6. M. Farber (1983), "Characterizations of strongly chordal graphs," *Discrete Math.* 43, 173-189.

7. M. C. Golumbic (1980), *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York.

8. A. J. Hoffman, A. W. J. Kolen and M. Sakarovitch (1985), "Totally balanced and greedy matrices," *SIAM J. Alg. Disc. Meth.* 6, 721-730.

9. R. Lasker and D. Shier (1981), "Construction of (r, d)-invariant chordal graphs," *Congressus Numerantium* 33, 155-165.

10. J. Lehel and Z. Tuza (1986), "Neighborhood perfect graphs," *Discrete Math.* 61, 93-101.

11. A. Lubiw (1987), "Doubly lexical ordering of matrices," *SIAM J. Comput.* 16, 854-879.

12. R. Paige and R. E. Tarjan (1987), "Three partition refinement algorithms," *SIAM J. Comput.* 16, 973-989.

13. E. Sampathkumar and P. S. Neeralagi (1985), "The neighborhood number of a graph," *Indian J. Pure Appl. Math.* 16, 126-132.

14. P. J. Slater (1976), "R-domination in graphs," *J. Assoc. Comput. Mach.* 23, 446-450.

15. J. P. Spinrad, "Doubly lexical ordering of dense 0-1 matrices," preprint.

16. J. Wu, "Neighborhood covering and neighborhood independence in strongly chordal graphs," preprint.

4.9, $i' > k-1$ and so $i' = k$, i.e., x_k is adjacent to $\bar{x}_{k-1}$. Thus $\bar{x}_{k-1} \leq z < \bar{x}_k$ and $z, \bar{x}_k \in N^+[\bar{x}_{k-1}]$. Similarly, $\bar{y}_{k-1} \leq z < \bar{y}_k$ and $z, \bar{y}_k \in N^+[\bar{y}_{k-1}]$. Without loss of generality, assume $\bar{x}_{k-1} \leq \bar{y}_{k-1}$. Then, by (3.2), $\bar{y}_{k-1}$ is adjacent to $\bar{x}_k$. Now, $\bar{x}_k$ k-neighborhood covers (x, x'). Also, $\bar{y}_{k-1}$ $(k-1)$-neighborhood covers (y, y') and so $\bar{x}_k$ k-neighborhood covers (y, y'). So we have $\bar{x}_k > z$ and $\bar{x}_k$ k-neighborhood covers both (x, x') and (y, y'), a contradiction. Q.E.D.

Theorem 4.4 *Algorithm CI finds a minimum k-neighborhood covering set C^* and a maximum k-neighborhood independent set I^* of a strongly chordal graph $G = (V, E)$ in linear time if a strong elimination order is given.*

Proof. The correctness of the algorithm follows from Theorems 4.1, 4.2, and 4.3. The algorithm is linear, since iteration i only costs $O(|N^+(i)|)$ time except line 19 and line 19 in the whole algorithm costs at most $O(|E|)$ time. Q.E.D.

References

1. R. P. Anstee and M. Farber (1984), "Characterization of totally balanced matrices," *J. Algorithms* 5, 215-230.

2. G. J. Chang (1988/89), "Labeling algorithms for the domination problems in sun-free chordal graphs," *Discrete Appl. Math.* 22, 21-34.

3. G. J. Chang and G. L. Nemhauser (1984), "The k-domination and k-stability problems on sun-free chordal graphs," *SIAM J. Alg. Disc. Meth.* 5, 332-345.

4. G. J. Chang and G. L. Nemhauser (1985), "Covering, packing and generalized perfection," *SIAM J. Alg. Disc. Meth.* 6, 109-132.

5. G. J. Chang, M. Farber, and Z. Tuza (1993), "Algorithmic aspects of neighborhood numbers," *SIAM Discrete Math.* 6, to appear.

$z \equiv x_r, x_{r-1}, \cdots, x_{i+1}, \bar{x}_i$. By Lemma 4.10, $b(\bar{x}_i) \leq r - i \leq k - i = a(\bar{x}_i)$ in iteration $\bar{x}_i$. But this is possible only when $k = r = i$, and this contradicts $i < r$. If $\bar{x}_{i-1} > x_{i+1}$, then $b(\bar{x}_i) \leq k - i$ after iteration x_{i+1}, by Lemma 4.10. In particular, $b(\bar{x}_i) \leq k - i$ in iteration $\bar{x}_{i-1}$. But Case 2 and line 10 of the algorithm hold in iteration $\bar{x}_{i-1}$. By line 10, $k - (i-1) = a(\bar{x}_{i-1}) \leq b(x_i) \leq k - 1$, a contradiction.

So, in any case $z \neq \bar{y}$, i.e., $z < \bar{y}$. Similarly, $z < \bar{x}$. Consider the two increasing paths P_1 and P_2, where P_1 is $z \equiv x_r, x_{r-1}, \cdots, x_i, \bar{x}_{i^*}, \bar{x}_{i^*+1}, \cdots, \bar{x}_u \equiv \bar{x}$ (with $\bar{x}_{i^*}$ omitted when $x_i = \bar{x}_{i^*}$ and $i = i^*$) and P_2 is $z \equiv y_t, y_{t-1}, \cdots, y_j, \bar{y}_{j^*}, \bar{y}_{j^*+1}, \cdots, \bar{y}_v \equiv \bar{y}$ (with $\bar{y}_{j^*}$ omitted when $y_j = \bar{y}_{j^*}$ and $j = j^*$). By Lemma 4.8, $(\alpha, \beta) \in E$ for some $\alpha \in V(P_1) - \{z\}$ and $\beta \in V(P_2) - \{z\}$. Note that $\alpha = x_a$ with $a \leq r - 1$ or $\alpha = \bar{x}_a$ with $a \leq u$, and $\beta = y_b$ with $b \leq t - 1$ or $\beta = \bar{y}_b$ with $b \leq v$. We can also assume it is not the case that $\alpha = \bar{y}_u$ and $\beta = \bar{y}_v$ unless P_1 and P_2 both are of length one. Let $a' = \max\{a, k-1\}$ and $b' = \max\{b, k-1\}$. Then α a'-neighborhood covers (x, x') and so β $(a'+1)$-neighborhood covers (x, x') except when $\alpha = x$, in which case α 1-neighborhood covers (x, x') and β 2-neighborhood covers (x, x'). Also, β b'-neighborhood covers (y, y') and so α $(b'+1)$-neighborhood covers (y, y') except when $\beta = y$, in which case β 1-neighborhood covers (y, y') and α 2-neighborhood covers (y, y'). Therefore, α or β k-neighborhood covers both (x, x') and (y, y'), which contradicts the choice of z, unless ($k = 1, \alpha = x$, and $\beta = y$) or ($u = v = k$ and P_1 and P_2 are both of length one). We shall argue that these two cases are also impossible.

First consider the case where $k = 1, \alpha = x$, and $\beta = y$. Without loss of generality, we may assume that $x < y$. Note that $z \leq x_0 = x < y$. $x', y \in N^+(x)$ imply that y 1-neighborhood covers (x, x') by (3.1). Therefore y k-neighborhood covers both (x, x') and (y, y'), which contradicts the choice of z.

Next, consider the case where $u = v = k$ and P_1 and P_2 are both of length one. Note that by the choice of i^*, either $x_i = \bar{x}_{i^*}$ with $i = i^*$ or $x_i < \bar{x}_{i^*}$ with $i^* \leq i$ and $(x_i, \bar{x}_{i^*}) \in E$. For the former case, either $r - 1 = i = i^* = u = k$, which is impossible, or $r = i = i^* = u - 1 = k - 1$, which imply $\bar{x}_{k-1} = z$. For the latter case, $r = i \geq i^* = u = k$ and so $r = k$. Since $\bar{x}_{k-1} < x_k$, by Lemma 4.8, $\bar{x}_{k-1}$ is adjacent to some $x_{i'} > \bar{x}_{k-1}$, where $i' \leq k$. By Lemma

to the assumption that Case 2 holds in iteration $\bar{x}_i$ for $0 \leq i \leq u-1$. Consider the increasing path $z \equiv x_r$, x_{r-1}, $\cdots$, x_1, $x_0 \equiv x \equiv \bar{x}_0$. By Lemma 4.10, $b(x_1) \leq r-1$ in iteration x_1 and so by line 17 $b(x_0) \leq b(x_1)+1 \leq r$ in iteration x_1. Then $b(x) \leq r \leq k$ in iteration x. On the other hand, by Lemma 4.5, $a^*(x) = a(\bar{x}_0) = k$ and so $a(x) = k$ in any iteration. Then $a(x) = b(x) = k$ in iteration x since Case 2 of the algorithm holds. Thus $b(x_1) = k-1$ and $b(x) = k$ in iteration x_1 and $a(x) = b(x) = k = r$ in iteration x. By line 18, $x \in R$ in iteration x_1. Now, $d(z,x) = r = k$. Since $x' > x$, we also have $d(z,x') = k$. By Lemma 4.6, we may assume that x' is adjacent to x_1. By line 17, $b(x') \leq b(x_1)+1 \leq k$ in iteration x_1 and hence $b(x') \leq k$ in iteration x. Since Case 2 and line 10 hold in iteration x, $a(x') \geq a(x) = k$ and $b(x') \geq a(x) = k$. So, $a(x') = b(x') = k$ in iterations x and x_1. By line 18, $x' \in R$ in iteration x_1. Therefore, by line 19, $x' \notin N_0^+(x)$ after iteration x_1 is completed. In iteration x, it is impossible that $x' = s(x) \equiv N_0^+(x)$. Thus $i \geq 1$.

Suppose $i^* \leq i-1$. Then $x_i < \bar{x}_{i^*}$ with $(x_i, \bar{x}_{i^*}) \in E$. In this case, $0 < i^* < k$. Since $z \in C^*$ and $z \equiv x_r$, x_{r-1}, $\cdots$, x_i, $\bar{x}_{i^*}$ is an increasing path, by Lemma 4.10, $b(\bar{x}_{i^*}) \leq r-i+1 \leq k-i^* = a(\bar{x}_{i^*})$ in iteration $\bar{x}_{i^*}$. However, this is possible only when $i^* = 0$ or k since Case 1 or 2 holds in iteration $\bar{x}_{i^*}$, a contradiction to $0 < i^* < k$. Hence, $i = i^*$ in any case. Now $x_i = \bar{x}_i$ or $x_i < \bar{x}_i$. For the case of $x_i < \bar{x}_i$, by $i > 0$ and the choice of i^*, we have $\bar{x}_{i-1} < x_i$. Consider the two increasing paths $x \equiv \bar{x}_0$, $\bar{x}_1$, $\cdots$, $\bar{x}_{i-1}$ and $x \equiv x_0$, x_1, $\cdots$, x_i. By Lemma 4.8, $\bar{x}_{i-1}$ is adjacent to some $x_{i''} > \bar{x}_{i-1}$, where $i'' \leq i$. By Lemma 4.9, $i-1 < i''$ and so $i'' = i$. Thus either $x_i = \bar{x}_i$ or $\bar{x}_{i-1} < x_i < \bar{x}_i$ with x_i adjacent to both $\bar{x}_{i-1}$ and $\bar{x}_i$.

Suppose $i = r$, i.e., $x_i = z = \bar{y} = \bar{y}_v \in C^*$. If $\bar{x}_{i-1} \leq \bar{y}_{v-1}$, then $\bar{y}_{v-1} \in N_{\bar{x}_{i-1}}[x_i]$. By (3.2), $N_{\bar{x}_{i-1}}[x_i] \subseteq N_{\bar{x}_{i-1}}[\bar{x}_i]$. Thus $\bar{y}_{v-1}$ is adjacent to $\bar{x}_i > x_i = \bar{y}_v$, in contradiction to $\bar{y}_v = \max N^+[\bar{y}_{v-1}]$. So, $\bar{y}_{v-1} < \bar{x}_{i-1}$. In iteration $\bar{y}_{v-1}$, $a(\bar{y}_v) = k-v$ by line 12 of the algorithm. Also, $N^+(\bar{y}_v) \neq \emptyset$ since it contains $\bar{x}_i$. By Lemma 4.5, $v = k$. Then, $a(\bar{y}_v) = 0$ in and after iteration $\bar{y}_{v-1}$. In particular, $a(\bar{y}_v) = 0$ in iteration $\bar{x}_{i-1}$. But, in iteration $\bar{x}_{i-1}$, Case 2 and line 10 hold. By line 9, $0 < a(\bar{x}_{i-1})$. But, by line 10, $0 = a(\bar{y}_v) \geq a(\bar{x}_{i-1})$, a contradiction. Therefore, $i < r$, i.e., x_{i+1} exists.

If $\bar{x}_{i-1} \leq x_{i+1}$, then x_{i+1} is adjacent to $\bar{x}_i$ by (3.2) again. Consider the increasing path

Theorem 4.3 *I^* is a k-neighborhood independent set of G.*

Proof. Suppose I^* is not a k-neighborhood independent set, i.e., suppose there exists some vertex $z \in V$ that k-neighborhood covers two distinct edges (x, x') and (y, y') in I^*, where $x' = s^*(x)$ and $y' = s^*(y)$. Without loss of generality, we may assume that z is chosen to be as large as possible.

By the algorithm, there exist $\bar{x}, \bar{y} \in C^*$ such that $x = A^*(\bar{x})$ and $y = A^*(\bar{y})$. Let $x \equiv \bar{x}_0$, $\bar{x}_1, \cdots, \bar{x}_u \equiv \bar{x}$ be the maximum path of $\bar{x}$ and $y \equiv \bar{y}_0, \bar{y}_1, \cdots, \bar{y}_v \equiv \bar{y}$ the maximum path of $\bar{y}$. Since $x \neq y$, $\bar{x} \neq \bar{y}$ by Lemma 4.5. Also, each $\bar{x}_a$ a-neighborhood covers (x, x') for $1 \leq a \leq u$ and each $\bar{y}_b$ b-neighborhood covers (y, y') for $1 \leq b \leq v$.

Choose a shortest x-z path $x \equiv x_0, x_1, \cdots, x_r \equiv z$ and a shortest y-z path $y \equiv y_0, y_1, \cdots, y_t \equiv z$. By Lemma 4.6, we may assume that each x_a $(k-1)$-neighborhood covers (x, x') for $1 \leq a \leq r-1$ and each y_b $(k-1)$-neighborhood covers (y, y') for $1 \leq b \leq t-1$. By Lemma 4.7, we may assume that $x_0 < x_1 < \cdots < x_i$ and $x_i > x_{i+1} > \cdots > x_r$ for some $0 \leq i \leq r$ and $y_0 < y_1 < \cdots < y_j$ and $y_j > y_{j+1} > \cdots > y_t$ for some $0 \leq j \leq t$.

Suppose $\bar{x}_u < x_i$. Consider the two increasing paths $x \equiv \bar{x}_0, \bar{x}_1, \cdots, \bar{x}_u \equiv \bar{x}$ and $x \equiv x_0, x_1, \cdots, x_i$. By Lemma 4.8, $\bar{x}_u$ is adjacent to some $x_{i'} > \bar{x}_u \equiv \bar{x}$ where $i' \leq i$. So $N^+(\bar{x}) \neq \emptyset$. By Lemma 4.5, $u = k$. By Lemma 4.9, $u < i'$. Then $k = u < i' \leq i \leq r$ and so $d(z, x) = r > k$, in contradiction to the assumption that z k-neighborhood covers the edge (x, x'). Therefore $x_i \leq \bar{x}_u$. Let i^* be the smallest index i' such that $x_i \leq \bar{x}_{i'}$. Note that $i^* \leq i$, since $x_i \leq \bar{x}_i$ by Lemma 4.9. Now either $x_i = \bar{x}_{i^*}$ or $x_i < \bar{x}_{i^*}$. For the former case, $i = d(x, x_i) = d(x, \bar{x}_{i^*}) = i^*$, i.e., $x_i = \bar{x}_i$. For the latter case, by Lemma 4.8 again, x_i is adjacent to some $\bar{x}_{i''} > x_i$ where $i'' \leq i^*$. By the minimality of i^*, $i^* = i''$ and so x_i is adjacent to $\bar{x}_{i^*}$. In conclusion, either $x_i = \bar{x}_{i^*}$ with $i = i^*$ or $x_i < \bar{x}_{i^*}$ with $i^* \leq i$ and $(x_i, \bar{x}_{i^*}) \in E$.

Similarly, $y_j < \bar{y}_v$. Let j^* be the smallest index j' such that $y_j < \bar{y}_{j'}$. Then, either $y_j = \bar{y}_{j^*}$ with $j = j^*$ or $y_j < \bar{y}_{j^*}$ with $j^* \leq j$ and $(y_j, \bar{y}_{j^*}) \in E$.

We now claim that $z \neq \bar{y}$. Suppose $z = \bar{y} \in C^*$. Suppose $i = 0$. First $r > 0$, otherwise $r = 0$ would imply $\bar{x}_0 = x = z \in C^*$ and then Case 1 holds in iteration x_0, in contradiction

$0 \leq i \leq r-1$. If the lemma is not true for r, then $i < r-1$ and $x_{r-2} > x_{r-1} < x_r$. Now x_r, $x_{r-2} \in N^+(x_{r-1})$ and so $(x_r, x_{r-2}) \in E$ by (3.1), which contradicts the assumption that x_0, x_1, $\cdots$, x_r is a shortest path. This proves the lemma. Q.E.D.

Lemma 4.8 *If $x \equiv x_0$, x_1, x_2, $\cdots$, x_r and $x \equiv y_0$, y_1, y_2, $\cdots$, y_t are two increasing paths with $x_r < y_t$, then x_r is adjacent to some $y_j > x_r$, where $1 \leq j \leq t$.*

Proof. We shall prove the lemma by induction on r. For $r = 0$, x_0 is adjacent to $y_1 > x_0$. Suppose $r \geq 1$ and x_{r-1} is adjacent to some $y_i > x_{r-1}$. Since $x_r, y_i \in N^+(x_{r-1})$, by (3.1), x_r is adjacent to or equal to y_i. Choose a maximum y_j such that x_r is adjacent to or equal to y_j. Suppose $y_j \leq x_r$. Since $y_j \leq x_r < y_t$, $j < t$ and $x_r, y_{j+1} \in N^+[y_j]$. By (3.1), x_r is adjacent to or equal to y_{j+1}, in contradiction to the maximality of y_j. Thus $y_j > x_r$ and x_r is adjacent to y_j. Q.E.D.

Lemma 4.9 *If $x \equiv x_0$, x_1, x_2, $\cdots$, x_r and $x \equiv y_0$, y_1, y_2, $\cdots$, y_r are two increasing paths with $x_i = \max N^+(x_{i-1})$ for $1 \leq i \leq r$, then $x_r \geq y_r$.*

Proof. We shall prove the lemma by induction on r. The lemma is trivial for $r = 0$. Suppose $r \geq 1$ and $x_{r-1} \geq y_{r-1}$. Suppose the lemma is not true for r, i.e., $x_r < y_r$. Then $x_{r-1} < y_r$. Consider the increasing paths $x \equiv x_0, x_1, x_2, \cdots, x_{r-1}$ and $x \equiv y_0, y_1, y_2, \cdots, y_r$. By Lemma 4.8, x_{r-1} is adjacent to some $y_j > x_{r-1} \geq y_{r-1}$, where $1 \leq j \leq r$. Consequently, $y_j = y_r$. Since $x_r = \max N^+(x_{r-1})$ and $y_r \in N^+(x_{r-1})$, $x_r \geq y_r$. Q.E.D.

Lemma 4.10 *If y_0, y_1, $\cdots$, y_r is an increasing path and $y_0 \in C^*$, then $b(y_i) \leq i$ after iteration y_{i-1} for $1 \leq i \leq r$. Consequently, $b(y_i) \leq i$ in and after iteration y_i for $0 \leq i \leq r$.*

Proof. We shall prove the lemma by induction on i. If $i = 1$, then, in iteration y_0, $b(y_0) = 0$ by line 8 of the algorithm and so $b(y_1) \leq 1$ by line 17. Suppose $i \geq 2$ and $b(y_{i-1}) \leq i-1$ after iteration y_{i-2}. In iteration y_{i-1}, by line 17, we have $b(y_i) \leq b(y_{i-1}) + 1 \leq i$. Hence the lemma holds. Q.E.D.

If $N^+(\bar{x}) \neq \emptyset$, then $a(\bar{x}) = 0$ in line 5 in iteration $\bar{x}$. So, $k - u = a^*(\bar{x}_u) = a^*(\bar{x}) = 0$, i.e., $u = k$. Q.E.D.

We shall call the unique path $x \equiv \bar{x}_0, \bar{x}_1, \cdots, \bar{x}_u \equiv \bar{x}$ from x to $\bar{x}$ (with $u \geq 1$) in Lemma 4.5 the *maximum path* for $\bar{x}$. For any $\bar{x} \in C^*$, the corresponding edge $(x, x') \in I^*$, where $x = A^*(\bar{x})$ and $x' = s^*(x)$. Note that Case 2 and line 10 hold in iteration $\bar{x}_i$ for $0 \leq i \leq u-1$, and Case 1 holds in iteration $\bar{x}_u \equiv \bar{x}$. Also, in Lemma 4.5, instead of saying $a^*(\bar{x}_i) = k-i$, we can say $a(\bar{x}_i) = k - i$ in iteration $\bar{x}_{i-1}$ when line 12 is executed for $1 \leq i \leq u$. $\bar{x}_1, x' \in N^+(x)$ imply that $d(\bar{x}_1, x') \leq 1$ by (3.1) and so $\bar{x}_1$ 1-neighborhood covers (x, x'). This together with that $d(\bar{x}_1, \bar{x}_a) \leq a - 1$ implies that $\bar{x}_a$ a-neighborhood covers (x, x') for $1 \leq a \leq u$.

Theorem 4.2 $|C^*| = |I^*|$.

Proof. By Lemma 4.5, for any two distinct vertices y, $y' \in C^*$, $A^*(y) \neq A^*(y')$, $s(A^*(y)) \neq \infty$, and $s(A^*(y')) \neq \infty$. Hence the theorem holds, because each time a new vertex is added into C in line 6 a new edge is added into I in line 7. Q.E.D.

Lemma 4.6 *If vertex z k-neighborhood covers an edge (x, y), then there exists a shortest x-z path $x \equiv x_0, x_1, \cdots, x_r \equiv z$ such that each x_a $(k-1)$-neighborhood covers (x, y) for $1 \leq a \leq r - 1$.*

Proof. The lemma is easy when $d(x, z) = 1 + d(y, z)$ or $d(y, z) = 1 + d(x, z)$. For the case where $d(x, z) = d(y, z) \leq k$, the lemma follows from the fact that there exists a vertex w such that $d(w, z) = d(x, z) - 1$ and w is adjacent to both x and y (see [9], Lemma 1 (d)). Q.E.D.

Lemma 4.7 *If $x_0, x_1, \cdots, x_r$ is a shortest path ¿from x_0 to x_r in G, then the sequence $x_0, x_1, \cdots, x_r$ is unimodal, i.e., $x_0 < x_1 < \cdots < x_{i-1} < x_i$ and $x_i > x_{i+1} > \cdots > x_r$ for some $0 \leq i \leq r$.*

Proof. We shall prove the lemma by induction on r. The lemma is clearly true for $r = 0$. Suppose $r \geq 1$ and $x_0 < x_1 < \cdots < x_{i-1} < x_i$ and $x_i > x_{i+1} > \cdots > x_{r-1}$ for some

If Case 2 holds but line 10 does not, then either $a(y) < a(w)$ or $b(y) < a(w)$. For the former case, $a^*(y) \leq a(y) < a(w) \leq k$. For the latter case, $b(y) < a(w)$ still holds in iteration y and so $a^*(y) \leq a(y) < b(y) < a(w) \leq k$.

If Case 3 of the algorithm holds, then $b(w) < k$ or $a(w) = b(w) = k$ with $N_0^+(w) = \emptyset$. For the former case, $b(y) \leq k$ by line 17. $b(y) \leq k$ still holds in iteration y and so $a^*(y) \leq a(y) < b(y) \leq k$. For the latter case, by the updating rule in lines 18 and 19, there exists some i such that $w, y \in N_0^+(i)$ and $a(y) = b(y) = k$ in iteration i. Thus $b(y) \leq k$ still holds in iteration y and so $a^*(y) \leq a(y) < b(y) \leq k$. Q.E.D.

Lemma 4.5 *For any $\bar{x} \in C^*$ with $A(\bar{x}) = x$, there exists a unique increasing path $x \equiv \bar{x}_0$, $\bar{x}_1$, $\cdots$, $\bar{x}_u \equiv \bar{x}$ from x to $\bar{x}$ (with $u \geq 1$) such that $\bar{x}_i = \max N^+(\bar{x}_{i-1})$ for $1 \leq i \leq u$ and $A^*(\bar{x}_i) = x$ and $a^*(\bar{x}_i) = k - i$ for $0 \leq i \leq u$. Furthermore, $s^*(x) \neq \infty$ and $A^*(y) \neq A^*(\bar{x})$ for any $y \in C^* - \{\bar{x}\}$. Also, $u = k$ when $N^+(\bar{x}) \neq \emptyset$.*

Proof. Let $u = k - a^*(\bar{x})$ and $x_u = \bar{x}$, i.e., $a^*(\bar{x}_u) = k - u$. By Lemma 4.4, $u \geq 1$. Note that $a^*(\bar{x}_u)$ initially is k and decreases only when Case 2 and line 10 of the algorithm hold in some iteration i and $\bar{x}_u = \max N^+(i)$. There may be several such i. Let $\bar{x}_{u-1}$ be the maximum of all such i. In this case, $\bar{x}_u = \max N^+(\bar{x}_{u-1})$, $A^*(\bar{x}_u) = A^*(\bar{x}_{u-1})$, and $a^*(\bar{x}_{u-1}) = a^*(\bar{x}_u) + 1 = k - (u-1)$. Continuing the same argument, we get an increasing path $x \equiv \bar{x}_0$, $\bar{x}_1$, $\cdots$, $\bar{x}_u \equiv \bar{x}$ such that $\bar{x}_i = \max N^+(\bar{x}_{i-1})$ for $1 \leq i \leq u$, $A^*(\bar{x}_0) = A^*(\bar{x}_1) = \cdots = A^*(\bar{x}_u)$, and $a^*(\bar{x}_i) = k - i$ for $0 \leq i \leq k$. Since $a^*(\bar{x}_0) = k$, $A^*(\bar{x}_0)$ keeps its original value x, i.e., $A^*(\bar{x}_i) = x$ for all $0 \leq i \leq u$.

Furthermore, since $\bar{x}_i$ is uniquely determined by $\bar{x}_{i-1}$, such a path is unique and there is no $y \in C^* - \{\bar{x}\}$ with $A^*(y) = A^*(\bar{x})$.

Since $a^*(x) = k$, $a(x) = k$ at any time. In iteration x, Case 2 and line 10 hold, i.e., $0 < a(x) < b(x)$ with $N^+(x) \neq \emptyset$ or $a(x) = b(x) = k$ with $N_0^+(x) \neq \emptyset$. For the latter case, $s^*(x) \neq \infty$ by line 14. For the former case, $b(x) > k$ in iteration x and so $b(x) > k$ in any previous iteration. By the definition of R in line 18, $x \notin R$ in any previous iteration. Hence by line 19, $N_0^+(x) = N^+(x) \neq \emptyset$ in iteration x. Again, $s^*(x) \neq \infty$.

Theorem 4.1 *C^* is a k-neighborhood covering set of G.*

Proof. For any edge $(x, z) \in E$, without loss of generality we may assume that $x < z$, i.e., $z \in N^+(x)$. We shall prove the theorem according to (1), (2), and (3) of Lemma 4.2.

(1) $b^*(x) \leq a^*(x)$. By Lemma 4.1, there exists some $y \in C^*$ such that $d(x, y) \leq b^*(x) \leq a^*(x) \leq k$. If $d(x, y) \leq k - 1$, then y k-neighborhood covers (x, z). So we may assume that $d(x, y) = b^*(x) = a^*(x) = k$. Since Lemma 4.2 (1) holds only when Case 2 of the algorithm does not, $N_0^+(x) = \emptyset$ in iteration x. By $z \in N^+(x)$ and the updating rule for $N_0^+(j)$ in lines 18 and 19, there exists i such that $x, z \in N_0^+(i)$ and $b(i) = k - 1$ in iteration i. By Lemma 4.1, there exists some $y \in C^*$ such that $d(i, y) \leq b(i) = k - 1$. Hence y $(k-1)$-dominates i and i 1-neighborhood covers (x, z), i.e., y k-neighborhood covers (x, z).

(2) $b^*(j) < a^*(x)$ for some $j \in N^+(x)$. By Lemma 4.1, there exists some $y \in C^*$ such that $d(j, y) \leq b^*(y) < a^*(x) \leq k$, i.e., y $(k-1)$-dominates j. Since $z, j \in N^+(x)$, j 1-neighborhood covers (x, z) by (3.1). So y k-neighborhood covers (x, z).

(3) $a^*(j) < a^*(x)$ for some $j \in N^+(x)$. By Lemma 4.3, there exists some $y \in C^*$ such that $d(j, y) \leq a^*(y) < a^*(x) \leq k$, i.e., y $(k-1)$-dominates j. Since $z, j \in N^+(x)$, j 1-neighborhood covers (x, z) by (3.1). So y k-neighborhood covers (x, z). Q.E.D.

Lemma 4.4 *If $y \in C^*$, then $a^*(y) < k$.*

Proof. Since $y \in C^*$, Case 1 of the algorithm holds in iteration y, i.e., $a(y) = 0$ or $a(y) < b(y)$ with $N^+(y) = \emptyset$. For the former case, clearly $a^*(y) < k$. For the latter case, since G has no isolated vertex, we can choose a largest neighbor w of y. Note that $y \in N^+(w)$. Suppose $N^+(w)$ contains a vertex w' other than y. Then by (3.1) w' is adjacent to y. So w' is a neighbor of y that is larger than w, a contradiction. Thus $N^+(w) = \{y\}$ and $y = \max N^+(w)$. We now consider the following cases in iteration w.

If Case 1 of the algorithm holds, then by line 8 $b(w) = 0$ and so by line 17 $b(y) \leq 1$. Thus $b(y) \leq 1$ in iteration y and so $a^*(y) \leq a(y) < b(y) \leq 1 \leq k$.

If Case 2 and line 10 of the algorithm hold, then $a^*(y) \leq a(y) = a(w) - 1 < k$.

$x \in N^+(i)$ and $b(x) > b(i) + 1$. By the induction hypothesis, there exists some $y \in C$ such that $d(i, y) \leq b(i)$. Therefore, $d(x, y) \leq d(x, i) + d(i, y) \leq 1 + b(i) < b(x)$. Thus the lemma holds. Q.E.D.

Lemma 4.2 *For any vertex $x \in V$, one of the following three statements holds.*

(1) $b^*(x) \leq a^*(x)$.

(2) $b^*(j) < a^*(x)$ for some $j \in N^+(x)$.

(3) $a^*(j) < a^*(x)$ for some $j \in N^+(x)$.

Proof. We shall prove the lemma by considering the following four cases in iteration x.

If Case 1 of the algorithm holds, then $b^*(x) = 0 \leq a^*(x)$, i.e., (1) holds.

If Case 2 and line 10 of the algorithm hold, then there exists $j = \max N^+(x)$ such that $a(j) = a(x) - 1$ and so $a^*(j) \leq a(j) < a(x) = a^*(x)$, i.e., (3) holds.

If Case 2 holds but line 10 does not, then there exists $j \in N^+(x)$ such that $a(j) < a(x)$ or $b(j) < a(x)$. For the former case, $a^*(j) \leq a(j) < a(x) = a^*(x)$, i.e., (3) holds. For the latter case, $b^*(j) \leq b(j) < a(x) = a^*(x)$, i.e., (2) holds.

If Case 3 of the algorithm holds, then $b(x) \leq a(x)$ and so $b^*(x) \leq b(x) \leq a(x) = a^*(x)$, i.e., (1) holds. Q.E.D.

Lemma 4.3 *For any vertex $x \in V$ there exists some vertex $y \in C^*$ such that $d(x, y) \leq a^*(x)$.*

Proof. Repeatedly apply Lemma 4.2 to get a sequence $x \equiv x_0, x_1, \cdots, x_{r-1}, x_r$ such that $x_i \in N^+(x_{i-1})$ for $1 \leq i \leq r$ and

$$b^*(x_r) \leq a^*(x_r) < a^*(x_{r-1}) < \cdots < a^*(x_1) < a^*(x_0) = a^*(x)$$

or

$$b^*(x_r) < a^*(x_{r-1}) < \cdots < a^*(x_1) < a^*(x_0) = a^*(x).$$

Then $r + b^*(x_r) \leq a^*(x)$. By Lemma 4.1, there exists some $y \in C^*$ such that $d(x_r, y) \leq b^*(x_r)$. Hence $d(x, y) \leq d(x, x_r) + d(x_r, y) \leq r + b^*(x_r) \leq a^*(x)$. Q.E.D.

14. $s(i) \longleftarrow min\ N_0^+(i);]$ {where min $\emptyset = \infty$}
15. **Case 3:** $(a(i) \geq b(i)$ **and** $k > b(i))$ **or** $(a(i) = b(i) = k$ **and** $N_0^+(i) = \emptyset)$
16. do nothing;
17. **for all** $j \in N^+(i)$ **do** $b(j) \longleftarrow \min\{b(j), b(i) + 1\}$;
18. $R \longleftarrow \{j \in N_0^+(i) : a(j) = b(j) = b(i) + 1 = k\}$;
19. **for all** $j \in R$ **do** $N_0^+(j) \longleftarrow N_0^+(j) - R$;
20. **end for.**

4 Correctness of the Algorithm

This section proves the correctness of Algorithm CI. First, we note that during the execution of the algorithm, $a(i)$, $b(i)$, $A(i)$, $s(i)$, C, and I are updated. We shall denote their final values by $a^*(i)$, $b^*(i)$, $A^*(i)$, $s^*(i)$, C^*, and I^*, respectively, when the algorithm terminates. Note that $a(i), b(i)$, and $A(i)$ keep decreasing and stay at their final values from the beginning of iteration i. $s(i)$ only changes from ∞ to min $N_0^+(i)$ in iteration i when Case 2 and line 10 of the algorithm hold.

The idea behind our proof of the correctness of Algorithm CI is to prove that C^* is a k-neighborhood covering set (Theorem 4.1), I^* is a k-neighborhood independent set (Theorem 4.3), and $|C^*| = |I^*|$ (Theorem 4.2). If these conditions hold, C^* is a minimum k-neighborhood covering set, I^* is a maximum k-neighborhood independent set, and $\alpha_N(G,k) = \rho_N(G,k)$.

Lemma 4.1 *In any iteration, for any vertex $x \in V$ there exists some vertex $y \in C$ such that $d(x,y) \leq b(x)$. Consequently, for any vertex $x \in V$, there exists some vertex $y \in C^*$ such that $d(x,y) \leq b^*(x)$.*

Proof. We shall prove the lemma by induction on the iteration number. Initially, $b(x) = \infty$ and $C = \emptyset$. The lemma holds if we interpret it as $\min_{y \in C} d(x,y) \leq b(x)$ and $\min \emptyset = \infty$. $b(x)$ only changes its value in lines 8 and 17. If $b(x)$ is reset to 0 in line 8, then x is added into C in line 6. So $d(x,x) = 0 = b(x)$ for $x \in C$. Suppose $b(x)$ is reset to $b(i) + 1$ in line 17 for

case where $a(j) \geq a(i)$ and $b(j) \geq a(i)$ for all $j \in N^+(i)$. Now we need to find a vertex that $(k-1)$-dominates m and so k-neighborhood covers all edges (i,j) with $j \in N_0^+(i)$.

When $a(i) \geq b(i)$ and $k > b(i)$, there is a vertex in the current C that is within distance $a(i)$ from i. This vertex in fact $(k-1)$-dominates i and so k-neighborhood covers all edges incident to i. When $a(i) = b(i) = k$ and $N_0^+(i) = \emptyset$, by the definition of $N_0^+(i)$, any edge (i,j) with $j \in N_0^+(i)$ is k-neighborhood covered by some vertex in C. In these two cases, we do not need to do anything.

Finally, we need to update $b(j)$ and $N_0^+(j)$ for all vertices $j \in N^+(i)$.

We can summarize the procedure described above in the following algorithm:

Algorithm CI.

Input. A strongly chordal graph $G = (V, E)$ without isolated vertices and in which $[1, 2, \cdots, n]$ is a strong elimination order.

Output. A minimum k-neighborhood covering set C and a maximum k-neighborhood independent set I of G.

Method.

1. $C \longleftarrow \emptyset$; $I \longleftarrow \emptyset$;
2. **for all** $i \in V$ **do**
3. $[a(i) \longleftarrow k$; $b(i) \longleftarrow \infty$; $A(i) \longleftarrow i$; $s(i) \longleftarrow \infty$; $N_0^+(i) \longleftarrow N^+(i)]$;
4. **for** $i = 1$ **to** n **do**
5. **Case 1**: $a(i) = 0$ **or** $(a(i) < b(i)$ **and** $N^+(i) = \emptyset)$
6. $C \longleftarrow C \cup \{i\}$;
7. $I \longleftarrow I \cup \{(A(i), s(A(i)))\}$;
8. $b(i) \longleftarrow 0$;
9. **Case 2:** $(0 < a(i) < b(i)$ **and** $N^+(i) \neq \emptyset)$ **or** $(a(i) = b(i) = k$ **and** $N_0^+(i) \neq \emptyset)$
10. **if** $[a(j) \geq a(i)$ **and** $b(j) \geq a(i)$ **for all** $j \in N^+(i)]$
11. **then** $[m \longleftarrow max\ N^+(i)$;
12. $a(m) \longleftarrow a(i) - 1$;
13. $A(m) \longleftarrow A(i)$;

are as follows. For each vertex i, the algorithm is going to put some vertex, which is within distance $a(i)$ ¿from i, into the k-neighborhood covering set C. At any time, there is a vertex in the current C that is within distance $b(i)$ from i. Both $a(i)$ and $b(i)$ keep decreasing as the algorithm proceeds. $a(i)$ decreases in the case where i is the maximum neighbor of a smaller vertex i' that is not properly neighborhood covered by a vertex of C within distance $a(i')$ in iteration i'. In this case, $a(i)$ is set to be $a(i') - 1$. Similarly, in a previous iteration, $a(i')$ is set to be $a(i'') - 1$. Continuing this argument, we find there exists a smallest vertex i^* that forces $\cdots$, i'', i', i to decrease their $a(\cdot)$ values, although $a(i^*)$ never changes. We use $A(i)$ to denote this *initial vertex* i^* from which $a(i)$ decreases. $s(i)$ is the optimal candidate for j such that (i, j) is put into I. $N_0^+(i)$ is a set of candidates for $s(i)$. More precisely, $N_0^+(i)$ contains those $j \in N^+(i)$ such that there is no vertex i' 1-neighborhood covering (i, j) and i' itself is $(k-1)$-dominated by a vertex in the current C. $s(i)$ is chosen to be min $N_0^+(i)$ in iteration i. If in iteration i the algorithm determines that i should be put into C, it will also put the edge $(A(i), s(A(i)))$ into I. Initially, $a(i) = k$, $b(i) = \infty$, $A(i) = i$, $s(i) = \infty$, and $N_0^+(i) = N^+(i)$ for all $i \in V$. The algorithm processes vertex i according to one of the following cases.

When $a(i) = 0$, vertex i is required to be in C. When $a(i) < b(i)$ and $N^+(i) = \emptyset$, all vertices in the current C are farther than the distance $a(i)$ from i and there is no vertex $j > i$ that is adjacent to i. So we need to put i into C. In these two cases, we also put the edge $(A(i), s(A(i)))$ into I. Since now i is in C, $b(i)$ is set to be 0.

When $0 < a(i) < b(i)$ and $N^+(i) \neq \emptyset$, all vertices in the current C are farther than the distance $a(i)$ from i. If $a(j) < a(i)$ or $b(j) < a(i)$ for some $j \in N^+(i)$, then a vertex of C that is within distance $a(j)$ from j shall be within distance $a(i)$ from i. Otherwise, $a(j) \geq a(i)$ and $b(j) \geq a(i)$ for all $j \in N^+(i)$. In this case, we need to find a vertex with distance no more than $a(i) - 1$ from the maximum neighbor m of i, since for any vertex $j > i$ there is a shortest path from j to i that passes m. When $a(i) = b(i) = k$ and $N_0^+(i) \neq \emptyset$, there is one vertex in the current C that is of distance k from vertex i. However, this vertex does not k-neighborhood cover any edge (i, j) with $j \in N_0^+(i)$. Again, we only have to consider the

As in [6], we use $N^+(i)$ for $N_i(i)$ and $N^+[i]$ for $N_i[i]$. A useful property of a chordal graph G, in which $[1, 2, \cdots, n]$ is a perfect elimination order, is that

$$N^+[i] \text{ is a clique for any } i. \tag{3.1}$$

A useful property of a strongly chordal graph G, in which $[1, 2, \cdots, n]$ is a strong elimination order, is that

$$j, l \in N^+[i] \text{ and } i \leq j \leq l \text{ imply } N_i[j] \subseteq N_i[l]. \tag{3.2}$$

(3.2) states that in the graph G_i induced by $\{i, i+1, \cdots, n\}$, the maximum neighbor of i has the largest closed neighborhood among all neighbors of i. So, the maximum neighbor is most powerful as a dominating vertex. This is important in our development below. Also (3.1) and (3.2) are frequently used in the proofs of lemmas and theorems in Section 4.

The idea behind our algorithm for the k-neighborhood covering and the k-neighborhood independence problems is analogous to but much more complicated than that behind the algorithms for the k-domination problem (see [2, 14]). In fact, a $(k-1)$-dominating set is a k-neighborhood covering set. However, the converse is not true. Note that a vertex set C is a k-neighborhood covering set if and only if for any edge e in G either one end vertex of e is $(k-1)$-dominated by some vertex in C or both end vertices of e are of distance exactly k from a same vertex in C. So our algorithm retains the spirit of the $(k-1)$-domination problem with special attention to cases where a *critical edge* e occurs, i.e., both end vertices of e are of distance exactly k from the same vertex in C. To handle critical edges, we employ some of the ideas in [5] for the 1-neighborhood covering and independence.

The algorithm processes vertices in the order 1, 2, $\cdots$, n. Initially the k-neighborhood covering set C and the k-neighborhood independent set I are empty. In iteration i, the algorithm determines if vertex i has to be put into C. If the answer is positive, the algorithm also finds an edge to put into I. After processing, vertex i is deleted ¿from the graph and $i+1$ becomes a simple vertex of the remaining graph.

For technical reasons, we associate each vertex i with two non-negative integers $a(i)$ and $b(i)$, two vertices $A(i)$ and $s(i)$, and a subset $N_0^+(i)$ of $N^+(i)$. The meanings of these items

Suppose I is a minimum 1-neighborhood independence set of G. It must be the case that each edge of I has one end in C and the other end in S. Also $S' = \{s \in S$: there is some $(x, s) \in I\}$ is a 2-independent set, i.e., no two distinct vertices of S' have a common neighbor. So $I' = \{(s_{k-1}, s_k) : s \in S'\}$ is a k-neighborhood independence set of G_k. This gives $\alpha_N(G_k, k) \geq \alpha_N(G, 1)$.

Conversely, suppose I' is a minimum k-neighborhood independence set G_k. Each edge of I' must be of the form (s_{k-1}, s_k) for some $s \in S$. Also $S' = \{s \in S$: some $(s_{k-1}, s_k) \in I'\}$ is a 2-independent set. For any $s \in S'$, choose a neighbor s' of s in C. Then $I = \{(s', s) : s \in S'\}$ is a 1-neighborhood independent set of G. This gives $\alpha_N(G_k, k) \leq \alpha_N(G, 1)$. Together these inequalities imply that $\alpha_N(G_k, k) = \alpha_N(G, 1)$. Q.E.D.

3 The Algorithm on Strongly Chordal Graphs

In this section we set forth a linear time algorithm for the k-neighborhood covering and the k-neighborhood independence problems on a strongly chordal graph G. Without loss of generality, we may assume that G has no isolated vertices. The algorithm in fact gives a k-neighborhood covering set C^* and a k-neighborhood independent set I^* with $|C^*| = |I^*|$. By definitions and the max-min inequality,

$$|C^*| = |I^*| \leq \alpha_N(G, k) \leq \rho_N(G, k) \leq |C^*|.$$

Hence all inequalities are equalities. Consequently, C^* and I^* are optimal and $\alpha_N(G, k) = \rho_N(G, k)$ for any strongly chordal graph G.

Suppose $G = (V, E)$ is a strongly chordal graph in which a strong elimination order $[v_1, v_2, \cdots, v_n]$ is given. Note that this is also a perfect elimination order of the chordal graph G. For simplicity we identify vertex v_i with i and hence $[1, 2, \cdots, n]$ is a strong elimination order. For any vertices i and j, $i \leq j$, let

$N_i(j) = \{l \in V : (j, l) \in E$ and $l \geq i\}$ and

$N_i[j] = N_i(j) \cup \{j\}$.

disjoint union of a clique C and an independent set S. Split graphs are chordal.

Theorem 2.1 *For any fixed positive integer k, the k-neighborhood covering and the k-neighborhood independence problems are NP-complete on chordal graphs.*

Proof. For any split graph $G = (V, E)$, whose vertex set V is the disjoint union of a clique C and an independent set S, construct the graph $G_k = (V_k, E_k)$ by attaching a path $s \equiv s_1$, $s_2, \cdots, s_k$ of length $k-1$ to each vertex s in S. In other words,

$V_k = C \cup \{s_i : s \in S \text{ and } 1 \leq i \leq k\}$, where s_1 is considered to be the same as s,

and

$E_k = E \cup \{(s_i, s_{i+1}) : 1 \leq i \leq k-1\}$.

G_k is clearly a chordal graph. We shall prove that $\rho_N(G_k, k) = \rho_N(G, 1)$ and $\alpha_N(G_k, k) = \alpha_N(G, 1)$. If these two equalities hold, then the theorem follows from the fact that the 1-neighborhood covering and the 1-neighborhood independence problems are NP-complete on split graphs (see [5]).

Suppose D is a minimum 1-neighborhood covering set of G. Any edge in E is 1-neighborhood covered and so k-neighborhood covered by some vertex in D. For any edge $(s_i, s_{i+1}) \in E_k - E$, there exists some $x \in D$ such that $d(x, s) \leq 1$. So $d(x, s_i) \leq i < k$ and $d(x, s_{i+1}) \leq i + 1 \leq k$, i.e., x k-neighborhood covers (s_i, s_{i+1}). Therefore D is a k-neighborhood covering set of G_k. This gives $\rho_N(G_k, k) \leq \rho_N(G, 1)$.

Conversely, suppose D is a minimum k-neighborhood covering set of G_k. We can assume that $D \subseteq C$, otherwise any s_i in D can be replaced by a vertex adjacent to $s \equiv s_1$ in C to form a new minimum k-neighborhood covering set. Consider any edge (x, y) in E. If x and y are both in C, then (x, y) is clearly 1-neighborhood covered by any vertex in D. Suppose $x \in C$ and $y \in S$. Since D is a k-neighborhood covering set of G_k, there exists some vertex z in D that k-neighborhood covers (y_{k-1}, y_k). Both $z \in C$ and $d(z, y_k) \leq k$ imply that z is adjacent to y and so z 1-neighborhood covers (x, y). Therefore D is a 1-neighborhood covering set of G. This gives $\rho_N(G_k, k) \geq \rho_N(G, 1)$. Together these inequalities imply that $\rho_N(G_k, k) = \rho_N(G, 1)$.

known that a graph $G = (V, E)$ is chordal if and only if it has a *perfect elimination order*, i.e., an ordering $[v_1, v_2, \cdots, v_n]$ of V such that each v_i is a simplicial vertex of the subgraph G_i induced by $\{v_i, v_{i+1}, \cdots, v_n\}$ (see [7]).

An *s-sun* (or *incomplete trampoline of order s*) is a chordal graph having $2s$ vertices a_1, $a_2, \cdots, a_s, b_1, b_2, \cdots, b_s$ such that $(a_1, a_2, \cdots, a_s, a_1)$ is a cycle and each b_i has exactly two neighbors a_i and a_{i+1}, where $a_{n+1} = a_1$. A graph is *sun-free* (resp. *odd-sun-free*) if it contains no *s-sun* as a subgraph for all $s \geq 3$ (resp. for all odd $s \geq 3$) (see [3, 4]). Sun-free chordal graphs are called *strongly chordal graphs* in [6]. A vertex is *simple* if the set $\{N[u] : u \in N[v]\}$ can be linearly ordered by inclusion. Farber [6] proved that a graph $G = (V, E)$ is strongly chordal if and only if it has a *simple elimination order*, i.e., an ordering $[v_1, v_2, \cdots, v_n]$ of V such that each v_i is a simple vertex of G_i. A *strong elimination order* is a simple elimination order such that $N_{G_i}[v_j] \subseteq N_{G_i}[v_k]$ whenever $i \leq j \leq k$ and $v_j, v_k \in N_{G_i}[v_i]$. Anstee and Farber [1] presented an $O(|V|^3)$ algorithm, Hoffman, Kolen, and Sakarovitch [8] presented an $O(|V|^3)$ algorithm, Lubiw [11] presented an $O(L \log^2 L)$ algorithm where $L = |V| + |E|$, Paige and Tarjan [12] presented an $O(L \log L)$ algorithm, and Spinrad [15] presented an $O(|V|^2)$ algorithm for finding a strong elimination order of a strongly chordal graph $G = (V, E)$.

In Section 2, we prove that the k-neighborhood covering and the k-neighborhood independence problems are NP-complete on chordal graphs. Section 3 gives a linear time algorithm for computing $\alpha_N(G, k)$ and $\rho_N(G, k)$ of a strongly chordal graph G. Section 4 verifies the correctness of the algorithm. The algorithm in fact gives a k-neighborhood covering set C^* and a k-neighborhood independent set I^* with $|C^*| = |I^*|$. Consequently, C^* and I^* are optimal and $\alpha_N(G, k) = \rho_N(G, k)$ for any strongly chordal graph G.

2 NP-Completeness on Chordal Graphs

In this section we show that for any fixed k the k-neighborhood covering and the k-neighborhood independence problems are NP-complete on chordal graphs. We shall do this by reducing the problems with $k = 1$ on split graphs, which are known to be NP-complete in [5], to the problems on strongly chordal graphs. A graph $G = (V, E)$ is *split* if its vertex set V is the

1 Introduction

All graphs in this paper are simple, i.e., finite, undirected, loopless, and without multiple edges. In a graph $G = (V, E)$, the *length* of a path is the number of edges in the path. The *distance* $d(x, y)$ ¿from vertex x to vertex y is the minimum length of a path from x to y; $d(x, y) = \infty$ if there is no path from x to y. A vertex x *k-dominates* another vertex y if $d(x, y) \leq k$. A vertex z *k-neighborhood covers* an edge (x, y) if $d(z, x) \leq k$ and $d(z, y) \leq k$, or, equivalently, z k-dominates both x and y. A vertex set $C \subseteq V$ is a *k-neighborhood covering set* of G if every edge in E is k-neighborhood covered by some vertex in C. The *k-neighborhood covering number* $\rho_N(G, k)$ of a graph G is the minimum cardinality of a k-neighborhood covering set of G. An edge set $I \subseteq E$ is a *k-neighborhood independent set* of G if no two distinct edges in I are k-neighborhood covered by the same vertex in V. The *k-neighborhood independence number* $\alpha_N(G, k)$ of a graph G is the maximum cardinality of a k-neighborhood independent set of G.

For any graph G and any positive integer k, $\alpha_N(G, k)$ and $\rho_N(G, k)$ are related by the following obvious *max-min inequality*:

$$\alpha_N(G, k) \leq \rho_N(G, k).$$

For $k = 1$, Lehel and Tuza [10] proved that the above inequality in fact is an equality for odd-sun-free chordal graphs G and presented a linear time algorithm for computing $\alpha_N(G, 1) = \rho_N(G, 1)$ for interval graphs. Wu [16] presented an $O(|V|^3)$ algorithm for determining $\rho_N(G, 1) = \alpha_N(G, 1)$ of a strongly chordal graph G. Chang, Farber, and Tuza [5] presented linear time algorithms for computing $\alpha_N(G, 1) = \rho_N(G, 1)$ of a strongly chordal graph G provided that a strong elimination order is known in advance. The purpose of this paper is to study $\alpha_N(G, k)$ and $\rho_N(G, k)$ for chordal graphs.

A graph is *chordal* (or *triangulated*) if every cycle of length greater than three has a chord, which is an edge joining two nonconsecutive vertices in the cycle. In a graph $G = (V, E)$, the (*open*) *neighborhood* $N(v)$ of a vertex v is the set of all vertices adjacent to v and the *closed neighborhood* $N[v] = N(v) \cup \{v\}$. A vertex v is *simplicial* if $N[v]$ is a clique. It is well

ABSTRACT

Suppose $G = (V, E)$ is a simple graph and k is a fixed positive integer. A vertex z k-neighborhood covers an edge (x, y) if $d(z, x) \leq k$ and $d(z, y) \leq k$. A k-neighborhood covering set is a set C of vertices such that every edge in E is k-neighborhood covered by some vertex in C. A k-neighborhood independent set is a set of edges in which no two distinct edges can be k-neighborhood covered by the same vertex in V. In this paper we first prove that the k-neighborhood covering and the k-neighborhood independence problems are NP-complete for chordal graphs. We then present a linear time algorithm for finding a minimum k-neighborhood covering set and a maximum k-neighborhood independent set of a strongly chordal graph.

Keywords. k-neighborhood covering, k-neighborhood independence, chordal graph, strongly chordal graph, strong elimination order

DIMACS Technical Report
January 1993

k-Neighborhood Covering and Independence Problems

by

Shiow-Fen Hwang
Department of Computer Science
Fen Chia University
Taichung 40724, Taiwan
Republic of China

Gerard J. Chang [1]
Department of Applied Mathematics
National Chiao Tung University
Hsinchu 30050, Taiwan
Republic of China
gjchang@cc.nctu.edu.tw

[1]Supported in part by the National Science Council of the Republic of China under grant NSC81-0208-M009-26. Supported in part by DIMACS.

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–88–09648; and also receives support from the New Jersey Commission on Science and Technology.