Taking $\mathbf{x}$ to be the weight or profit function, find the maximum profit clique in G_1 and G_2, using the above algorithm. If both these cliques have profit at most 1, then we know that $\mathbf{x}$ is a feasible solution. Otherwise, the constraint corresponding to the clique having profit greater than 1 is violated. Therefore, we can construct the required separation oracle. Given such an oracle, the ellipsoid method can solve the LP and find the optimal fraction solution (see [8]).

Let $\mathbf{x} : V \to [0,1]$ be the optimal fractional solution to the LP. For a subset of vertices $X \subseteq V$, let $\mathrm{LP}^*(X) = \sum_{u \in X} p(u)\mathbf{x}(u)$. Let $\mathrm{LP}^* = \mathrm{LP}^*(V)$ denote the profit of the LP solution. Let $\mathrm{P}_{\max} = \max_{u \in V} p(u)$ be the maximum profit.

Rounding Algorithm: We now describe our rounding scheme. First, partition the vertex set V in to SML and LRG, where $\mathrm{SML} = \{u : 0 \leq \mathbf{x}(u) \leq 1/\sqrt{n}\}$ and $\mathrm{LRG} = \{u : 1/\sqrt{n} \leq \mathbf{x}(u) \leq 1\}$. Thus, $\mathrm{LP}^* = \mathrm{LP}^*(\mathrm{SML}) + \mathrm{LP}^*(\mathrm{LRG})$.

As we shall see, it is easy to handle the set SML. So, let us ignore SML for now, and focus on LRG. We geometrically divide the interval $[(1/\sqrt{n}), 1]$ into $\ell = (\log n)/2$ ranges and classify the vertices u in LRG based on the range into which $\mathbf{x}(u)$ falls. Namely, for $0 \leq i < \ell$, define

$$U_i = \{u : (1/\sqrt{n})2^i \leq \mathbf{x}(u) \leq (1/\sqrt{n})2^{i+1}\}.$$

Thus, $U_0, U_1, \ldots, U_{\ell-1}$ forms a partition of LRG.

The rounding algorithm is as follows. For $0 \leq j < \ell$, apply the algorithm given in

Lemma 2 on the set U_j (taking $X = U_j$) and find an independent set I_j of $\widehat{G}$ such that

$$p(I_j) \geq \frac{p(U_j)}{\omega_1(U_j)\omega_2(U_j)}. \tag{1}$$

Then, among these ℓ independent sets, choose the one having the maximum profit. Let I^* be the chosen set. Finally, consider the following two options: (i) the singleton set containing the vertex having the maximum profit $\mathrm{P}_{\max}$; (ii) the set I^*. Between these options, output the one having the maximum profit. Let $\mathrm{I}_{\mathrm{alg}}$ be the set output by the above rounding scheme.

Analysis: We shall now analyze the rounding scheme. The goal is to show that our algorithm has an approximation ratio of $O(\sqrt{n})$. The following lemma presents the main claim of the analysis.

LEMMA 3. $\mathrm{LP}^*(\mathrm{LRG}) \leq (4\sqrt{n})p(I^*)$.

We shall prove the lemma shortly. Let us complete the analysis assuming the lemma. Recall that $\mathrm{LP}^* = \mathrm{LP}^*(\mathrm{SML}) + \mathrm{LP}^*(\mathrm{LRG})$. Observe that $\mathrm{LP}^*(\mathrm{SML}) \leq \sqrt{n}\mathrm{P}_{\max}$ and $p(\mathrm{I}_{\mathrm{alg}}) \geq \mathrm{P}_{\max}$. Hence, $\mathrm{LP}^*(\mathrm{SML}) \leq \sqrt{n} \cdot p(\mathrm{I}_{\mathrm{alg}})$. The lemma implies that $\mathrm{LP}^*(\mathrm{LRG}) \leq (4\sqrt{n})p(\mathrm{I}_{\mathrm{alg}})$. It follows that $\mathrm{LP}^* \leq (5\sqrt{n})p(\mathrm{I}_{\mathrm{alg}})$. Thus, our algorithm has $O(\sqrt{n})$ approximation ratio. We now proceed to prove the lemma. The claim given below is useful for this purpose.

LEMMA 4. *For any* $0 \leq j < \ell$, $\mathrm{LP}^*(U_j) \leq 2(\sqrt{n}/2^j)p(I_j)$.

PROOF. Let $\beta = (1/\sqrt{n})2^j$. Write $U = U_j$ and $I = I_j$. Our goal is to show that $\mathrm{LP}^*(U) \leq 2(1/\beta)p(I)$. Let $\omega_{\min} = \min\{\omega_1(U), \omega_2(U)\}$ and let $\omega_{\max} = \max\{\omega_1(U), \omega_2(U)\}$.

Equation 1 shows that

$$p(I) \geq \frac{p(U)}{\omega_{\min}\omega_{\max}}. \tag{2}$$

Let us now derive a bound on $\mathrm{LP}^*(U)$. By the definition of U, we have that $\beta \leq \mathbf{x}(u) \leq 2\beta$. Therefore, $\mathrm{LP}^*(U) \leq (2\beta)p(U)$. There exists a subset $C \subseteq U$ which is a clique in $G_1[X]$ or $G_2[X]$ such that $|C| = \omega_{\max}$. The LP contains a constraint corresponding to this clique and hence, $\sum_{u \in C} \mathbf{x}(u) \leq 1$. Since every $u \in U$ satisfies $\mathbf{x}(u) \geq \beta$, we have that $\beta\omega_{\max} \leq 1$. Thus, $\beta \leq (1/\omega_{\max})$. Putting together, we get that

$$\mathrm{LP}^*(U) \leq \frac{2p(U)}{\omega_{\max}}. \tag{3}$$

Equations 2 and 3 imply that $\mathrm{LP}^*(U) \leq 2\omega_{\min}p(I)$. Recall that $\beta\omega_{\max} \leq 1$. Thus, $\omega_{\max} \leq (1/\beta)$ and hence, $\omega_{\min} \leq (1/\beta)$. We have proved the lemma. ∎

Proof of Lemma 3: We have $\mathrm{LP}^* = \sum_{j=0}^{\ell-1} \mathrm{LP}^*(U_j)$. Lemma 4 shows that for all U_j, $\mathrm{LP}^*(U_j) \leq 2(\sqrt{n}/2^j)p(I^*)$. Therefore,

$$\mathrm{LP}^* \leq (2\sqrt{n})p(I^*)\sum_{j=0}^{\ell-1}(1/2^j) \leq (4\sqrt{n})p(I^*).$$

∎

Our algorithm and analysis can easily be extended to the case of weighted k-MAXCIS problem on perfect graphs, for a constant k. For this problem, we get an algorithm with an approximation ratio of $O(n^{(k-1)/k})$.

4 Integrality Gaps

Here, we show that the LP considered in the previous section has an integrality gap of $\sqrt{n}$, even when the input graphs are unions of two comparability graphs and the input is unweighted. Then, we consider a strengthening of the LP and show that it has an integrality gap of $n^{0.16}$.

Fix any square number n. We shall construct a graph $G = (V, E)$ on n vertices such that G is the union of two comparability graphs and the LP has an integrality gap of $\sqrt{n}$ on G. Let $k = \sqrt{n}$. Let $A = \{1, 2, \ldots, k\}$ and $V = A \times A$. We will describe G by presenting two DAGs $D_1 = (V, E_1)$ and $D_2 = (V, E_2)$. The DAG D_1 is as follows. Consider a pair of vertices $u = \langle a_1, a_2 \rangle$ and $v = \langle b_1, b_2 \rangle$ belonging to V, where $a_1, a_2, b_1, b_2 \in A$. Add an edge from u to v, if $b_1 \geq a_1 + 1$. The DAG D_2 is constructed in a similar fashion. Add an edge from a vertex $u = \langle a_1, a_2 \rangle$ to a vertex $v = \langle b_1, b_2 \rangle$, if $b_2 \geq a_2 + 1$. The two DAGs can be visualized by considering a $k \times k$ grid of vertices. In D_1, an edge is drawn from a vertex u to all the vertices appearing in rows below the row of u; in D_2, an edge is drawn from a vertex u to all the vertices appearing in columns to the right of column of u. It is easy to see that D_1 and D_2 are acyclic. Let G_1 and G_2 be the comparability graphs of D_1 and D_2, respectively. Take G to be the union of G_1 and G_2. Notice that G is the complete graph on n vertices and so, $\alpha(G) = 1$. On the other hand, $\omega(G_1) = k$ and $\omega(G_2) = k$. So, we get a feasible LP solution by setting $\mathbf{x}(u) = 1/k$, for all $u \in V$. This LP solution has profit $n/k = \sqrt{n}$. We conclude that the LP has an integrality gap of $\sqrt{n}$ on G.

We now describe a strengthening of the previous LP and exhibit integrality gap. The drawback with the previous LP is that it adds one constraint for each clique of G_1 and one constraint for each clique of G_2. A natural idea is to add a constraint for each clique of the union graph G. Namely, for each clique C of G, add a constraint $\sum_{u \in C} x(u) \leq 1$. Notice that this LP does not have any integrality gap on the graph instances of the previous construction. Nevertheless, we show that even this strengthened LP has an integrality gap of $n^{0.16}$. For this, we make use of a nice combinatorial result proved by Dumitrescu and Tóth [5].

THEOREM 5. *[[5]] There exists an infinite family of graphs $\{G\}$ such that each graph G is the union of two comparability graphs and $\omega(G) \leq n^{0.42}$ and $\alpha(G) \leq n^{0.42}$, where n is the number of vertices in G.*

The integrality gap for the strengthened LP follows immediately from the above theorem. Consider any graph $G = (V, E)$ given by this theorem. For all $u \in V$, set $\mathbf{x}(u) = 1/n^{0.42}$. Notice that this is feasible solution to the strengthened LP. Its profit is $n/n^{0.42} = n^{0.58}$. On the other hand, the maximum independent set of G is of size at most $n^{0.42}$. It follows that the LP has an integrality gap of $n^{0.16}$.

5 2-MAXCIS : Hardness Results

Bar-Yehuda et al. [1] show that the 2-MAXCIS problem is APX-hard on linear forests (a linear forest is a graph in which each component is a path). They proved this result by showing that any 3-regular graph can be expressed as the union of two linear forests. Linear

forests are bipartite graphs, which are in turn comparability graphs. It follows that the 2-MAXCIS problem on comparability graphs in APX-hard. This means that there exists an $\epsilon > 0$ such that it is NP-hard to approximate the problem within a factor of $(1+\epsilon)$. The above APX-hardness proof can be transformed into a hardness gap, as stated in the theorem below. A similar claim for the MAXIS problem on 3-regular graphs is implicit in the work of Chlebík and Chlebíková [2]. We can derive the theorem below by combining their result with that that of Bar-Yehuda et al.

THEOREM 6. *There exists a polynomial time algorithm and a constant $\epsilon > 0$ with the following property. The algorithm takes as input a boolean formula φ and outputs two comparability graphs G_1 and G_2, and a number k such that*

$$\begin{aligned} \varphi \in \text{SAT} &\implies \alpha(\widehat{G}) \geq (1+\epsilon)k \\ \varphi \notin \text{SAT} &\implies \alpha(\widehat{G}) \leq k, \end{aligned}$$

where $\widehat{G} = G_1 \oplus G_2$.

We next amplify the gap produced by Theorem 6 and show that the MAXCIS problem on comparability graphs cannot be approximated within any constant factor. We shall perform the amplification by using the well-known approach of taking graph powers. Towards that goal, we define a graph power satisfying two important properties: (i) if G is the union of two comparability graphs, then G^r is also the union of two comparability graphs; (ii) $\alpha(G^r) = [\alpha(G)]^r$.

The graph power is described next. Let $G = (V, E)$ be any graph and let r be any integer. Construct a graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ as follows. The vertex set $\widetilde{V}$ is given by $\widetilde{V} = V \times V \times \cdots V$, where the cartesian product is taken r times. The edges of $\widetilde{G}$ are described next. We add an edge between two vertices $\widetilde{u} = \langle u_1, u_2, \ldots, u_r \rangle$ and $\widetilde{v} = \langle v_1, v_2, \ldots, v_r \rangle$, if $(u_j, v_j) \in E$, where j is the smallest number such that $u_j \neq v_j$. We define the graph power G^r to be the graph $\widetilde{G}$ constructed above. The above graph powering has the following properties.

LEMMA 7. *Consider any integer r.*

- *For any graph G, $\alpha(G^r) = [\alpha(G)]^r$.*
- *If G is a comparability graph then G^r is also a comparability graph.*
- *Let $\widehat{G} = G_1 \oplus G_2$ be the union of two graphs G_1 and G_2. Then, $\widehat{G}^r = G_1^r \oplus G_2^r$.*

PROOF. Consider the first claim. Let $\alpha(G) = k$ and let I be an independent set of G of size k. Let $I' = I \times I \times \cdots I$, where the cartesian product is taken r times. I' is an independent set in G^r and it is of size k^r. Thus, $\alpha(G^r) \geq k^r$. To see the reverse direction, consider an independent set I' of G^r. For $1 \leq j \leq r$, define

$$U_j = \{\langle u_1, u_2, \ldots, u_j \rangle : \text{there exists } u_{j+1}, u_{j+2} \ldots, u_r \text{ such that } \langle u_1, u_2, \ldots, u_r \rangle \in I'\}.$$

By induction on j, we shall show that for all $1 \leq j \leq r$, $|U_j| \leq k^j$. For the base case of $j = 0$, notice that the set U_1 is an independent set in G and hence, $|U_1| \leq k$. To prove the induction step, consider the set U_{j+1}. Pick any $\langle u_1, u_2, \ldots, u_j \rangle \in U_j$. The set $\{u : \langle u_1, u_2, \ldots, u_j, u \rangle \in U_{j+1}\}$ is an independent set in G and so, its cardinality is at most k. By induction $|U_j| \leq k^j$ and hence, $|U_{j+1}| \leq k^{j+1}$. It follows that $I' = U_r$ has cardinality at most k^r.

Consider the second claim. Let $G = (V, E)$ be the comparability graph of some DAG $D = (V, E')$. Without loss of generality assume that D is transitively closed (i.e., if $(a, b) \in E'$ and $(b, c) \in E'$ then $(a, c) \in E'$). We shall define a directed graph version of our graph power and show that G is the comparability graph of D^r. Construct a directed graph $\widetilde{D} = (\widetilde{V}, \widetilde{E})$ as follows. Define $\widetilde{V} = V \times V \times \cdots V$, where the cartesian product is taken r times. Add an edge from a vertex $\widetilde{u} = \langle u_1, u_2, \ldots, u_r \rangle$ to the vertex $\widetilde{v} = \langle v_1, v_2, \ldots, v_r \rangle$, if $(u_j, v_j) \in E'$, where j is the smallest number such that $u_j \neq v_j$. Let $D^r = \widetilde{D}$. We first claim that the graph $\widetilde{D}$ constructed above is transitively closed. To see this, consider three vertices $\widetilde{x} = \langle x_1, x_2, \ldots, x_r \rangle, \widetilde{y} = \langle y_1, y_2, \ldots, y_r \rangle$ and $\widetilde{z} = \langle z_1, z_2, \ldots, z_r \rangle$ such that $(\widetilde{x}, \widetilde{y}) \in \widetilde{E}$ and $(\widetilde{y}, \widetilde{z}) \in \widetilde{E}$. Let j_1 and j_2 be the smallest integers such that $x_{j_1} \neq y_{j_1}$ and $y_{j_2} \neq z_{j_2}$, respectively. Consider the three cases of $j_1 < j_2$, $j_2 < j_1$ and $j_1 = j_2$. In the first case, j_1 is the smallest number such that $x_{j_1} \neq z_{j_1}$. For this index j_1, we have that $(x_{j_1}, y_{j_1}) \in E'$ and $y_{j_1} = z_{j_1}$, and hence, $(x_{j_1}, z_{j_1}) \in E'$. It follows that $(\widetilde{x}, \widetilde{z}) \in \widetilde{E}$. The second case is handled in a similar fashion. For the third case, let $j = j_1 = j_2$. Notice that $(x_j, y_j) \in E'$ and $(y_j, z_j) \in E'$ and hence, $(x_j, z_j) \in E'$ (since D' is transitively closed). It follows that $(\widetilde{x}, \widetilde{z}) \in \widetilde{E}$. We have shown that $\widetilde{D}$ is transitively closed. This also shows that $\widetilde{D}$ is acyclic. We can now easily argue that G is the comparability graph of $\widetilde{D}$.

The third claim is easy to see. ∎

Using the graph power and Lemma 7, we next argue that it is NP-hard to approximate the 2-MAXCIS problem on comparability graphs within any constant factor. Theorem 6 provides an algorithm that produces a gap of $(1 + \epsilon)$. We can amplify the gap by graph powering, as described next. Fix any constant r. Let the algorithm given by Theorem 6 be denoted as $\mathcal{A}$. Consider an algorithm $\mathcal{B}$ that takes as input a formula φ and outputs graphs G_1^r and G_2^r, where G_1 and G_2 are the graphs output by $\mathcal{A}$ on input φ. Lemma 7 shows that G_1^r and G_2^r are comparability graphs. The same lemma shows that $\alpha(G') = [\alpha(\widehat{G})]^r$, where $G' = G_1^r \oplus G_2^r$ and $\widehat{G} = G_1 \oplus G_2$. It follows that

$$\begin{aligned} \varphi \in \text{SAT} &\implies \alpha(G') \geq (1+\epsilon)^r k^r \\ \varphi \notin \text{SAT} &\implies \alpha(G') \leq k^r. \end{aligned}$$

The algorithm $\mathcal{B}$ runs in polynomial time and produces a gap of $(1 + \epsilon)^r$. Consider any constant c and set $r = (\log c)/\log(1 + \epsilon)$. Then, the above algorithm with parameter r produces a gap of c. The above argument leads to the following theorem.

THEOREM 8. *If* NP $\neq$ P *then the* MAXCIS *problem on comparability graphs cannot be approximated within any constant factor.*

We can amplify the gap further. Instead of fixing r to be a constant, let us set $r = d \log n$, where n is the number of vertices in $\widehat{G}$ and d is a suitably defined constant. The output graph $\widehat{G}^r$ will have $N = n^r$ vertices and the gap would become $(1 + \epsilon)^{d \log n}$. Setting $d = (1/\log(1 + \epsilon))^2$, we get a gap of $2^{\sqrt{N}}$. However, taking the rth power would take $n^{O(\log n)}$ time. The above construction leads to the following result.

THEOREM 9. *If* NP $\not\subseteq$ DTIME$[n^{O(\log n)}]$, *then* MAXCIS *problem on comparability graphs cannot be approximated within a factor of* $2^{\sqrt{\log n}}$.

References

[1] R. Bar-Yehuda, M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. *SIAM Journal of Computing*, 36(1):1–15, 2006.

[2] M. Chlebík and J. Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theoretical Computer Science*, 354(3):320–338, 2006.

[3] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vuskovic. Recognizing Berge graphs. *Combinatorica*, 25(2):143–186, 2005.

[4] M. Chudnovsky, N. Robertson, P. Seymour, and R.Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164(1):51–229, 2006.

[5] A. Dumitrescu and G. Tóth. Ramsey-type results for unions of comparability graphs. *Graphs and Combinatorics*, 18(2):245–251, 2002.

[6] A. Frank. Some polynomial algorithms for certain graphs and hypergraphs. In *5th British Combinatorial Conference*, 1976.

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

[8] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[9] D. Hermelin and D. Rawitz. Optimization problems in multiple subtree graphs. In *5th WAOA*, 2009.

[10] D. Hochbaum, editor. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., Boston, 1997.

[11] R. McConnell and J. Spinrad. Linear-time transitive orientation. In *SODA*, 1997.

[12] R. Möhring. Algorithmic aspects of comparability graphs and interval graphs. In I. Rival, editor, *Graphs and orders*, pages 41–101. D. Reidel, Boston, 1985.

[13] D. Zuckerman. Linear degree extractors and the inapproximability of max-clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.