Chapter 3

Testing superperfection of k-trees

Much work has been done in recognizing classes of perfect graphs in polynomial time [35, 72, 76, 135, 155]. An exception seems to be the class of superperfect graphs.

The results presented in this chapter can be summarized as follows. First we give in section 3.2 a complete characterization, by means of forbidden induced subgraphs, of 2-trees that are superperfect. We also characterize those 2-trees that are comparability graphs and those that are permutation graphs. Secondly, in section 3.3 we give *for each constant* k an $O(1)$ time algorithm which produces a complete characterization of superperfect k-trees, by means of forbidden configurations. With the aid of this characterization we find, for each constant k, a linear-time algorithm to test superperfection of k-trees.

Until now we have not been able to find a polynomial algorithm to test superperfection on partial k-trees (for general k). Since by definition a graph is superperfect if for each assignment of non-negative weights to the vertices the interval chromatic number is equal to the maximum weight clique, the following observation is of interest. Determining the interval chromatic number of a weighted interval graph with weights one and two is NP-hard. When restricted to weighted partial k-trees, for some constant k, it can be seen that the interval chromatic number can be determined in linear time.

The class of superperfect graphs contains the class of comparability graphs (hence also the class of bipartite graphs), but these classes are not equal. This has been pointed out by Golumbic [76] who showed the existence of an infinite class $\mathcal{S}$ of superperfect graphs which are not comparability graphs. However all graphs in $\mathcal{S}$ are neither triangulated nor cotriangulated, and therefore Golumbic [76] raises the question whether for triangulated graphs the classes of superperfect and comparability graphs coincide. For split graphs this equivalence has been shown. Our results show it is not the case for triangulated graphs in general.

3.1 Preliminaries

We start with some definitions and easy lemmas. Most definitions and results in this section are taken from [76]. For further information on superperfect graphs the reader is referred to this book.

Definition 3.1.1 *An undirected graph* $G = (V, E)$ *is called a comparability graph, or a transitively orientable graph, if there exists an orientation of the edges such that the resulting oriented graph* (V, F) *satisfies the following conditions.*

$$F \cap F^{-1} = \emptyset \text{ and } F + F^{-1} = E \text{ and } F^2 \subseteq F$$

where $F^2 = \{(a, c) \mid \exists_{b \in V} (a, b) \in F \wedge (b, c) \in F\}$. *An orientation* F *of the edges satisfying the conditions above is called a* transitive *orientation.*

So, if F is a transitive orientation, then $(a, b) \in F$ and $(b, c) \in F$ imply $\{a, c\}$ is an edge with orientation $(a, c) \in F$. There exists a somewhat weaker equivalent condition (which we do not use): a graph is a comparability graph if and only if it admits an orientation of its edges that represents a pseudo-order relation (see [15], page 76).

If a graph G is a comparability graph, then this also holds for every induced subgraph of G. In [76] it is shown that comparability graphs are perfect, and can be recognized in polynomial time (see also [155]).

A *weighted* graph is a pair (G, w), where G is a graph and w a weight function which associates to every vertex x a non-negative weight $w(x)$. For a subset S of the vertices we define the weight of S, denoted by $w(S)$, as the sum of the weights of the vertices in S.

Definition 3.1.2 *An* interval coloring *of a weighted graph* (G, w) *maps each vertex* x *to an open interval* I_x *on the real line, of width* $w(x)$, *such that adjacent vertices are mapped to disjoint intervals. The* total width *of an interval coloring is defined to be* $|\bigcup_x I_x|$. *The* interval chromatic number $\chi(G, w)$ *is the least total width needed to color the vertices with intervals.*

Determining whether $\chi(G, w) \leq r$ is an NP-complete problem, even if w is restricted to values one and two and G is an interval graph. (This has been shown by L. Stockmeyer as reported in [76].) In this paper we shall only use the following alternative characterization of the interval chromatic number (see [76]).

Theorem 3.1.1 *If* (G, w) *is a weighted undirected graph, then*

$$\chi(G, w) = \min_F \left(\max_\mu w(\mu) \right)$$

where F *is an acyclic orientation of* G *and* μ *is a path in* F.

If w is a weight function and F is an acyclic orientation, then we say that F is a *superperfect orientation* with respect to w if the weight of the heaviest path in F does not exceed the weight of the heaviest clique.

Definition 3.1.3 *The* clique number $\Omega(G, w)$ *of a weighted graph* (G, w) *is defined as the maximum weight of a clique in* G.

In this chapter we use the capital Ω to denote the (weighted) clique number rather then ω, to avoid confusion with the weight function w. It is easy to see that $\Omega(G, w) \leq \chi(G, w)$ holds for all weighted graphs, since for any acyclic orientation and for every clique there exists a path in the orientation which contains all vertices of the clique.

Definition 3.1.4 *A graph* G *is called* superperfect *if for* every non-negative weight function w, $\Omega(G, w) = \chi(G, w)$.

Notice that each induced subgraph of a superperfect graph is itself superperfect, and also that every superperfect graph is perfect. If G is a comparability graph, then there exists an orientation such that every path is contained in a clique. This proves the following theorem (see also [76]).

Theorem 3.1.2 *Every comparability graph is superperfect.*

The converse of this theorem is not true. In [76] an infinite class of superperfect graphs is given that are not comparability graphs. However, none of these graphs is triangulated. In [76] (page 214) the question is raised whether the converse of the theorem holds for triangulated graphs; is it true or false that, for *triangulated* graphs, G is a comparability graphs if and only if G is superperfect? In the next section we answer this question in the negative, and we give a complete characterization of superperfect 2-trees.

3.2 2-trees and superperfection

In this section we give a characterization of the 2-trees that are superperfect by means of forbidden subgraphs. In 1967 Gallai [68] published a list of all minimal forbidden subgraphs of the comparability graphs (see also [15] page 78). Extracting from this list the triangulated graphs which are subgraphs of 2-trees (or: have treewidth at most two), we find a characterization of the 2-trees which are comparability graphs. We find two types of forbidden induced subgraphs, which we call the 3-*sun* and the *odd wing*. They are illustrated in figure 3.1. Notice that a 3-sun and a wing are 2-trees, and that a wing has at least seven vertices. We call a wing odd (even) if the total number of vertices is odd (even). The following lemma is easy to check.

Lemma 3.2.1 *A wing is a comparability graph if and only if it is even.*

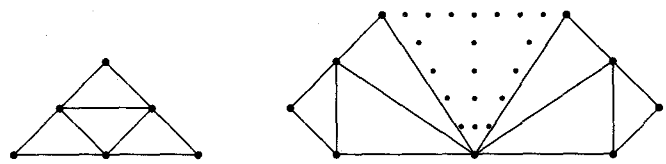

Figure 3.1: 3-sun (left) and wing (right)

We thus find the following characterization of 2-trees that are comparability graphs.

Theorem 3.2.1 *A 2-tree is a comparability graph if and only if it does not contain a 3-sun or an odd wing.*

It is interesting to notice that we can get a characterization of the 2-trees that are permutation graphs. Recall that a graph is an interval graph if and only if it is triangulated and a cocomparability graph [72]. Also, a graph is a permutation graph if and only if the graph and its complement are comparability graphs [135]. It follows that a 2-tree is a permutation graph if and only if it is an interval graph without an induced odd wing.

The next theorem shows that the smallest odd wing, with seven vertices, (which is not a comparability graph) is superperfect. As we shall see later, this is in fact the only odd wing that is superperfect. Notice that in [76] (page 212, figure 9.9) this graph is mistakenly placed in the position of a non-superperfect graph. See also [125] and [62]; the result of [125] is wrong: A wing is an interval graph.

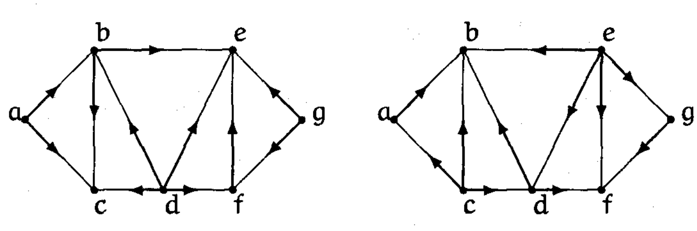

Figure 3.2: two orientations of the wing with seven vertices

Theorem 3.2.2 *The odd wing with seven vertices is superperfect and hence there exists a triangulated graph which is superperfect but not a comparability graph.*

Proof. Label the vertices of the graph as in figure 3.2. We consider two orientations of this wing as illustrated in figure 3.2 and we show that for every weight function one of these orientations is superperfect. Notice that both orientations are such that there is exactly one path not contained in a triangle. In the first orientation this is the path $\{a, b, e\}$ and in the second orientation the path $\{c, d, f\}$. Consider a non-negative weight function w of the vertices. Suppose the orientation of the first type is *not* superperfect with respect to w. Then the path $\{a, b, e\}$ must be heavier then every triangle. Since $\{a, b, c\}$ is a triangle, this implies that $w(e) > w(c)$. But then $w(\{c, d, f\}) < w(\{e, d, f\})$, and since $\{e, d, f\}$ is a triangle, the second orientation is superperfect with respect to w. □

In the last part of this section we give a complete characterization of the superperfect 2-trees. In figure 3.3 we give a list of forbidden induced subgraphs. Notice that the fourth subgraph starts an infinite series. The following lemma can be easily checked.

Lemma 3.2.2 *The graphs shown in figure 3.3 are not superperfect. For each of the graphs the weight function that is shown is such that for any acyclic orientation, there exists a path which is heavier than the heaviest clique.*

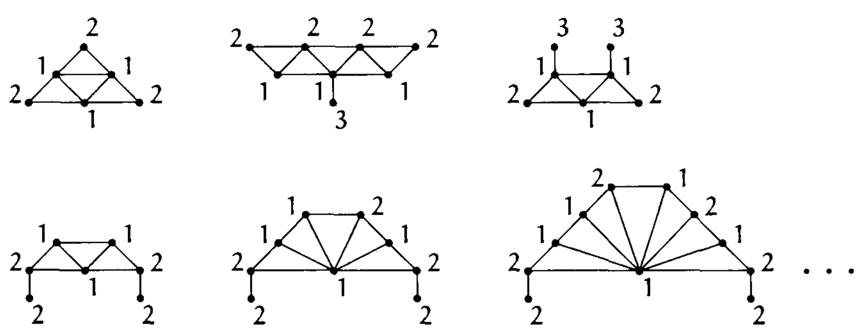

Figure 3.3: critical non-superperfect graphs

Theorem 3.2.3 *A 2-tree is superperfect if and only if it does not contain an induced subgraph isomorphic to one of the graphs shown in figure 3.3.*

Proof. Assume the 2-tree G has no induced subgraph from this list. Then the graph cannot have an induced odd wing with nine or more vertices. We may assume the graph is not a comparability graph, hence it contains an odd wing with seven vertices. Consider the labeled wing of figure 3.2. Let H be the subgraph obtained from this wing by removing the vertices a and g. Let (x, y) be an edge of H. We say that C is a component at (x, y) if C is a maximal connected subgraph of $G[V \setminus \{x, y\}]$ *containing no vertices of* H. If C is a component at (x, y), we say that x is a degenerate vertex of this component if x is adjacent to all vertices of C. Notice that there are only four edges of H at which there can be components, namely (b, c), (b, d), (d, e) and (e, f), otherwise the 3-sun would be an induced subgraph. The following remarks restrict the possible components.

1. It can be checked that if C is a component at (x, y), then either x or y must be degenerate.

2. For components at (b, d), b must be degenerate and for components at (d, e), e must be degenerate, otherwise the second graph in the list is an induced subgraph.

3. Consider components at (b, c) with at least two vertices. Either b or c is degenerate for *all* these components, otherwise the second graph from list 3.3 is an induced subgraph.

4. If there is a component at (b, c) with at least two vertices for which c is degenerate, then for all components at (e, f), e must be degenerate, otherwise the fourth graph in the list is an induced subgraph. If there is a component at (b, c) with at least two vertices for which b is degenerate then for all components at (e, f), f must be degenerate, otherwise the third subgraph is an induced subgraph. Without loss of generality, we assume that for all components at (b, c), c is degenerate and for all components at (e, f), e is degenerate.

5. If there is a component at (b, d) and one at (d, e), then the third graph in the list is an induced subgraph, hence this can not be the case.

6. If there is a component at (e, f) with at least two vertices (for which e is degenerate) then there can be no component at (b, d), otherwise the third graph from the list is an induced subgraph.

7. If there is a component at (b, c) with at least two vertices, then all components at (b, d) can have only one vertex, otherwise the second graph of the list is an induced subgraph.

8. Suppose there is a component at (b, d). Then all components at (b, c) can have at most two vertices. Furthermore, if there is a component at (b, c) with two vertices, then it is the only component at (b, c). Otherwise, the third subgraph of the list is an induced subgraph.

It follows that only one of two cases can occur.

- There is a component at (b, d). Then there is no component at (d, e). All components at (b, d) and at (e, f) have one vertex. Either all components at (b, c) have one vertex or there is only one component at (b, c), in which case it can have at most two vertices and c is degenerate.

- There are only components at (b, c), (d, e) and (e, f). For all components at (b, c), c is degenerate. For all components at (d, e) and at (e, f), e is degenerate.

It is easily checked (e.g. by methods described in the next section) that both types are superperfect. □

Notice that the list of forbidden induced subgraphs is infinite. In the next section we show that for each k there is a finite characterization of the class of superperfect k-trees, by means of forbidden configurations. Furthermore we give for each k a $O(1)$ time algorithm to find this characterization. As a consequence we find a linear time algorithm to check whether a k-tree is superperfect.

3.3 k-trees and superperfection

In this section, let k be some constant, and let G be a k-tree. We start by showing that we can restrict the set of orientations if we want to test whether G is superperfect. In this chapter a *coloring* of a graph $G(V, E)$ with $k+1$ colors is a function $C : V \to \{1, \ldots, k+1\}$, such that $C(x) \neq C(y)$ whenever x and y are adjacent. We only use colorings with $k+1$ colors; we do not always mention the number of colors. A coloring of a k-tree is unique up to a permutation of the colors, as is easily seen by induction.

Lemma 3.3.1 *If C and C' are two colorings of a k-tree $G = (V, E)$ then there exists a permutation π of the colors $\{1, \ldots, k+1\}$ such that for every vertex x, $C(x) = \pi(C'(x))$.*

By this fact, the following set of orientations is uniquely defined for each k-tree.

Definition 3.3.1 *Let G be a graph and let C be a coloring of G with $k+1$ colors. For each permutation π of the colors we define an orientation F_π as follows. Direct the edge (x, y) from x to y if $\pi(C(x)) < \pi(C(y))$. Let $\mathcal{F}_c(G)$ be the set of orientations obtained in this way.*

The following lemma is an immediate consequence of Definition 3.3.1.

Lemma 3.3.2 *If G is a k-tree then:*

1. $|\mathcal{F}_c| = (k+1)!$.
2. *Each* $F \in \mathcal{F}_c$ *is acyclic.*
3. *If* $F \in \mathcal{F}_c$ *then each path* μ *in* F *has at most* $k+1$ *vertices.*

Definition 3.3.2 *Let G be a k-tree. Let $\mathcal{F}^*(G)$ be the set of those acyclic orientations of G in which every path contains at most $k+1$ vertices.*

Notice that $\mathcal{F}_c \subseteq \mathcal{F}^*$.

Lemma 3.3.3 $\mathcal{F}_c = \mathcal{F}^*$.

Proof. We prove that $|\mathcal{F}^*| = (k+1)!$ which implies the result. Let $F \in \mathcal{F}^*$. Let S be a k-clique in G, and let x and y be two vertices which are adjacent to all vertices of S. Since S is a clique and F is acyclic, there is a unique ordering of the vertices of S, say $s_1, s_2, \ldots, s_k$, such that there is an arc from s_i to s_j ($s_i \to s_j$) if and only if $i > j$. Since x is adjacent to all vertices of S, there exists an index $0 \leq t_x \leq k$ such that $x \to s_i$ for all $1 \leq i \leq t_x$ and $s_j \to x$ for all $t_x < j \leq k$. The same holds for y with index t_y. Consider the case $t_x < t_y$. Then F has a path of length $k+2$:

$$(s_k, s_{k-1}, \ldots, s_{t_y+1}, y, s_{t_y}, \ldots, s_{t_x+1}, x, s_{t_x}, \ldots, s_1)$$

Since $F \in \mathcal{F}^*$, we find that $t_x = t_y$. Now consider the inductive construction of G as a k-tree. Start with an acyclic orientation of a $(k+1)$-clique. This can be done in $(k+1)!$ manners. If we add a new vertex v and make it adjacent to a k-clique, by the argument above, the orientations of the edges incident with v are determined. Hence $|\mathcal{F}^*| = (k+1)!$. □

Definition 3.3.3 *Let F be an acyclic orientation. A path μ in F is* contained *in a path μ', if all vertices of μ are also vertices of μ'.*

Lemma 3.3.4 *Let $F \in \mathcal{F}_c$. Then any path μ in F is contained in a path μ' with $k+1$ vertices.*

Proof. Let C be a coloring and let $F = F_\pi$ for some permutation π. The colors in the path μ must appear in the same order as in the permutation. Assume there is a gap between adjacent colors c_1 and c_2 in the path (i.e., there is a color in the permutation between c_1 and c_2). Since the edge of the path with colors c_1 and c_2 is contained in a $(k+1)$-clique, the missing colors can be put between c_1 and c_2. Thus we can make a longer path μ' containing μ. □

Theorem 3.3.1 *Let* G *be a* k*-tree. Then* G *is superperfect if and only if for all weight functions* w

$$\min_{F \in \mathcal{F}_c} \max_{\mu} w(\mu) = \Omega(G, w)$$

Proof. Assume G is superperfect. Let w be a non-negative weight function. There is an orientation F such that $\max_\mu w(\mu) = \Omega(G, w)$. If every path in F has at most $k+1$ vertices, we are done. Assume F has a path with more than $k+1$ vertices. Now increase all weights with some constant $L > \Omega(G, w)$. Let w' be this new weight function. Notice that $\Omega(G, w') = \Omega(G, w) + (k+1)L$. Since G is superperfect, there must be an orientation F' for this new weight function w'. Suppose F' also has a path μ with more than $k+1$ vertices. Then (with $|\mu|$ the number of vertices of μ):

$$\begin{aligned} \Omega(G, w') = \Omega(G, w) + (k+1)L &\geq w'(\mu) \\ &= w(\mu) + |\mu| L \\ &\geq w(\mu) + (k+2)L \\ &\geq (k+2)L \end{aligned}$$

Since $L > \Omega(G, w)$, this is a contradiction. We may conclude that $F' \in \mathcal{F}^* = \mathcal{F}_c$. We show that F' is also a good orientation for the weight function w. Let ν be a path in F'. By Lemma 3.3.4, ν is contained in a path ν^* with $k+1$ vertices. Hence

$$w(\nu) \leq w(\nu^*) = w'(\nu^*) - (k+1)L \leq \Omega(G, w') - (k+1)L = \Omega(G, w)$$

The converse is trivial. □

Definition 3.3.4 *Let* G *be a triangulated graph and let* C *be a coloring of* G *with* $k+1$ *colors. For each permutation* π *of the colors, let* $\mathcal{P}(\pi)$ *be the set of paths in* F_π*, which are* not *contained in a clique and which have* $k+1$ *vertices. If* $\mathcal{Q}$ *is a set of paths in* G*, we say that* $\mathcal{Q}$ *is a* cover *if, for every permutation* π*, there is a path* $\mu \in \mathcal{Q}$ *which can be oriented such that it is in* $\mathcal{P}(\pi)$*. A cover is called* minimal *if it contains* $\frac{1}{2}(k+1)!$ *paths.*

Lemma 3.3.5 *Let* G *be a* k*-tree. If for some permutation* π, $\mathcal{P}(\pi) = \emptyset$*, then* G *is a comparability graph (hence superperfect).*

Proof. Suppose $\mathcal{P}(\pi) = \emptyset$. Consider the orientation F_π. If there is a path in F_π which is not contained in a clique then, by Lemma 3.3.4, $\mathcal{P}(\pi)$ cannot be empty. Hence, every path in F_π is contained in a clique. Since F_π is acyclic, the lemma follows. □

Definition 3.3.5 *Let* G *be a* k*-tree and let* C *be a coloring of* G*. Let* S *be a maximal clique of* G*, and let* $\mathcal{Q}$ *be a minimal cover. Define* $LP(G, S, \mathcal{Q})$ *as the following set of inequalities:*

1. *For each vertex* x*:* $w(x) \geq 0$.
2. *For each maximal clique* $S' \neq S$*:* $w(S') \leq w(S)$.
3. *For each path* μ *of* $\mathcal{Q}$*:* $w(\mu) > w(S)$.

We call the second type of inequalities, the clique inequalities*. The inequalities of the third type are called the* path inequalities.

Lemma 3.3.6 *There are* $n - k - 1$ *clique inequalities and* $\frac{1}{2}(k+1)!$ *path inequalities.*

Proof. Notice that a k-tree has $n - k$ cliques with $k + 1$ vertices. □

Theorem 3.3.2 *Let* G *be a* k*-tree with a coloring* C*.* G *is* not *superperfect if and only if there is a maximal clique* S *and a minimal cover* $\mathcal{Q}$ *such that* $LP(G, S, \mathcal{Q})$ *has a solution.*

Proof. Suppose $LP(G, S, \mathcal{Q})$ has a solution. Take this solution as a weight function. Then, clearly, for any orientation F_π there is a path (in $\mathcal{Q}$ and in $\mathcal{P}(\pi)$) which is heavier than the heaviest clique S. By Theorem 3.3.1 G is not superperfect. On the other hand, if G is not superperfect, there exists a weight function w such that for every orientation F_π there is a path which is heavier than the heaviest clique (hence it can not be contained in a clique). By Lemma 3.3.4 we may assume this path has $k + 1$ vertices, hence it is in $\mathcal{P}(\pi)$. Take S to be the heaviest clique and let $\mathcal{Q}$ be a minimal cover for these paths. □

Notice that we could use Theorem 3.3.2 for a polynomial time algorithm to test superperfection on k-trees. There are at most n^{k+1} different paths of length k, hence the number of minimal covers is at most $(n^{k+1})^{\frac{1}{2}(k+1)!}$. Since the number of maximal cliques of G is at most $n - k$, we only have to check for a polynomial number of sets of inequalities if it has a solution. This checking can be done in polynomial time, e.g. by the ellipsoid method. We now show that there also exists a linear time algorithm.

Consider a set of inequalities $LP(G, S, \mathcal{Q})$ which has a solution. Notice that if some vertex y does *not* appear in the path inequalities then we can set the weight $w(y) = 0$. This new weight function is also a solution. Hence we can transform the set of inequalities as follows:

Definition 3.3.6 *Let* G *be a* k*-tree and let* C *be a coloring of* G. *Let* S *be a maximal clique, and let* $\mathcal{Q}$ *be a cover. Let* H *be the subgraph of* G *induced by the vertices of* S *and of all paths in* $\mathcal{Q}$. *Define* $LP'(H, S, \mathcal{Q})$ *as the following set of inequalities:*

1. *For each vertex* x *of* H: $w(x) \geq 0$.
2. *For each maximal clique* $S' \neq S$ *of* H: $w(S') \leq w(S)$.
3. *For each path* μ *in* $\mathcal{Q}$: $w(\mu) > w(S)$.

The following lemma follows directly from Definitions 3.3.5 and 3.3.6.

Lemma 3.3.7 $LP(G, S, \mathcal{Q})$ *has a solution if and only if* $LP'(H, S, \mathcal{Q})$ *has a solution.*

Lemma 3.3.8 *The number of inequalities of* $LP'(H, S, \mathcal{Q})$ *is bounded by a constant.*

Proof. There are $\frac{1}{2}(k+1)!$ paths in $\mathcal{Q}$, each involving $k+1$ variables. The clique S has also $k+1$ vertices, hence it follows that the subgraph H, has at most $(\frac{1}{2}(k+1)!+1)(k+1)$ vertices. Since H is triangulated, the number of maximal cliques in H is bounded by the number of vertices. Hence the number of clique inequalities is bounded by $(\frac{1}{2}(k+1)!+1)(k+1)$. □

We now have the following algorithm to test superperfection of k-trees.

Algorithm to test superperfection of G

Step 1 Generate a list of all $(k+1)$-colored triangulated graphs H, with at most $(1+\frac{1}{2}(k+1)!)(k+1)$ vertices, for which there exists:

1. A maximal clique S with $k+1$ vertices
2. A set $\mathcal{Q}$ of $\frac{1}{2}(k+1)!$ paths, which is a minimal cover,

such that $LP'(H, S, \mathcal{Q})$ has a solution.

Step 2 Make a coloring of G (with $k+1$ colors).

Step 3 Check whether a graph H from the list is an induced subgraph of G (preserving colors). If G has a subgraph from the list then G is not superperfect, otherwise it is.

Notice that generating the list takes $O(1)$ time (if k is a constant). Since the subgraphs have constant size, we can check whether such a subgraph is an induced subgraph of G in linear time, using standard techniques for (partial) k-trees (see [5]).

Theorem 3.3.3 *The algorithm correctly determines whether* G *is superperfect, and does so in linear time.*

Proof. Assume G is not superperfect. Let C be a coloring of G. By Theorem 3.3.2, $\mathsf{LP}(\mathsf{G}, \mathsf{S}, \mathcal{Q})$ has a solution for some maximal clique S and some minimal cover $\mathcal{Q}$. Take H the colored subgraph induced by vertices of S and of paths in $\mathcal{Q}$. By Lemma 3.3.7, $\mathsf{LP}'(\mathsf{H}, \mathsf{S}, \mathcal{Q})$ has a solution, so the subgraph H is in the list. Conversely, suppose the colored graph G has a colored induced subgraph H from the list. Then H has a clique S with $k+1$ vertices and a minimal cover $\mathcal{Q}$ such that $\mathsf{LP}'(\mathsf{H}, \mathsf{S}, \mathcal{Q})$ has a solution. Since H is an induced subgraph of G preserving colors, the clique S is also a maximal clique in G and the cover $\mathcal{Q}$ is also a cover for G. Since $\mathsf{LP}(\mathsf{G}, \mathsf{S}, \mathcal{Q})$ has a solution, G can not be superperfect. $\square$

Notice that the list only has to contain those subgraphs H, of which every vertex is either in the maximal clique S or on some path in $\mathcal{Q}$ (with S and $\mathcal{Q}$ as defined in the algorithm). For reasons of simplicity, we left this detail out of the algorithm.