



ELSEVIER

Discrete Applied Mathematics 63 (1995) 215–222

DISCRETE
APPLIED
MATHEMATICS

Weighted independent perfect domination on cocomparability graphs

Gerard J. Chang^{a,*}, C. Pandu Rangan^b, Satyan R. Coorg^{b,2}

^a Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan

^b Department of Computer Science and Engineering, Indian Institute of Technology, Madras-600 036, India

Received 15 April 1993; revised 29 April 1994

Abstract

Suppose $G = (V, E)$ is a graph in which every vertex $v \in V$ is associated with a cost $c(v)$. This paper studies the weighted independent perfect domination problem on G , i.e., the problem of finding a subset D of V such that every vertex in V is equal or adjacent to exactly one vertex in D and $\sum \{c(v) : v \in D\}$ is minimum. We give an $O(|V||E|)$ algorithm for the problem on cocomparability graphs. The algorithm can be implemented to run in $O(|V|^{2.37})$ time. With some modifications, the algorithm yields an $O(|V| + |E|)$ algorithm on interval graphs, which are special cocomparability graphs.

1. Introduction

A *dominating set* of a graph $G = (V, E)$ is a subset D of V such that every vertex not in D is adjacent to some vertex in D . The concept of domination in graph theory arises naturally from the facility location problem in operations research. Depending on the different requirements of various location problems, domination has many variants, e.g., independent domination, connected domination, total domination, edge domination, k -domination, and perfect domination.

A *perfect dominating set* of a graph $G = (V, E)$ is a subset D of V such that every vertex not in D is adjacent to *exactly one* vertex in D . The *perfect domination problem* is to find a minimum-sized perfect dominating set. Suppose that every vertex $v \in V$ is associated with a cost $c(v)$ and every edge $e \in E$ has a cost $c(e)$. The

* Corresponding author. E-mail: gjchang@math.nctu.edu.tw.

¹ Supported partly by the National Science Council of the Republic of China under grant NSC82-0208-M009-050. This work was done while the author visited DIMACS from July 1992 to June 1993.

² Current address: Laboratory for Computer Science, MIT, Cambridge, MA 02139, USA. E-mail: satyan@abp.lcs.mit.edu.

weighted perfect domination problem is to find a perfect dominating set D such that its total cost

$$\bar{c}(D) = \sum \{c(v): v \in D\} + \sum \{c(u, v): u \notin D, v \in D, \text{ and } (u, v) \in E\}$$

is minimum. Note that the perfect domination problem is just the weighted perfect domination problem with $c(v) = 1$ for each vertex v and $c(e) = 0$ for each edge e . Yen and Lee [25] proved that the perfect domination problem is NP-complete for bipartite graphs and chordal graphs. They also give linear time algorithms for the weighted perfect domination problem on trees [25] and series-parallel graphs [26].

Yen [24] and Yen and Lee [27] also considered the following variants of perfect domination. A perfect dominating set D is *independent*, *connected*, or *total* if the subgraph $G[D]$ induced by D has no edge, is connected, or has no isolated vertex, respectively. They gave NP-complete results of these variants in bipartite graphs and chordal graphs, except for connected domination in chordal graphs. They also gave linear time algorithms for these variants in trees, block graphs, and series-parallel graphs. On the other hand, Chang and Liu [8] gave a linear time algorithm for the weighted connected perfect domination problem in chordal graphs by using clique-tree structures of chordal graphs. They [9] also use the same technic to give linear time algorithms for the weighted perfect domination problem and its three variants in interval graphs. Independent perfect domination was called *efficient domination* by Bange et al. [2], who proved that the efficient domination problem is NP-complete for general graphs and gave a linear time algorithm for the problem on trees. If we consider the application of perfect domination to coding theory, Biggs [6] studied perfect d -codes. A *perfect d -code* of a graph $G = (V, E)$ is a vertex set C such that every vertex $v \in V$ is within distance d of exactly one vertex in D . In conjunction with the study of the interconnection networks used in parallel computers, Livingston and Stout [17] studied *perfect d -dominating sets*, which are precisely the perfect d -codes. The concept of independent perfect domination in this paper is the same concept as their perfect 1-domination.

The main results of this paper are an $O(|V||E|)$ algorithm for the weighted perfect domination problem on a cocomparability graph $G = (V, E)$ and an $O(|V| + |E|)$ time algorithm for the same problem on an interval graph. The algorithm for cocomparability graphs can also be implemented to run in $O(|V|^{2.37})$ time. Note that cocomparability graphs include interval graphs and permutation graphs. Other variants of domination have been extensively studied in interval graphs, permutation graphs, and cocomparability graphs.

For any independent perfect domination set D , $c(D) = \bar{c}(D)$ if $\bar{c}(v) = c(v) + \sum \{c(u, v): (u, v) \in E\}$ for every $v \in V$ and $\bar{c}(e) = 0$ for every $e \in E$. So, for the solution to the weighted independent perfect domination in this paper, without loss of generality, we may assume that $c(e) = 0$ for all $e \in E$. In this case, $c(D) = \sum \{c(v): v \in D\}$.

2. Cocomparability graphs

In this section we give an $O(|V||E|)$ algorithm for the weighted independent domination problem on a cocomparability graph $G = (V, E)$ in which each vertex v is associated with a cost $c(v)$. Domination and its three variants on cocomparability graphs have been studied by Kratsch and Stewart [16] and Arvind and Pandu Rangan [1].

A *comparability graph* is a graph $G = (V, E)$ whose vertex set has a *transitive ordering*, i.e., an ordering of V into $1, 2, \dots, n$ such that

$$i < j < k, (i, j) \in E, \text{ and } (j, k) \in E \text{ imply } (i, k) \in E. \tag{1}$$

There is an $O(|V|^2)$ algorithm [22] to test if a graph is a comparability graph. In the case of a positive answer, the algorithm produces a transitive ordering. A *cocomparability graph* is the complement of a comparability graph, or equivalently, if its vertex set has a *cocomparability ordering*, which is an ordering of V into $1, 2, \dots, n$ such that

$$i < j < k \text{ and } (i, k) \in E \text{ imply } (i, j) \in E \text{ or } (j, k) \in E. \tag{2}$$

In this section we assume that $G = (V, E)$ is a cocomparability graph with a given cocomparability ordering. For technical reasons, we add two isolated vertices 0 and $n + 1$ with $c(0) = c(n + 1) = 0$ to G to obtain a new cocomparability graph, which for simplicity we also call G with a cocomparability ordering $0, 1, 2, \dots, n, n + 1$. Note that D is an independent perfect dominating set of the original graph if and only if $D \cup \{0, n + 1\}$ is an independent perfect dominating set of the new graph. For convenience, we need the following notation, where v is a vertex:

$$\begin{aligned} N(v) &= \{u \in V : (u, v) \in E\}, & N[v] &= \{v\} \cup N(v), & \text{high}(v) &= \max N[v], \\ \text{low}(v) &= \min N[v], & N^+[v] &= \{u \in N[v] : u \geq v\}, \\ d^+(v) &= |N^+[v]|, & N^-[v] &= \{u \in N[v] : u \leq v\}, & d^-(v) &= |N^-[v]|. \end{aligned}$$

Theorem 1. $D = \{0 \equiv v_0 < v_1 < v_2 < \dots < v_r < v_{r+1} \equiv n + 1\}$ is an independent perfect dominating set of cocomparability graph G if and only if the following three conditions hold for all $1 \leq i \leq n + 1$.

- (1) $\text{high}(v_{i-1}) < v_i$.
- (2) $v_{i-1} < \text{low}(v_i)$.
- (3) $\{x \in V : v_{i-1} \leq x \leq v_i\}$ is the disjoint union of $N^+[v_{i-1}]$ and $N^-[v_i]$.

Proof. Suppose D is an independent perfect dominating set of G .

(1) Suppose $\text{high}(v_{i-1}) \geq v_i$. Note that $v_i \neq \text{high}(v_{i-1})$, since D is independent. Then $v_{i-1} < v_i < \text{high}(v_{i-1})$. By (2), either $(v_{i-1}, v_i) \in E$ or $(v_i, \text{high}(v_{i-1})) \in E$. The former case contradicts the assumption that D is an independent set. The latter case contradicts the fact that a vertex not in D is adjacent to exactly one vertex in D . So $\text{high}(v_{i-1}) < v_i$.

(2) Suppose $v_{i-1} \geq \text{low}(v_i)$. Note that $\text{low}(v_i) \neq v_{i-1}$, since D is independent. Then $\text{low}(v_i) < v_{i-1} < v_i$. By (2), either $(\text{low}(v_i), v_{i-1}) \in E$ or $(v_{i-1}, v_i) \in E$. The former case contradicts the fact that a vertex not in D is adjacent to exactly one vertex in D . The latter case contradicts the assumption that D is an independent set. So $v_{i-1} < \text{low}(v_i)$.

(3) For any vertex x such that $v_{i-1} < x < v_i$, we claim that x is adjacent to v_{i-1} or v_i . By the definition of perfect domination, x is adjacent to exactly one v_j . If $j > i$, then $x < v_i < v_j$ and $(x, v_j) \in E$ and $(x, v_i) \notin E$. By (2), $(v_i, v_j) \in E$, which contradicts the assumption that D is an independent set. If $j < i - 1$, then $v_j < v_{i-1} < x$, and $(v_j, x) \in E$ and $(v_{i-1}, x) \notin E$. By (2), $(v_j, v_{i-1}) \in E$, a contradiction again. So x is adjacent to v_{i-1} or v_i . This together with (1) and (2), implies (3).

Conversely, suppose conditions (1)–(3) hold. For any vertex x not in D , assume $v_{i-1} < x < v_i$. First, by (3), x is adjacent to exactly one of v_{i-1} and v_i . Suppose x is adjacent to some other v_j with $j - 1 \geq i$ or $j + 1 \leq i - 1$. If $j - 1 \geq i$, then $\text{low}(v_j) \leq x < v_i \leq v_{j-1}$, which contradicts (2). If $j + 1 \leq i - 1$, then $\text{high}(v_j) \geq x > v_{i-1} \geq v_{j+1}$, which contradicts (1). \square

Theorem 1 can be rewritten in the following form, which is more useful in designing an efficient algorithm to solve the weighted independent perfect domination problem on cocomparability graphs.

Theorem 2. $D = \{0 \equiv v_0 < v_1 < v_2 < \dots < v_r < v_{r+1} \equiv n + 1\}$ is an independent perfect dominating set of a cocomparability graph G if and only if the following four conditions hold for all $1 \leq i \leq n + 1$.

- (1) $\text{high}(v_{i-1}) = \max(\{0, 1, 2, \dots, v_i\} - N[v_i])$.
- (2) $v_{i-1} < \text{low}(v_i)$.
- (3) $N[v_{i-1}] \cap N[v_i] = \emptyset$.
- (4) $d^+(v_{i-1}) + d^-(v_i) = v_i - v_{i-1} + 1$.

Working from Theorem 2, we can derive the following algorithm for finding a weighted independent dominating set of a cocomparability graph.

Algorithm WIPD-CC. Find a weighted independent perfect dominating set of a cocomparability graph.

Input. A cocomparability graph $G = (V, E)$ with a cocomparability ordering $0, 1, \dots, n, n + 1$, in which each vertex v is associated with a weight $c(v)$.

Output. A minimum weighted independent perfect dominating set D of G .

Method.

1. $\text{cost}(0) \leftarrow 0$;
2. **for** $v = 1$ **to** $n + 1$ **do**
3. $\text{cost}(v) \leftarrow \infty$;
4. $h \leftarrow \max(\{0, 1, \dots, v\} - N[v])$;

5. **for all** $u \in N^-[h]$ **satisfying**
 (C1) $\text{high}(u) = h$,
 (C2) $u < \text{low}(v)$,
 (C3) $N[u] \cap N[v] = \emptyset$,
 (C4) $d^+(u) + d^-(v) = v - u + 1$ **do**
6. **if** $(\text{cost}(u) + c(v) < \text{cost}(v))$
7. **then** $\{\text{cost}(v) \leftarrow \text{cost}(u) + c(v); \text{previous}(v) \leftarrow u;\}$
 end do;
- end do;**
8. **if** $\text{cost}(n + 1) = \infty$ **then STOP** since there is no feasible solution;
9. $D \leftarrow \emptyset$;
10. $v \leftarrow \text{previous}(n + 1)$;
11. **while** $(v \neq 0)$ **do** $\{D \leftarrow D \cup \{v\}; v \leftarrow \text{previous}(v);\}$

Theorem 3. *Algorithm WIPD-CC gives a minimum weighted independent perfect dominating set D of a cocomparability graph $G = (V, E)$ in $O(|V||E|)$ time.*

Proof. The correctness of Algorithm WIPD-CC follows from Theorem 2.

Note that the functions high , low , d^+ , and d^- can be calculated in $O(|V| + |E|)$ time. Steps 2–7 of the algorithm take $n + 1$ iterations. To implement step 4 and (C3) of step 5 efficiently, we use an array $\text{mark}[0..n + 1]$, which is set to 0 initially. At iteration v we reset $\text{mark}(x)$ to be v for all $x \in N[v]$. A simple backward do loop for x from v down to 0 will find the first x with $\text{mark}(x) < v$ in $O(|N[v]|)$ time. This is the desired h . For condition (C3) of step 5, we only need to check if $\text{mark}(x) < v$ for all $x \in N[u]$. This takes $O(|N[u]|)$ time. So the do loop from steps 5–7 takes $O(|E|)$ time. Altogether, steps 2–7 take $O(|V||E|)$ time. Steps 8–10 clearly take $O(|V| + |E|)$ time. \square

There is an alternative way to check condition (C3) of step 5, by using matrix multiplication. Consider the *closed neighborhood matrix* $A = (a_{uv})$ of the graph G defined by $a_{uv} = 1$ if and only if $u \in N[v]$. Let $AA = (b_{uv})$. Then $N[u] \cap N[v] = \emptyset$ if and only if $b_{ij} = 0$. If we pre-calculate AA , then condition (C3) of step 5 can be checked in a constant time. Thus, the fastest known algorithm for matrix multiplication to date (see [10]) gives us the following theorem.

Theorem 4. *Algorithm WIPD-CC can be implemented in $O(|V|^{2.37})$ time.*

3. Interval graphs

An *interval graph* is a graph each of whose vertices can be associated with an interval in the real line so that two vertices are adjacent in the graph if and only if the corresponding intervals intersect. Interval graphs have many applications to the real

world (see [11]). One of the most basic problems in the study of interval graphs is the problem of recognizing these graphs. Booth and Lueker [7] gave a linear time algorithm for recognizing an interval graph using PQ -trees. Korte and Möhring [15] simplified the operations on a PQ -tree. Simon [21], Hsu and Ma [13], and Hsu [12] designed new algorithms without using PQ -trees.

Domination and its variants in interval graphs have been studied extensively [3–5, 14, 18–20, 23]. The following vertex ordering methodology is of particular interest from our point of view.

Theorem 5 (Ramalingam and Pandu Rangan [20]). *A graph $G = (V, E)$ is an interval graph if and only if it has an interval ordering, i.e., its vertex set V can be ordered into $[1, 2, \dots, n]$ such that*

$$i < j < k \text{ and } (i, k) \in E \text{ imply } (j, k) \in E. \quad (3)$$

Note that an interval ordering is a cocomparability ordering. A vertex v is said to have a *consecutive neighborhood* if $N[v]$ contains precisely all vertices from $\text{low}(v)$ to $\text{high}(v)$. The property (3) implies that $N^-[v]$ contains precisely all the vertices from $\text{low}[v]$ to v . Therefore Theorem 1 for interval graphs can be rewritten as follows.

Theorem 6. *$D = \{v_1 < v_2 < \dots < v_r\}$ is an independent perfect dominating set of an interval graph G if and only if all v_i have consecutive neighborhoods and V is the disjoint union of all $N[v_i]$ where $1 \leq i \leq r$.*

Theorem 6 suggests the following algorithm for the weighted independent perfect domination problem on interval graphs.

Algorithm WIPD-I. Find a minimum weighted independent perfect dominating set of an interval graph.

Input. An interval graph $G = (V, E)$ with an interval ordering $1, 2, \dots, n$, in which each vertex v is associated with a weight $c(v)$.

Output. A minimum weighted independent perfect dominating set D of G .

Method.

1. find the set C of all vertices having consecutive neighborhoods;
2. $\text{cost}(0) \leftarrow 0$;
3. **for** $v = 1$ **to** n **do**
4. $\text{cost}(v) \leftarrow \infty$;
5. **for all** $u \in N^-[v] \cap C$ **with** $v = \text{high}(u)$ **do**
6. **if** $(\text{cost}(\text{low}(u) - 1) + c(u) < \text{cost}(v))$
7. **then** $\{\text{cost}(v) \leftarrow \text{cost}(\text{low}(u) - 1) + c(u);$
 $\text{previous}(v) \leftarrow \text{low}(u) - 1;$
 $\text{center}(v) \leftarrow u;\}$
- end do;**
- end do;**

8. if $\text{cost}(n) = \infty$ then STOP since there is no feasible solution;
9. $D \leftarrow \emptyset$;
10. $v \leftarrow n$;
11. while $(v \neq 0)$ do $\{D \leftarrow D \cup \{\text{center}(v)\}; v \leftarrow \text{previous}(v)\};$

Theorem 7. Algorithm WIPD-I gives a minimum weighted independent perfect dominating set D of an interval graph $G = (V, E)$ in $O(|V| + |E|)$ time.

Proof. The correctness of Algorithm WIPD-I follows from Theorem 6 and the fact that each $\text{cost}(v)$ is precisely the minimum weighted sum $\sum_{i=0}^j c(v_i)$, where $v_1 < v_2 < \dots < v_j$ are vertices having consecutive neighborhoods and $\{1, 2, \dots, v\}$ is the disjoint union of their closed neighborhoods. The claim about the running time follows from the fact that each iteration v takes $O(|N[v]|)$ time. \square

We illustrate Algorithm WIPD-I by the following example. Consider the interval graph G of 10 vertices in Fig. 1. Assume that $[1, 2, \dots, 10]$ is an interval ordering of G in which each vertex i has a cost $c(i)$ beside it.

Algorithm WIPD-I first decides the set of all vertices having consecutive neighborhoods: $C = \{1, 2, 5, 6, 7, 8, 9, 10\}$. And then compute all $\text{cost}(v)$, $\text{previous}(v)$, and $\text{center}(v)$:

- $\text{cost}(0) = 0$;
- $\text{cost}(1) = \infty$;
- $\text{cost}(2) = 3$, $\text{previous}(2) = 0$, $\text{center}(2) = 1$;
- $\text{cost}(3) = 1$, $\text{previous}(3) = 0$, $\text{center}(3) = 2$;
- $\text{cost}(4) = \infty$;
- $\text{cost}(5) = \infty$;
- $\text{cost}(6) = \infty$;
- $\text{cost}(7) = 4$, $\text{previous}(7) = 2$, $\text{center}(7) = 5$;
- $\text{cost}(8) = 3$, $\text{previous}(8) = 3$, $\text{center}(8) = 7$;
- $\text{cost}(9) = \infty$;
- $\text{cost}(10) = 6$, $\text{previous}(10) = 8$, $\text{center}(10) = 10$.

So, $\{10, 7, 2\}$ is a minimum weighted independent perfect dominating set with total cost 6.

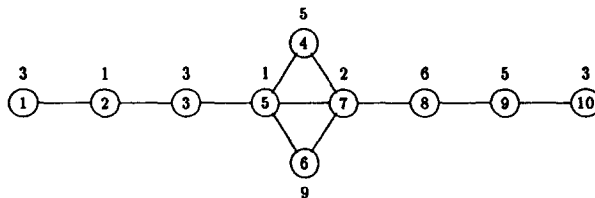


Fig. 1. An interval graph G of 10 vertices.

Acknowledgement

The authors thank an anonymous referee for many useful suggestions on the revision of this paper.

References

- [1] K. Arvind and C. Pandu Rangan, Efficient algorithms for domination problems on cocomparability graphs, Technical Report TR-TCS-90-18, Department of Computer Science and Engineering, Indian Institute of Technology, Madras (1990).
- [2] D.W. Bange, A.E. Barkauskas and P.T. Slater, Efficient dominating sets in graphs, in: R.D. Ringeisen and F.S. Roberts, eds., *Applications of Discrete Mathematics* (SIAM, Philadelphia, PA, 1988) 189–199.
- [3] A.A. Bertossi, Total domination in interval graphs, *Inform. Process. Lett.* 23 (1986) 131–134.
- [4] A.A. Bertossi, On the domatic number of interval graphs, *Inform. Process. Lett.* 28 (1988) 275–280.
- [5] A.A. Bertossi and A. Gori, Total domination and irredundance in weighted interval graphs, *SIAM J. Discrete Math.* 1 (1988) 317–327.
- [6] N. Biggs, Perfect codes in graphs, *J. Combin. Theory Ser. B* 15 (1973) 289–296.
- [7] K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, *J. Comput. System Sci.* 13 (1976) 335–379.
- [8] M.S. Chang and Y.C. Liu, Polynomial algorithms for the weighted perfect domination problems on chordal graphs and split graphs, manuscript (1993).
- [9] M.S. Chang and Y.C. Liu, Polynomial algorithms for the weighted perfect domination problems on interval graphs, manuscript (1993).
- [10] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progression, in: *Proceedings 19th Annual ACM Symposium on Theory of Computing* (1987) 1–6.
- [11] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs* (Academic Press, New York, 1980).
- [12] W.L. Hsu, A simple test for interval graphs, preprint (1991).
- [13] W.L. Hsu and T.H. Ma, Substitution decomposition on chordal graphs and applications, *Lecture Notes in Computer Science* 557 (Springer, Berlin, 1991) 52–60.
- [14] J.K. Keil, Total domination in interval graphs, *Inform. Process. Lett.* 22 (1986) 171–174.
- [15] N. Korte and R.H. Möhring, An incremental linear-time algorithm for recognizing interval graphs, *SIAM J. Comput.* 18 (1989) 68–81.
- [16] D. Kratsch and L. Stewart, Domination on cocomparability graphs, preprint (1989).
- [17] M. Livingston and Q.F. Stout, Perfect dominating sets, *Congr. Numer.* 79 (1990) 187–203.
- [18] T.L. Lu, P.H. Ho and G.J. Chang, The domatic number problem in interval graphs, *SIAM J. Discrete Math.* 3 (1990) 531–536.
- [19] G. Ramalingam and C. Pandu Rangan, Total domination in interval graphs revisited, *Inform. Process. Lett.* 27 (1988) 17–21.
- [20] G. Ramalingam and C. Pandu Rangan, A unified approach to domination problems on interval graphs, *Inform. Process. Lett.* 27 (1988) 271–274.
- [21] K. Simon, A new simple linear algorithms to recognize interval graphs, *Lecture Notes in Computer Science* 553 (Springer, Berlin, 1991) 289–308.
- [22] J. Spinrad, Transitive orientation in (n^2) time, in: *15th STOC Proceedings* (1983) 457–466.
- [23] A. Srinivasa Rao and C. Pandu Rangan, Linear algorithm for domatic number problem on interval graphs, *Inform. Process. Lett.* 33 (1989/1990) 29–33.
- [24] C.C. Yen, Algorithmic aspects of perfect domination, Ph. D. Thesis, Institute of Information Science, National Tsing Hua University, Taiwan (1992).
- [25] C.C. Yen and R.C.T. Lee, The weighted perfect domination problem, *Inform. Process. Lett.* 35 (1990) 295–299.
- [26] C.C. Yen and .C.T. Lee, A linear time algorithm to solve the weighted perfect domination problem in series-parallel graphs, *European J. Oper. Res.*, to appear.
- [27] C.C. Yen and R.C.T. Lee, The weighted perfect domination problem and its variants, manuscript.