Optimization Problems in Multiple Subtree Graphs*

Danny Hermelin**[1] and Dror Rawitz[2]

[1] Department of Computer Science,
University of Haifa, Haifa 31905, Israel
danny@cri.haifa.ac.il
[2] School of Electrical Engineering,
Tel-Aviv University, Tel-Aviv 69978, Israel
rawitz@eng.tau.ac.il

Abstract. We consider various optimization problems in *t-subtree graphs*, the intersection graphs of t-subtrees, where a t-subtree is the union of t disjoint subtrees of some tree. This graph class generalizes both the class of chordal graphs and the class of t-interval graphs, a generalization of interval graphs that has recently been studied from a combinatorial optimization point of view. We present approximation algorithms for the Maximum Independent Set, Minimum Coloring, Minimum Vertex Cover, Minimum Dominating Set, and Maximum Clique problems in t-subtree graphs.

1 Introduction

Geometric intersection graphs are a very popular topic in algorithmic graph theory. This is mostly because of the many natural applications they model, and due to the rich combinatorial structure that comes along with most of them, allowing for numerous algorithmic techniques and frameworks. Two of the oldest and most well-studied graph classes in this area are the class of interval graphs, intersection graphs of intervals of a line, and the class of chordal graphs, intersection graphs of subtrees of a tree [10]. Many classical NP-complete problems become polynomial-time solvable in both these classes of graphs [10].

In [3, 6], various optimization problems were considered in the class of t-interval graphs, a natural generalization of interval graphs. These are defined as intersection graphs of t-intervals, which are 1-dimensional objects formed by taking the union of t disjoint intervals. In this paper, we consider t-subtree graphs which generalize t-interval graphs by replacing intervals with subtrees. Thus, t-subtree graphs form a hybrid between t-interval graphs and chordal graphs, and provide a natural generalization for these two graph classes.

We consider various classical optimization problems in t-subtree graphs. Given a t-subtree graph G along with its t-subtree representation $\mathcal{S}$, we present approximation algorithms for the following problems:

- Minimum Dominating Set – Find a minimum weight subset $\mathcal{S}' \subseteq \mathcal{S}$ such that for each t-subtree $S \in \mathcal{S}$ there is a t-subtree $S' \in \mathcal{S}'$ which intersects S.
- Maximum Independent Set – Find a maximum weight pairwise non-intersecting subset $\mathcal{S}' \subseteq \mathcal{S}$.
- Minimum Coloring – Partition $\mathcal{S}$ into the smallest number of subsets such that each subset is pairwise non-intersecting.
- Minimum Vertex Cover – Find a minimum weight subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $\mathcal{S} \setminus \mathcal{S}'$ is pairwise non-intersecting.
- Maximum Clique – Find a maximum weight pairwise intersecting subset $\mathcal{S}' \subseteq \mathcal{S}$.

* This work was done at the Caesarea Edmond Benjamin de Rothschild Foundation Institute, University of Haifa.
** Supported by the Adams Fellowship of the Israel Academy of Sciences and Humanities.

Gavril [9] showed that these problems, with the exception of Minimum Dominating Set, can be solved in polynomial-time in chordal graphs. On the other hand, Minimum Dominating Set in chordal graphs was shown to be NP-hard in [5]. In t-interval graphs all problems are computationally hard. Griggs and West [11] showed that the class of graphs with maximum degree Δ are $\lceil(\Delta+1)/2\rceil$-interval graphs. It follows that Minimum Dominating Set and Minimum Vertex Cover are APX-hard, for $t \geq 2$ [13], and that Minimum Coloring is NP-hard, for $t \geq 3$ [8]. Maximum Clique was shown to be NP-hard for $t \geq 3$ in [6]. Bar-Yehuda et al. [3] showed that, for $t \geq 2$, Maximum Independent Set in t-interval graphs is APX-hard, and that it cannot be approximated within a factor of $O(t/\log t)$ unless P=NP. Finally, it is NP-hard to determine whether a given graph is t-interval for $t \geq 2$ [16].

In this paper we present approximation algorithms for optimization problems in t-subtree graphs. For Maximum Independent Set, Minimum Coloring, Minimum Vertex Cover, and Maximum Clique we obtain approximation ratios of $2t$, $2t$, $2-1/t$, and $(t^2-t+1)/2$, respectively, which match the approximation ratios of the corresponding algorithms for t-interval graphs in [3, 6] by extending these in a natural manner. Minimum Dominating Set is different in that it is already as hard to approximate as Minimum Set Cover even in chordal graphs (*i.e.* 1-subtree graphs). We therefore consider the special case where each t-subtree has ℓ leaves, and provide an ℓ^2-approximation algorithm for this case.

2 Preliminaries

All graphs in this paper are simple and undirected. As usual, we denote the vertex-set and edge-set of a given graph G by $V(G)$ and $E(G)$, respectively. For a graph G and a subset of vertices $V \subseteq V(G)$, we let $G-V$ denote the graph obtained by deleting all vertices of V from G, and by $G[V]$ the subgraph $G-(V(G)\setminus V)$. For a vertex $v \in V(G)$, we let $N(v)$ denote the set of neighbors of v, *i.e.* $N(v)=\{u \in V \ : \ (v,u) \in E\}$, and we let $N[v]=N(u)\cup\{v\}$.

Let $\mathcal{T}$ be an infinite rooted tree with vertices denoted by Greek letters. A set of vertices $T \subseteq V(\mathcal{T})$ is a *subtree* of $\mathcal{T}$ if $\mathcal{T}[T]$ is a tree. Two subtrees T and T' *intersect*, denoted $T \cap T'$, if they share a common vertex, and otherwise they are *disjoint*. A set S of at most t subtrees of $\mathcal{T}$ is called a t-subtree over $\mathcal{T}$. Two t-subtrees S and S' *intersect*, denoted $S \cap S'$, if there is a subtree $T \in S$ which intersects a subtree $T' \in S'$, and they are *disjoint* whenever they are non-intersecting. If $\alpha \in T$ for a subtree T in a t-subtree S, we slightly abuse notation by writing $\alpha \in S$.

A family of t-subtrees $\mathcal{S}$ is a *representation* of some graph G if there exists a bijective correspondence $v \rightarrow S_v$ from the vertices of G to the t-subtrees in $\mathcal{S}$ such that $\{u,v\} \in E(G)$ if and only if $S_u \cap S_v$. In this case, G is the *intersection graph* of $\mathcal{S}$, and thus a t-subtree graph, and we write this as $G = G_{\mathcal{S}}$. We will be considering weighted *t-subtree graphs*, *i.e.* t-subtree graphs G with weight functions $w : V(G) \rightarrow \mathbb{Q}^+$.

Let Π be an optimization problem, and let A be an algorithm for Π. Denote by $A(I)$ the value of the solution computed by A for the instance I of Π, and let $\mathrm{OPT}(I)$ denote the optimal value for the instance I. The solution computed by A for an instance I is said to be r-approximate, if $A(I) \leq \mathrm{OPT}(I)\cdot r$, if Π is a minimization problem, and $A(I) \geq \mathrm{OPT}(I)/r$, if Π is a maximization problem. A is an r-approximation algorithm for Π if it computes an r-approximate solution for any instance I of Π. In this case we say that the *approximation ratio* of algorithm A is r. For example, an r-approximation algorithm for Maximum Independent Set returns a pairwise non-intersecting subset of t-subtrees which has total weight at most r times the optimal, for any t-subtree graph.

3 Minimum Dominating Set

We begin with Minimum Dominating Set. We show that this problem is as hard as Minimum Set Cover even when $t = 1$, *i.e.* in chordal graphs. This implies that it is NP-hard to approximate Minimum Dominating Set in 1-subtree graphs within $c \ln n$, for some constant c. We also show that Minimum Dominating Set is NP-hard to approximate within $\ell(\mathcal{S})-1-\varepsilon$, for any $\varepsilon > 0$, where $\ell(\mathcal{S})$ is the maximum number of leaves in any subtree belonging to a t-subtree of $\mathcal{S}$. On the positive side, we present an $\ell(\mathcal{S})^2$-approximation algorithm that extends the t^2-approximation algorithm for Minimum Dominating Set in t-intervals graphs. This algorithm is based on a reduction to the Minimum Path Hitting problem, and on the approximation algorithm of Parekh and Segev [14] given for this problem.

3.1 Approximation Lower Bounds

We start by showing lower bounds for the approximation factor guarantee of any polynomial-time algorithm for Minimum Dominating Set in t-subtree graphs. These are obtained by simple approximation preserving reductions for Minimum Set Cover.

Lemma 1. *There is an approximation preserving reduction from* Minimum Set Cover *to* Minimum Dominating Set *in 1-subtree (chordal) graphs.*

Proof. Let $(X,\mathcal{C})$ be a Minimum Set Cover instance, where $X = \{x_1, \dots, x_n\}$ is a universe of n elements, and $\mathcal{C} = \{C_1, \dots, C_m\}$ is the family of m subsets of X. We assume without loss of generality that $\bigcup_j C_j = X$. We construct a tree $\mathcal{T}$ and a family $\mathcal{S}$ of (1-)subtrees as follows. First, $\mathcal{T}$ is a star with a center denoted by α_0 and n leaves denoted $\alpha_1, \dots, \alpha_n$. The family $\mathcal{S}$ is a family of $n+m$ subtrees, $\mathcal{S} = \{T_1, \dots, T_{n+m}\}$, one per each element in X and set in $\mathcal{C}$. The subtree T_i, for $i \in \{1, \dots, n\}$, contains only the ith leaf α_i of $\mathcal{T}$. The subtree T_{n+j}, for $j \in \{1, \dots, m\}$, is the subtree induced by $\{\alpha_0\} \cup \{\alpha_i : i \in C_j\}$. Finally, let $G = G_{\mathcal{S}}$ be the intersection graph of the subtrees in $\mathcal{S}$. Clearly the construction of $\mathcal{S}$ can be carried out in polynomial-time. To complete the proof, we argue that any set cover of X corresponds to a dominating set in G of equal size, and vice-versa.

Let $\mathcal{C}' \subseteq \mathcal{C}$ be a set cover of X. Then the subset of vertices $\{v_{n+j} : C_j \in \mathcal{C}'\}$, with v_{n+j} the vertex corresponding to the subtree T_{n+j}, dominates all vertices in G. Indeed, all vertices v_{n+j}, $1 \le j \le m$, dominate each other, and if a vertex v_i that corresponds to a leaf α_i is not dominated by D, then x_i is not covered by $\mathcal{C}'$. Conversely, suppose that D is a dominating set in G. We may assume that D does not contain vertices that correspond to leaves, since each vertex that correspond to a leaf can be replaced by a vertex that correspond to a subtree containing this leaf. Furthermore, since all leaves are dominated, the set $\mathcal{C}' = \{C_j : S_{n+j} \in D\}$ is a set cover of U. □

Next, we present a reduction from the special case of Minimum Set Cover in which each element appears in at most t sets, *i.e.* the t-Minimum Set Cover problem, to Minimum Dominating Set in t-subtree graphs.

Lemma 2. *For every $t \in \mathbb{N}$, there is an approximation preserving reduction from t-*Minimum Set Cover *to* Minimum Dominating Set *in t-subtree graphs.*

Proof. Given a t-Minimum Set Cover instance $(X,\mathcal{C})$, with $X = \{x_1, \dots, x_n\}$ and $\mathcal{C} = \{C_1, \dots, C_m\}$, we construct a tree $\mathcal{T}$ and a family $\mathcal{F}$ of subtrees as follows. The tree $\mathcal{T}$ consists of a root α_0 adjacent to m vertex-disjoint paths $\alpha_{j_1}, \dots, \alpha_{j_n}$, for $j \in \{1, \dots, m\}$. For each element

$C_j \in \mathcal{C}$, we designate the subtree $T_j = \{\alpha_0, \alpha_{j_1}, \ldots, \alpha_{j_n}\}$. Also, we let $T_{j_i} = \{\alpha_{j_i}\}$. Now the set of t-subtrees $\mathcal{S}$ consists of $n + m$ t-subtrees, where $S_i = \{T_{j_i} : x_i \in C_j\}$, for $i \in \{1, \ldots, n\}$, and $S_{n+j} = \{T_j\}$, for $j \in \{1, \ldots, m\}$. Observe that $\mathcal{S}$ is a t-subtree family, and that it can be constructed in polynomial-time. We argue that dominating set in $G = G_{\mathcal{S}}$, the intersection graph of $\mathcal{S}$, corresponds to a set cover of X of equal size, and vice-versa.

Let $\mathcal{C}' \subseteq \mathcal{C}$ be a set cover of X. We claim that $D = \{v_{n+j} : C_j \in \mathcal{C}'\}$ is a dominating set in G. First, observe that all vertices v_{n+j}, where $1 \leq j \leq m$, dominate each other since each S_{n+j} includes the root α_0. Also, if a vertex v_i is not dominated by D, then x_i is not covered by any set in $\mathcal{C}'$. Conversely, let D be a dominating set in G. Without loss of generality, we may assume that $v_i \notin D$ for all $1 \leq i \leq n$, since if $x_i \in C_j$ then we can replace v_i with v_{n+j} in D. Thus, since all t-subtrees that correspond to elements are dominated, the set $\mathcal{C}' = \{C_j : T_{n+j} \in D\}$ is a set cover of X. □

It is known that it is NP-hard to approximate Minimum Set Cover within a factor of $c \ln n$ for some constant c [15], and that it is NP-hard to approximate t-Minimum Set Cover within $t - 1 - \varepsilon$ for any $\varepsilon > 0$ [7]. Thus, the two lemmas above show:

Corollary 1. Minimum Dominating Set *in t-subtree graphs is NP-hard to approximate within*

- $c \ln n$ *for some constant c.*
- $\ell(\mathcal{S}) - 1 - \varepsilon$ *for any $\varepsilon > 0$.*

3.2 Bounded Number of Leaves

We next present an $\ell(\mathcal{S})^2$-approximation algorithm for Minimum Dominating Set in t-chordal graphs. As a first step, we show that we can consider edge intersections in the subtrees, rather than vertex intersections.

Lemma 3. *Let G be a t-chordal graph with a t-subtree representation $\mathcal{S}$ over a tree $\mathcal{T}$. Then G has an t-subtree representation $\mathcal{S}^*$ over another tree $\mathcal{T}^*$, with $(u, v) \in E(G)$ if and only if the t-subtrees corresponding to u and v in $\mathcal{S}^*$ share an edge. Furthermore, $\mathcal{S}^*$ can be computed in polynomial-time, and $\ell(\mathcal{S}^*) = \ell(\mathcal{S})$.*

Proof. We construct a t-subtree representation $\mathcal{S}^*$ over $\mathcal{T}^*$ as follows. First, $\mathcal{T}^*$ is obtain by splitting every vertex α of $\mathcal{T}$ into two adjacent vertices:

$$V(\mathcal{T}^*) = \{\alpha' : \alpha \in V(\mathcal{T})\} \cup \{\alpha'' : \alpha \in V(\mathcal{T})\}$$
$$E(\mathcal{T}^*) = \{\{\alpha'', \beta'\} : \{\alpha, \beta\} \in E(\mathcal{T})\} \cup \{\{\alpha', \alpha''\} : \alpha \in V(\mathcal{T})\}$$

Second, for every subtree T belonging to some t-subtree in $\mathcal{S}$, we define the associated subtree T^* with $V(T^*) = \{\alpha' : \alpha \in V(T)\} \cup \{\alpha'' : \alpha \in V(T)\}$. Each t-subtree $S = \{T_1, \ldots, T_t\}$ in $\mathcal{S}$ is then associated with the t-subtree $S^* = \{T_1^*, \ldots, T_t^*\}$. It is not hard to verify that two subtrees S_1 and S_2 contain a node α if and only if the subtrees induced by S_1^* and S_2^* contain the edge (α', α''). Clearly, $\mathcal{S}^*$ can be computed in polynomial-time, and in addition $\ell(\mathcal{S}^*) = \ell(\mathcal{S})$ by construction. □

Next, we define a more general variant of Minimum Dominating Set in t-subtree graphs. In this variant, we are given two (not necessarily disjoint) families of t-subtrees, a red family $\mathcal{R} = \{R_1, \ldots, R_n\}$ and a blue family $\mathcal{B} = \{B_1, \ldots, B_m\}$, and our goal is to find a minimum weight subset $\mathcal{R}' \subseteq \mathcal{R}$ which dominates $\mathcal{B}$, *i.e.* every t-subtree $B \in \mathcal{B}$ is intersected by some $R \in \mathcal{R}'$. (We assume that $\mathcal{R}$ dominates $\mathcal{B}$.) Notice that when $\mathcal{B} = \mathcal{R}$, we return to Minimum Dominating Set in t-chordal graphs.

The extended variant of Minimum Dominating Set in t-subtree graphs can be formulated using the following linear integer program:

$$\begin{array}{lll} \min & \sum_{R \in \mathcal{R}} w(R)x(R) & \\ \text{s.t.} & \sum_{R \,:\, B \sqcap R} x(R) \geq 1 & \forall B \in \mathcal{B} \\ & x(R) \in \{0,1\} & \forall R \in \mathcal{R} \end{array} \qquad \text{(DS)}$$

where $x(R)$, as usual, is a variable corresponding $R \in \mathcal{R}$ and is interpreted as to $x(R) = 1$ if and only if R is taken to the solution. We use $\sqcap$ instead of $\cap$ to remind the reader that consider edge intersections, rather than vertex intersections. The linear programming relaxation of DS is obtained by replacing the integrality constraints by: $x(R) \geq 0$, for every $R \in \mathcal{R}$.

We will need the notion of descending paths and ℓ-paths in $\mathcal{T}$. Recall that $\mathcal{T}$ is rooted. A *descending path* in $\mathcal{T}$ is a path $\alpha_1, \alpha_2, \ldots, \alpha_p$, where α_i is an ancestor of α_j for $i < j$. An ℓ-path P in $\mathcal{T}$ is a collection of at most ℓ *descending paths* which are pairwise edge-disjoint. An *ℓ-path representation* of a t-subtree graph G, is a representation which consists only of ℓ-paths.

The proof of the next lemma is immediate:

Lemma 4. *If G is a t-subtree graph with a t-subtree representation $\mathcal{S}$, then there exists an ℓ-path representation of G with $\ell = \ell(\mathcal{S})$.*

For the rest of this section we design an approximation algorithm for (DS) in graphs with ℓ-path representations (where intersections are by edges). We first address the case where both $\mathcal{R}$ and $\mathcal{B}$ contain 1-paths, each consisting of a single descending path. In [14], Parekh and Segev devised a 4-approximation algorithm for the Minimum Path Hitting problem using a reduction to this special case of (DS). Formally, they obtained the following result:

Lemma 5 ([14]). *If $\mathcal{R}$ and $\mathcal{B}$ consist of* 1*-subtrees, each of which contains a single descending path, then the LP-relaxation of* (DS) *has an integral optimal solution, and this solution can be computed in polynomial time.*

We next show how to use this result to obtain an approximation algorithm the case of $\ell > 1$. Let $\mathcal{R}$ and $\mathcal{B}$ be two families of ℓ-paths. We may assume without loss of generality that all ℓ-paths in $\mathcal{R}$ and $\mathcal{B}$ contain exactly ℓ paths. Now, let x^* denote the corresponding optimal solution of the LP-relaxation of DS.

For a descending path P in a blue ℓ-path $B \in \mathcal{B}$, let $\mathcal{R}(P) \subseteq \mathcal{R}$ denote that subset of red ℓ-paths that have descending paths intersecting P. For each $B \in \mathcal{B}$, we select a unique *representative* descending path $P_B \in B$ that maximizes the expression $\sum_{R \in \mathcal{R}(P)} x^*(R)$. In other words, the representative P_B is the maximum dominated descending path of B. For a red ℓ-path $R \in \mathcal{R}$, we let P_R^i denote the ith path in R, for $1 \leq i \leq \ell$. We construct a new 1-path instance $(\mathcal{R}', \mathcal{B}', w')$ by the taking the representatives of the blue ℓ-subtrees to be $\mathcal{B}'$, *i.e.* $\mathcal{B}' = \{P_B \,:\, B \in \mathcal{B}\}$, and defining $\mathcal{R}' = \left\{P_R^i \,:\, P_i \in R, R \in \mathcal{R}\right\}$ with $w'(P_R^i) = w(R)/\ell$ for $P_R^i \in \mathcal{R}'$. Next we use Parekh and Segev's result (Lemma 5) to obtain an integral optimal solution x' of the LP-relaxation of (DS) with respect to the new instance $(\mathcal{R}', \mathcal{B}', w')$. We output the solution $\mathcal{D} \subseteq R$ defined by $\mathcal{D} = \left\{R : x'(P_R^i) = 1 \text{ for some } i\right\}$.

Lemma 6. *$\mathcal{D}$ is an ℓ^2-approximate dominating set of $\mathcal{B}$.*

Proof. First observe that $\mathcal{D}$ dominates $\mathcal{B}$, since all representatives are dominated by $\mathcal{D}$. Also, note that
$$\sum_{R\in\mathcal{D}} w(R)x(R) \le \ell \cdot \sum_{R'\in\mathcal{R}'} w'(R')x'(R'),$$
and that this inequality is tight in case no two descending paths of the same ℓ-path appear in $\mathcal{D}$. To complete the proof of the lemma, we show that
$$\sum_{R\in\mathcal{R}} w(R)x(R) \le \ell^2 \cdot \sum_{R\in\mathcal{R}} w(r)x^*(R).$$
For this, we define a fractional solution $\bar{x}$ for the new instance $(\mathcal{R}', \mathcal{B}', w')$ by setting $\bar{x}(P^i_R) = \ell{\cdot}x^*(R)$ for each $P^i_R \in \mathcal{R}'$. To see that $\bar{x}$ is feasible consider any blue ℓ-interval $B \in \mathcal{B}$ and its representative P_B. By our selection of P_B, we have $\sum_{R\in\mathcal{R}(P_B)} x^*(R) \ge 1/\ell$, since otherwise x^* would not be feasible. It follows that $\sum_{R'\in\mathcal{R}',R'\sqcap B'} \bar{x}(R') \ge 1$ for each $B' \in \mathcal{B}'$. By Lemma 5, we know that the integral solution x' is an optimal fractional solution for $(\mathcal{R}', \mathcal{B}', w')$. Hence, the weight of $\bar{x}$ is at least as high as the weight of x'. It follows that
$$\sum_{R\in\mathcal{R}} w(R)x(R) \le \ell \cdot \sum_{R'\in\mathcal{R}'} w'(R')x'(R') \le \ell \cdot \sum_{R'\in\mathcal{R}'} w'(R')\bar{x}(R')$$
Furthermore,
$$\sum_{R'\in\mathcal{R}'} w'(R')\bar{x}(R') = \left(\sum_{R\in\mathcal{R}}\sum_{P\in R} \frac{w(R)}{\ell} \cdot \ell \cdot x^*(R)\right) = \ell \cdot \sum_{R\in\mathcal{R}} w(R)\cdot x^*(R)$$
and we are done. □

Corollary 2. Minimum Dominating Set *in t-subtree graphs can be approximated in polynomial-time within a factor of $\ell(\mathcal{S})^2$, where $\mathcal{S}$ is the representation of the input graph.*

We remark that our algorithm works for the case where $\mathcal{R}$ contains r-paths and $\mathcal{B}$ contains b-paths, and in this case the approximation ratio is $r \cdot b$. In fact, the instance that is generated by our second reduction (see Section 3.1) can be described by $r = 1$ and $b = t$, and in this case the approximation ratio of our algorithm comes close to the $\ell(\mathcal{S})$ lower bound given in Corollary 1.

4 Maximum Independent Set

We next consider Maximum Independent Set. We present a $2t$-approximation algorithm for Maximum Independent Set that is based on the fractional local-ratio $2t$-approximation algorithm for t-interval graphs from [3].

Let $G = (V, E)$ be a t-subtree graph, and let $(\mathcal{T}, \mathcal{S})$ be its t-subtree representation, with ρ the root of $\mathcal{T}$. We define the *root* of a subtree T of $\mathcal{T}$, denoted $\rho(T)$, to be the node in T that is closest to ρ in $\mathcal{T}$. (Note that $\rho(T)$ can be ρ itself.) We let $\text{root}(\mathcal{S})$ denote the set of roots of subtrees that belong to t-subtrees of $\mathcal{S}$. That is, $\text{root}(\mathcal{S}) = \{\rho(T) \;:\; T \in S, S \in \mathcal{S}\}$.

Lemma 7. *Let S and S' be two intersecting t-subtrees in $\mathcal{S}$. Then, there exists some vertex $\alpha \in$ root($\mathcal{S}$) such that $\alpha \in S$ and $\alpha \in S'$.*

Proof. If S and S' intersect, they have a pair of intersecting subtrees $T \in S$ and $T' \in S'$. For these two trees we have either $\rho(T) \in T'$ or $\rho(T') \in T$. □

Recall that, for a vertex v of G, S_v denotes the t-subtree in $\mathcal{S}$ corresponding to v, and $w(v)$ denotes the weight of v. Lemma 7 implies that it is sufficient to look for intersections at root$(\mathcal{S})$. This allows us to formulate Maximum Independent Set in t-subtree graphs as the following integer program:

$$\begin{array}{lll} \max & \sum_v w(v) \cdot x(v) & \\ \text{s.t.} & \sum_{\alpha \in S_v} x(v) \leq 1 & \alpha \in \text{root}(\mathcal{S}) \\ & x(v) \in \{0,1\} & \forall v \in V(G) \end{array} \tag{IS}$$

As usual, $x(v)$ above denotes a variable corresponding to a vertex v of G which is to be interpreted as $x(v) = 1$ if and only if the vertex v is chosen to the independent set. The linear programming relaxation of (IS) is obtained by replacing the constraints $x(v) \in \{0,1\}$ with $0 \leq x(v) \leq 1$, for every vertex v of G. Note that the integer (and linear) programming formulation for Maximum Independent Set in t-interval graphs given in [2] is identical to the one above when $\mathcal{T}$ is a path.

Given a feasible solution x to the LP-relaxation of (IS), let us call the sum $\sum_{u \in N[v]} x(u)$ the *fractional neighborhood* of a vertex v with respect to x (recall that $N[v]$ denotes the set of neighbors of v in G including v itself). The fractional local-ratio based algorithm for Maximum Independent Set in t-interval graphs given in [3] essentially works by repeatedly selecting a vertex with a minimal fractional neighborhood, and deciding whether this vertex is in the solution independent set according to the recursive solution for $G - N[v]$. They proved that if for any feasible solution x, there is always some vertex v with fractional neighborhood at most r, then their algorithm computes an r-approximate independent set. In order to extend their algorithm to t-subtree graphs, we prove the following lemma which generalizes the corresponding lemma for t-interval graphs from [3]:

Lemma 8. *Given any feasible solution x of the LP-relation of of* (IS)*, there exists a vertex v of G with fractional neighborhood at most $2t$ with respect to x.*

Proof. In order to prove this lemma, it is enough to show that

$$\sum_v \sum_{u \in N[v]} x(v) \cdot x(u) = \sum_v x(v) \sum_{u \in N[v]} x(u) \leq 2t \cdot \sum_v x(v) \ .$$

If $\{u,v\} \in E$ then $u \in N[v]$ and $v \in N[u]$. Therefore, the term $x(v) \cdot x(u)$ is counted twice in the sum on the left hand side for every pair of neighboring vertices v and u. Furthermore, if $\{u,v\} \in E$ then either there exists $T \in S_v$ such that $\rho(T) \in S_u$ or there exists $T \in S_u$ such that $\rho(T) \in S_v$. Thus,

$$\sum_v \sum_{u \in N[v]} x(v) \cdot x(u) \ \leq \ 2 \cdot \sum_v \sum_{T \in S_v} \sum_{\rho(T) \in S_u} x(v) \cdot x(u) \ .$$

Since x is a feasible solution of P, for every v and $T \in S_v$, we get that

$$\sum_{\rho(T) \in S_u} x(v) \cdot x(u) \ = \ x(v) \cdot \sum_{\rho(T) \in S_u} x(u) \ \leq \ x(v) \ .$$

Therefore,

$$\sum_v \sum_{u \in N[v]} x(v) \cdot x(u) \ \leq \ 2 \cdot \sum_v \sum_{T \in S_v} x(v) \ = \ 2t \cdot \sum_v x(v) \ ,$$

and we are done. □

Corollary 3. Maximum Independent Set *in t-subtree graphs can be approximated in polynomial-time within a factor of $2t$.*

5 Minimum Coloring and Vertex Cover

In this section we present approximation algorithms for Minimum Coloring and Minimum Vertex Cover. The former achieves an approximation factor of $2t$, and the latter achieves a factor of $2-1/t$. Both algorithms rely on the following structural lemma for t-subtree graphs, which extends the corresponding lemma for t-interval graphs from [3]:

Lemma 9. *Any t-subtree graph G with maximum clique size k can be colored in polynomial-time using at most $2t(k-1)$ colors.*

Proof. Let G be a t-subtree, and let $\mathcal{S}$ denote its t-subtree representation. Denote by G^* the intersection graph of $\bigcup \mathcal{S}$, the set of subtrees that appear in $\mathcal{S}$. Clearly, $|V(G^*)| \leq t \cdot |V(G)|$, since each vertex in G corresponds to at most t vertices in G, and $|E(G)| \leq |E(G^*)|$, since each edge in G corresponds to at least one edge in G^*. Notice that G^* is chordal and has maximum clique size at most k. Thus, $|E(G^*)| < (k-1)|V(G^*)|$, as can be seen by counting all forwards edges in a simplicial ordering of G. Therefore,

$$|E(G)| \;\leq\; |E(G^*)| \;<\; (k-1)|V(G^*)| \;\leq\; t(k-1)|V(G)| \;.$$

It follows that the average degree of G is less than $2t(k-1)$, and so the standard greedy algorithm can be used to color G with at most $2t(k-1)$ colors. □

Since the maximum clique size of a graph G is a lower bound on the number of colors used in any coloring of G, and in particular in an optimal one, we obtain:

Corollary 4. Minimum Coloring *in t-subtree graphs can be approximated in polynomial-time within a factor of* $2t$.

Let us next consider Minimum Vertex Cover. Here we propose an algorithm which consists of two stages. The first stage involves removing triangles from our input graph G by applying a technique originally introduced in [1] in order to remove short odd cycles. Using this technique, we obtain in polynomial-time a triangle-free subgraph G' of G such that any r-approximate vertex cover of G' can be easily transformed into a $\max\{r, 1.5\}$-approximate vertex cover of G. The same triangle-cleaning phase is also performed in [6] to approximate Minimum Vertex Cover in t-interval graphs.

The second stage consists of using the algorithm of Hochbaum [12] which gives a factor of $(2-2/c)$ for Minimum Vertex Cover in graphs that can be colored in polynomial-time with c colors. Combining this with Lemma 9, we get an algorithm that computes in polynomial-time a $(2-1/t)$-approximate vertex cover for the triangle-free subgraph G' produced in the first stage. As mentioned above, this vertex cover can be transformed into a vertex cover of G which is $(2-1/t)$-approximate for $t \geq 2$.

Corollary 5. Minimum Vertex Cover *in t-subtree graphs can be approximated in polynomial-time within a factor of* $(2-1/t)$.

6 Maximum Clique

In this section we present a $(t^2-t+1)/2$-approximation algorithm for Maximum Clique in t-subtree graphs.

We begin with the notion of a transversal. A *transversal* of a subset $\mathcal{S}' \subseteq \mathcal{S}$ is a set of vertices $\{\alpha_1, \ldots, \alpha_\tau\} \subset V(\mathcal{T})$ such that for every $S \in \mathcal{S}$ there is at least one $\alpha_i \in \{\alpha_1, \ldots, \alpha_\tau\}$ with $\alpha_i \in S$. Note that due to Lemma 7, we can assume that any transversal is a subset of $\text{root}(\mathcal{S}')$. The *transversal number* of $\mathcal{S}'$ is the minimum size of any transversal of $\mathcal{S}$.

A pairwise intersecting subset of $\mathcal{S}$ is called a τ-clique if it has transversal number equal to τ. Berger [4] proved upper bounds on the transversal number of any pairwise intersecting family of t-subtrees.

Lemma 10 ([4]). *Any pairwise intersecting subset $\mathcal{S}' \subseteq \mathcal{S}$ is a $(t^2 - t + 1)$-clique.*

As in t-interval families, the maximum weight 2-clique in $\mathcal{S}$ can be computed in polynomial-time due to a couple of simple observations.

Lemma 11. *The maximum weight 2-clique in $\mathcal{S}$ can be computed in polynomial time.*

Proof. Consider a pair of vertices $\alpha, \beta \in \text{root}(\mathcal{S})$, and let $\mathcal{S}' = \{S \in \mathcal{S} \,:\, \alpha \in S \text{ or } \beta \in S\}$. Then the intersection graph $G_{\mathcal{S}'}$ is the complement of a bipartite graph, since both $\{S \in \mathcal{S} \,:\, \alpha \in S\}$ and $\{S \in \mathcal{S} \,:\, \beta \in S\}$ are pairwise intersecting. Since Maximum Independent Set is polynomial-time solvable in bipartite graphs, we can compute the maximum weight clique in $G_{\mathcal{S}'}$ in polynomial-time. Thus, by iterating over all $O(n^2)$ pairs of vertices in $\text{root}(\mathcal{S})$, we can compute the maximum weight 2-clique in $\mathcal{S}$ in polynomial time. □

Combining both lemmas above, we obtain:

Corollary 6. Maximum Clique *in t-subtree graphs can be approximated in polynomial-time within a factor of $(t^2 - t + 1)/2$.*

Acknowledgments

This paper is dedicated to Martin C. Golumbic on his 60th birthday. Both authors are grateful for having the privilege of working at the Caesarea Edmond Benjamin de Rothschild Foundation Institute at the University of Haifa, headed and founded by Martin. We are grateful not only for the many seminars and research-related discussions we had with Martin along the years (in particular, the first author was introduced to the topic of geometric intersection graphs by Martin), but also for the numerous hallway discussions with him about life and academia. Martin has definitely been a cornerstone in both of our academic careers.

References

1. R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–46, 1985.
2. R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. In *13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 732–741, 2002.
3. R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. *SIAM Journal on Computing*, 36(1):1–15, 2006.
4. E. Berger. KKM – A topological approach for trees. *Combinatorica*, 25(1):1–18, 2004.
5. K. S. Booth and J. H. Johnson. Dominating set in chordal graphs. *SIAM Journal on Computing*, 11(1):191–199, 1982.
6. A. Butman, D. Hermelin, M. Lewenstein, and D. Rawitz. Optimization problems in multiple-interval graphs. In *18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 268–277, 2007.
7. I. Dinur, V. Guruswami, S. Khot, and O. Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM Journal on Computing*, 34(5):1129–1146, 2005.

8. M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.
9. F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1(2):180–187, 1972.
10. M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
11. J. R. Griggs and D. B. West. Extremal values of the interval number of a graph. *SIAM Journal on Algebraic and Discrete Methods*, 1(1):1–14, 1980.
12. D. S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6:243–254, 1983.
13. C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and Systems Sciences*, 43:425–440, 1991.
14. O. Parekh and D. Segev. Path hitting in acyclic graphs. *Algorithmica*, 52(4):466–486, 2008.
15. R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *29th ACM Symposium on the Theory of Computing*, pages 475–484, 1997.
16. D. B. West and D. B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. *Discrete Applied Mathematics*, 8:295–305, 1984.