# A linear time algorithm for computing a minimum distance total $k$-dominating set in interval graphs

**Abstract**

For a graph $G = (V, E)$, a *total dominating set* is a set $D \subseteq V$ such that every vertex $v \in V$ has a neighbor in $D$. Given a graph $G = (V, E)$ and a fixed positive integer $k$, the *distance total $k$-dominating set* for $G$ is a set $TD$ such that for every vertex in $v \in V$, there exists some vertex $u \in TD$ different from $v$ such that $d_G(u, v) \leq k$, where $d_G(u, v)$ is the distance between $u$ and $v$ in $G$. In this paper, we give a linear time algorithm to compute a minimum distance total $k$-dominating set in interval graphs.

**Keywords:** Domination, total domination, distance total domination, proper interval graph, interval graph.

## 1. Introduction

For a graph $G = (V, E)$, the sets $N_G(v) = \{u \in V | uv \in E\}$ and $N_G[v] = N_G(v) \cup \{v\}$ denote the *neighborhood* and the *closed neighborhood* of a vertex $v$, respectively. A set $D \subseteq V$ of a graph $G = (V, E)$ is a *dominating set* of $G$ if every vertex in $V \setminus D$ is adjacent to a vertex in $D$. The *domination number* of a graph $G$, denoted by $\gamma(G)$, is the minimum cardinality of a dominating set of $G$. The concept of domination has been extensively studied, both in structural and algorithmic graph theory, because of its numerous applications to a variety of areas. Domination naturally arises in facility location problems, in problems involving finding set of representatives, in monitoring communication or electrical networks, and in land surveying. The two books [7, 8] discuss the main results and applications of domination in graphs. Many variants of the basic concepts of domination have appeared in literature due to different practical applications. Among several important variations of domination in graphs, total domination is one of those. A set $D \subseteq V$ of a graph $G = (V, E)$ is a *total dominating set* of $G$ if every vertex in $V$ is adjacent to a vertex in $D$. Equivalently, a set $D \subseteq V$ is a total dominating set of $G$ if $|N_G(v) \cap D| \geq 1$ for every $v \in V$. The *total domination number* of a graph $G$, denoted by $\gamma^t(G)$, is the minimum cardinality of a total dominating set of $G$. Note that a graph without isolated vertices always possesses a total dominating set. Applications and algorithmic aspects of domination and total domination and their variations can be found in [7, 8, 9, 10, 15, 16, 18].

The concepts of domination and total domination are also generalized as follows considering the concept of distance in graphs. The *distance* between two vertices $u$ and $v$ in a graph $G$, denoted by $d_G(u, v)$, is the length of a shortest path between $u$ and $v$ in $G$. Let $k \geq 1$ be an integer. The sets $N_G^k(v) = \{u \in V|\ 0 < d_G(u, v) \leq k\}$ and $N_G^k[v] = N_G^k(v) \cup \{v\}$ denote the *distance $k$-neighborhood* and the *closed distance $k$-neighborhood* of a vertex $v$ in $G$, respectively. Given a graph $G = (V, E)$ and a positive integer $k$, a set $D \subseteq V$ is called a *distance $k$-dominating set* of $G$, if for every vertex $v \in V$, there exists some vertex $u \in D$ such that $d_G(u, v) \leq k$. The *distance $k$-domination number*, denoted by $\gamma_k(G)$ is the minimum cardinality of a distance $k$-dominating set of $G$. Given a graph $G = (V, E)$ and a positive integer $k$, a set $D \subseteq V$ is called a *distance total $k$-dominating set* of $G$, if for every vertex $v \in V$, there exists some vertex $u \in D$ which is different from $v$ such that $d_G(u, v) \leq k$. The *distance total $k$-domination number*, denoted by $\gamma_k^t(G)$ is the minimum cardinality of a distance total $k$-dominating set of $G$. Note that if $k = 1$, then these are usual domination and total domination in graphs.

Since the problem of finding a minimum dominating set and the problem of finding a minimum total dominating set are NP-hard for chordal graphs [1, 2, 13], the problem of finding a minimum distance $k$-dominating set and the problem of finding a minimum distance total $k$-dominating set are also NP-hard for chordal graphs. However, several polynomial time algorithms have been designed for finding a minimum dominating set and also for finding a minimum distance $k$-dominating set in subclasses of chordal graphs [2, 4, 5]. Though several polynomial time algorithms have been obtained for finding a minimum total dominating set in subclasses of chordal graphs [4, 5, 10, 12, 13], a few algorithms are known for finding a minimum distance total $k$-dominating set in subclasses of chordal graphs [18]. Though a linear time algorithm is presented in [5] for finding a minimum total dominating set of strongly chordal graphs, nothing much have been studied on the problem of finding a minimum distance total $k$-dominating set in subclasses of chordal graphs. Recently, Zhao and Shan [18] have presented an $O(n^3)$ time algorithm for finding a minimum distance total $k$-dominating set in block graphs, which is a subclass of strongly chordal graphs.

In this paper, we first present a simple linear time algorithm for finding a minimum distance total $k$-dominating set in proper interval graphs (see Section 3). Then we show the difficulty in applying the same algorithm to find a minimum distance total $k$-dominating set in interval graphs and later on, we present a labelling technique based linear time algorithm in Section 4 for finding a minimum distance total $k$-dominating set in interval graphs.

## 2. Preliminaries

Let $G = (V, E)$ be a graph. For $S \subseteq V$, let $G[S]$ denote the subgraph induced by $G$ on $S$. If $G[C]$, $C \subseteq V$, is a complete subgraph of $G$, then $C$ is called a *clique* of $G$. A clique $C$ of $G$ is called a *maximal clique* of $G$ if no proper superset of $C$ is a clique of $G$. Let $k$ be a positive integer. A vertex $v$ *$k$-dominates* (*totally $k$-dominates*) the vertex $u$ if $u \in N_G^k[v]$ ($u \in N_G^k(v)$).

A graph $G$ is a *chordal graph* if every cycle in $G$ of length at least 4 has a *chord* i.e., an edge joining two non-consecutive vertices of the cycle. Let $\mathscr{F}$ be a nonempty family of sets. A graph $G = (V, E)$ is called an *intersection graph* for a finite family $\mathscr{F}$ of a nonempty set if there is a one-to-one correspondence between $\mathscr{F}$ and $V$ such that two sets in $\mathscr{F}$ have nonempty intersection if and only if their corresponding vertices in $V$ are adjacent. We call $\mathscr{F}$ an *intersection model* of $G$. For an intersection model $\mathscr{F}$, we use $G(\mathscr{F})$ to denote the intersection graph for $\mathscr{F}$. If $\mathscr{F}$ is a family of intervals on a real line, then $G$ is called an *interval graph* for $\mathscr{F}$ and $\mathscr{F}$ is called an *interval model* of $G$. An $O(n + m)$ time algorithm has been given in [3] for recognizing an interval graph and constructing an interval model using PQ-trees. If $\mathscr{F}$ is a family of intervals on a real line such that no interval in $\mathscr{F}$ contains another interval in $\mathscr{F}$ set theoretically, then $G$ is called a *proper interval graph* for $\mathscr{F}$ and $\mathscr{F}$ is called a *proper interval model* of $G$. A vertex $v \in V(G)$ is a *simplicial* vertex of $G$ if $N_G[v]$ is a clique of $G$. An ordering $\alpha = (v_1, v_2, ..., v_n)$ is a *perfect elimination ordering* (PEO) of $G$ if $v_i$ is a simplicial vertex of $G_i = G[\{v_i, v_{i+1}, ..., v_n\}]$ for all $i$, $1 \leq i \leq n$. A PEO $\alpha = (v_1, v_2, \ldots, v_n)$ of a chordal graph is a *bi-compatible elimination ordering* (BCO) if $\alpha^{-1} = (v_n, v_{n-1}, \ldots, v_1)$, i.e. the reverse of $\alpha$, is also a PEO of $G$. This implies that $v_i$ is simplicial in $G[\{v_1, v_2, \ldots, v_i\}]$ as well as in $G[\{v_i, v_{i+1}, ..., v_n\}]$. A graph $G$ is chordal if and only if it has a PEO [6]. Similarly the proper interval graphs are characterized in terms of BCO [11]. In [11], it has also been shown that if $\alpha = (v_1, v_2, \ldots, v_n)$ is a BCO of $G$, then $P :< v_1, v_2, \ldots, v_n >$ is a Hamiltonian path (a path containing all the vertices) of $G$.

## 3. Minimum distance total $k$-dominating set in proper interval graphs

In this section, we present a linear time algorithm for computing a minimum distance total $k$-dominating set in a proper interval graph.

**Lemma 3.1.** *Let $\sigma = (v_1, v_2, \ldots, v_n)$ be a BCO of a proper interval graph $G = (V, E)$. If $v_i v_j \in E$ for $i < j$, then $G[\{v_i, v_{i+1}, \ldots, v_j\}]$ is a clique.*

*Proof.* Since $v_i$ is a simplicial vertex of $G[\{v_i, v_{i+1}, \ldots, v_n\}]$, $v_{i+1}v_j \in E$. Again $v_{i+1}$ is a simplicial vertex of $G[\{v_{i+1}, v_{i+2}, \ldots, v_n\}]$. So $v_{i+2}v_j \in E$. By similar argument, $v_k v_j \in E$ for all $i \leq k \leq j-1$. Since $v_j$ is a simplicial vertex of $G[\{v_1, v_2, \ldots, v_j\}]$ and $v_i v_j \in E$, $v_i v_k \in E$ for all $i+1 \leq k \leq j-1$. Now repeating this process, we can show that $v_k v_{k'} \in E$ for all $i+1 \leq k \leq j-1$ and $i+2 \leq k' \leq j$. So $G[\{v_i, v_{i+1}, \ldots, v_j\}]$ is a clique. □

Let $G = (V, E)$ be a proper interval graph and $\sigma = (v_1, v_2, \ldots, v_n)$ be a BCO of $G$. With respect to $\sigma$, for a vertex $v_i \in V$, we define $F(v_i) = v_j$, where $j = \max\{r \mid v_r v_i \in E \text{ and } r \geq i\}$. In particular, we assume that $F(v_n) = v_n$. We define the notation $F_l(v)$ as follows:

$$F_l(v) = \begin{cases} F(v), & \text{if } l = 1; \\ F(F_{l-1}(v)), & \text{if } l \geq 2. \end{cases}$$

Now we are ready to present the algorithm, namely DistTot-$k$-PIG which computes a minimum distance total $k$-dominating set of a given proper interval graph $G$.

**Algorithm 1:** DistTot-$k$-PIG

**Input**: A proper interval graph $G = (V, E)$;
**Output**: A distance total $k$-dominating set of $G$;
Obtain a BCO $\sigma = (v_1, v_2, \ldots, v_n)$ of $G$;
Initialize $TD = \emptyset$;
**while** $(i \leq n)$ **do**
  **if** $(F_k(v_i) = v_n)$ **then**
    $TD = TD \cup \{v_n, v_{n-1}\}$;
  **else**
    Let $v_l = F_k(v_i)$ and $v_{l'} = F_k(v_l)$;
    $TD = TD \cup \{v_l, v_{l'}\}$;
    $i = l'' + 1$, where $v_{l''} = F_k(v_{l'})$;
  **end**
**end**
Return $TD$;

If $G$ is a proper interval graph such that $F_k(v_1) = v_n$, then it is easy to see that $\{v_n, v_{n-1}\}$ is a distance total $k$-dominating set of $G$. Therefore, we have the following lemma.

**Lemma 3.2.** *Suppose $G$ is a proper interval graph with a BCO $\sigma = (v_1, v_2, \ldots, v_n)$ and $k \geq 1$ is a fixed integer. If $F_k(v_1) = v_n$, then $\{v_n, v_{n-1}\}$ is a minimum distance total $k$-dominating set of $G$.*

**Lemma 3.3.** *Suppose $G$ is a proper interval graph with a BCO $\sigma = (v_1, v_2, \ldots, v_n)$ and $k \geq 1$ is a fixed integer. Let $v_l = F_k(v_1), v_{l'} = F_k(v_l)$ and $v_{l''} = F_k(v_{l'})$. If $D'$ is a minimum distance total $k$-dominating set of $G' = G[\{v_{l''+1}, v_{l''+2}, \ldots, v_n\}]$, then the following conditions are true.*

*(i) $D' \cup \{v_l, v_{l'}\}$ is a distance total $k$-dominating set of $G$;*

*(ii) $\gamma_t^k(G) = \gamma_t^k(G') + 2$.*

*Proof.* Since $\{v_1, v_2, \ldots, v_l, v_{l+1}, \ldots, v_{l'}\} \subseteq N_G^k(v_l) \cup N_G^k(v_{l'})$ and $v_l \in N_G^k(v_{l'})$, $D' \cup \{v_l, v_{l'}\}$ is a distance total $k$-dominating set of $G$ and hence $\gamma_t^k(G) \leq |D'| + 2 = \gamma_t^k(G') + 2$. To prove $(ii)$, we only require to prove that $\gamma_t^k(G) \geq \gamma_t^k(G') + 2$.

Assume that $D$ is a minimum distance total $k$-dominating set of $G$. If $v_l, v_{l'} \in D$, then without loss of generality we can assume that there is no vertex $v_r \in D$ with $1 \leq r \leq l''$ because in that case, we can replace such vertices by some vertices from $G[\{v_{l''+1}, v_{l''+2}, \ldots, v_n\}]$. Now $D \setminus \{v_l, v_{l'}\}$ must be a distance total $k$-dominating set of $G'$. So assume that $v_l \notin D$ or $v_{l'} \notin D$. In these cases, we will construct another minimum distance total $k$-dominating set $D'$ of $G$ that contains $v_l$ and $v_{l'}$.

Let $v_p \in N_G^k(v_1)$ be the minimum indexed vertex such that $v_p$ $k$-dominates $v_1$ in $D$. Again there must a vertex $v_{p'}$ in $D$. Clearly $p \geq 1$. Let $F_i(v_1) = v_{1_i}$ for each $1 \leq i \leq k$. By Lemma 3.1, $\{v_1, v_2 \ldots, F_1(v_1) = v_{1_1}\}, \{v_{1_1}, v_{1_1+1}, \ldots, F_2(v_1) = v_{1_2}\}, \ldots, \{v_{1_{k-1}}, v_{1_{k-1}+1}, \ldots, F_k(v_1) = v_{1_k} = v_l\}$ are cliques, $\{v_1, v_2, \ldots, v_l\} \subseteq N_G^k(v_1)$. So $1 \leq p \leq l$. Since $F_k(v_l) = v_{l'}$, $N_G^k(v_p) \subseteq N_G^k(v_l) \cup N_G^k(v_{l'})$. It can also be verified that $N_G^k(v_{p'}) \subseteq N_G^k(v_l) \cup N_G^k(v_{l'})$.

If $p = l$, then by our assumption $p' \neq l'$. Now $(D \setminus \{v_{p'}\}) \cup \{v_{l'}\}$ is a minimum distance total $k$-dominating set of $G$ containing $v_l$ and $v_{l'}$. Similarly if $p \neq l$ and $p' = l'$, then $(D \setminus \{v_p\}) \cup \{v_l\}$ is a minimum distance total $k$-dominating set of $G$ containing $v_l$ and $v_{l'}$. So assume that $p \neq l$ and $p' \neq l'$. Then $D^* = (D \setminus \{v_p, v_{p'}\}) \cup \{v_l, v_{l'}\}$ is also a minimum distance total $k$-dominating set of $G$. Now we can see that $D^* \setminus \{v_l, v_{l'}\}$ must be a distance total $k$-dominating set of $G$. So, $\gamma_t^k(G') \leq |D^*| - 2 = \gamma_t^k(G) - 2$ which implies $\gamma_t^k(G) \geq \gamma_t^k(G') + 2$ and completes the proof. □

**Theorem 3.4.** *Given a proper interval graph $G = (V, E)$ with $n$ vertices and $m$ edges, the algorithm* DistTot-$k$-PIG *correctly computes a minimum distance total $k$-dominating set in $O(n + m)$ time.*

*Proof.* By Lemma 3.2 and Lemma 3.3, it is clear that the algorithm DistTot-$k$-PIG correctly computes a minimum distance total $k$-dominating set in a proper interval graph. A BCO of $G$ can be computed in $O(n + m)$ time [14]. If $F(u)$ for each $u \in V$ is computed, then we can find $F_k(u)$ for each $u \in V$ by using the recursive definition of $F_k(u)$ which can be done in constant number of steps (as $k$ is fixed positive integer). Since $\{F(u) | u \in V\}$ can be computed in at most $O(n + m)$ time, the algorithm DistTot-$k$-PIG can be executed in at most $O(n + m)$ time. □

Next we explain the difficulty in applying the algorithm DistTot-$k$-PIG in interval graphs. Suppose $G$ is a proper interval graph with a BCO $\sigma = (v_1, v_2, \ldots, v_n)$. The values of $F_k(v)$ which we need to explain are shown in Figure 1 for $k = 2$. The algorithm DistTot-$k$-PIG chooses the vertices $F_2(v_1) = v_l$ and $F_2(v_l) = v_r$ at the first iteration and then the algorithm is applied on the graph $G[\{v_{s+1}, v_{s+2}, \ldots, v_n\}]$. By Lemma 3.1, $G[\{v_1, v_2, \ldots, F_1(v_1)\}], G[\{F_1(v_1), \ldots, F_2(v_1) = v_l\}], G[\{v_l, \ldots, F_1(v_l)\}], G[\{F_1(v_l), \ldots, F_2(v_l) = v_r\}], G[\{v_r, \ldots, F_1(v_r)\}]$ and $G[\{F_1(v_r), \ldots, F_2(v_r) = v_s\}]$ are cliques. So it is clear that $\{v_1, v_2, \ldots, v_s\} \subseteq N_G^2(v_l) \cup N_G^2(v_r)$.

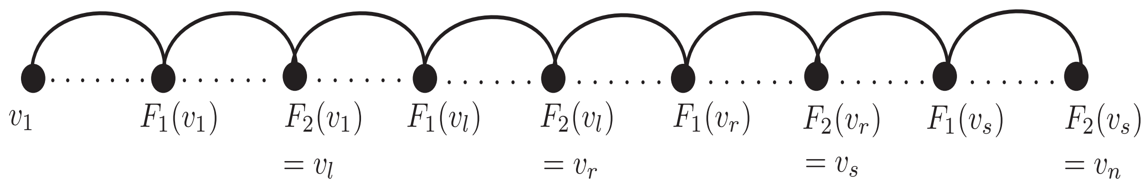


Figure 1: The illustration of difficulty in finding minimum distance total 2-dominating set in a proper interval graph and an interval graph

Now suppose that $G$ is an interval graph with an interval ordering $\sigma = (v_1, v_2, \ldots, v_n)$. If we apply the algorithm DISTTOT-$k$-PIG on $G$, then $F_2(v_1) = v_l$ and $F_2(v_l) = v_r$ will be chosen. In interval graphs, $v_r$ cannot totally 2-dominates all the vertices $\{v_r, v_{r+1}, \ldots, v_s\}$ because there may be vertices in the set $\{F_1(v_r), \ldots, F_2(v_r)\}$ that are not at distance at most 2 from $v_r$. So we need a different technique to handle such cases in interval graphs. We adopt a labelling technique to address this question and finally design an algorithm based on this technique which is presented in Section 4.

## 4. Minimum distance total $k$-dominating set in interval graphs

Recall that an $O(n+m)$ time algorithm has been given in [3] for recognizing an interval graph and constructing an interval model using PQ-trees. Suppose $G$ is an interval graph and $I$ is its interval representation. For every vertex $v_i \in V$, let $I_i$ be the corresponding interval, and let $a_i$ and $b_i$ denote the left endpoint and right endpoint of the interval $I_i$, respectively. We order the vertices of $G$ as $\sigma = (v_1, v_2, \ldots, v_n)$ in increasing order of their right endpoints. It is easy to see that if $v_i v_k \in E$ with $i < k$, then $v_j v_k \in E$ for every $i+1 \leq j \leq k$. We call such an ordering of $G$ as an *interval ordering*. The interval ordering can be computed from the set of maximal cliques of a given interval graph $G = (V, E)$ in linear time [17]. Let $G_i = G[\{v_i, v_{i+1}, \ldots, v_n\}]$ for $1 \leq i \leq n$. If $G$ is connected interval graph, then $G_i$ is also connected. For the sake of simplicity, if not specified, we consider only connected interval graphs.

In an interval graph $G = (V, E)$ with interval ordering $\sigma = (v_1, v_2, \ldots, v_n)$, for a vertex $v_i \in V$, we define $F(v_i) = v_j$, where $j = \max\{r \mid v_r v_i \in E \text{ and } r \geq i\}$. In particular, we assume that $F(v_n) = v_n$. We define the notation $F_l(v)$ as follows:

$$F_l(v) = \begin{cases} F(v), & \text{if } l = 1; \\ F(F_{l-1}(v)), & \text{if } l \geq 2. \end{cases}$$

The following corollary is straightforward from the definition of an interval ordering of an interval graph.

**Corollary 4.1.** *Suppose $\sigma = (v_1, v_2, \ldots, v_n)$ is an interval ordering and $k \geq 2$ is a fixed positive integer. Then $d_{G_i}(F_k(v_i), v_r) \leq k$ for every $i \leq r \leq l$, where $v_l = F_k(v_i)$.*

**Lemma 4.2.** *Suppose $\sigma = (v_1, v_2, \ldots, v_n)$ is an interval ordering and $k$ is a fixed positive integer. If $d_{G_i}(v_i, v_r) \leq k$, then $d_{G_i}(F_k(v_i), v_r) \leq k$.*

*Proof.* Let $F_j(v_i) = v_{i_{F_j}}$ for $1 \leq j \leq k$. Let $P :< v_i = v_{i_1}, v_{i_2}, \ldots, v_{i_r} = v_r >$ be a shortest path between $v_i$ and $v_r$ in $G_i$. Notice that $i_r \leq k$. Now $v_{i_2}$ is adjacent to $F_1(v_i)$. If $i_3 < i_{F_1}$, then $v_{i_3}$ is adjacent to $F_1(v_i)$. If $i_3 > i_{F_1}$, then $v_{i_3}$ is adjacent to $F_1(v_i)$ and $F_2(v_i)$. If $v_{i_4}$ is adjacent to

$F_1(v_i)$, then $P_1 :< v_i, F_1(v_i), v_{i_4}, \ldots, v_r >$ is a shorter path than $P$ which is a contradiction. So $i_4 > i_{F_1}$. If $i_3 < i_{F_1}$, then $v_{i_4}$ is adjacent to $F_1(v_i)$ as $\sigma$ is an interval ordering. So $i_{F_1} < i_3 < i_{F_2}$. If $v_{i_5}$ is adjacent to $i_{F_2}$, then $P_2 :< v_i, F_1(v_i), F_2(v_i), v_{i_5}, \ldots, v_r >$ is a shorter path than $P$ which is a contradiction. So $i_5 > i_{F_2}$. If $i_4 < i_{F_2}$, then $v_{i_5}$ is adjacent to $i_{F_2}$ as $\sigma$ is an interval ordering. So $i_{F_2} < i_4 < i_{F_3}$. Similarly we can prove that $i_{F_{j-2}} < i_j < i_{F_{j-1}}$. Therefore, we get that either $v_r$ is adjacent to $F_k(v_i)$ or $P^* =< v_i, F_1(v_i), F_2(v_i), F_3(v_i), \ldots, v_r >$, a path of length at most $k$ (this path does not contain the vertex $F_k(v_i)$). □

From the above lemma, we have the following corollary.

**Corollary 4.3.** *Suppose $\sigma = (v_1, v_2, \ldots, v_n)$ is an interval ordering and $k$ is a fixed positive integer. If $d_{G_i}(v_r, v_i) = k$, then $v_r$ is adjacent to $F_{k-1}(v_i)$.*

We are now ready to describe our algorithm DistTot-$k$-Interval that computes a minimum distance total $k$-dominating set in a given interval graph $G$. The algorithm DistTot-$k$-Interval uses two arrays $D$ and $L$. Initially $D[v_i] = 0$ and $L[v_i] = 0$ for all $1 \leq i \leq n$. The array $D$ is used to keep track of whether a vertex is already totally $k$-dominated or yet to be totally $k$-dominated. In particular, if $D[v] = 1$ at the end of some iteration of the algorithm, then $v$ is already totally $k$-dominated by the so far constructed subset of the distance total $k$-dominating set to be constructed by the algorithm. The array $L$ is used to keep track of the selected vertices by the algorithm. At the end of the algorithm, $TD = \{v \in V(G) | L[v] = 2\}$, the set consisting of all the vertices whose $L$ labels are 2 forms a minimum distance total $k$-dominating set of $G$. During the algorithm $L[v]$ is either 0, or 1 or 2. If $L[v] = 1$, then the vertex $v$ belongs to the distance total $k$-dominating set to be constructed by the algorithm. If $L[v] = 2$, then the vertex $v$ belongs to the distance total $k$-dominating set to be constructed by the algorithm and a vertex $u \in N_G^k(v)$ has also been found by then.

The following lemma is straightforward from the algorithm DistTot-$k$-Interval.

**Lemma 4.4.** *At starting of the $i$-th iteration of the algorithm* DistTot-$k$-Interval*, the following are true:*

*(i)* $D[v_l] = 1$ *for all* $1 \leq l \leq i-1$*;*

*(ii)* $L[v_l] = 0$ *or* 2 *for all* $1 \leq l \leq i-1$.

The algorithm DistTot-$k$-Interval processes vertex $v_i$ with respect to an interval ordering $\sigma = (v_1, v_2, \ldots, v_n)$ at the $i$-th iteration. Let $TD_i = \{v | L[v] > 0\}$ at the end of $i$-th iteration, $1 \leq i \leq n$.

We now present some lemmas which will be used in proving the correctness of the algorithm DistTot-$k$-Interval.

**Lemma 4.5.** *Assume that $TD_{i-1}$ is contained in some minimum distance total $k$-dominating set $D'$ of $G$. If $D[v_i] = 0, L[v_i] = 0$ and $F_k(v_i) \neq v_i$, then there is a minimum distance total $k$-dominating set $D^*$ of $G$ containing $TD_{i-1} \cup \{F_k(v_i)\}$.*

*Proof.* If $F_k(v_i) \in D'$, then we are done. So assume that $F_k(v_i) \notin D'$. Let $p$ be minimum index such that $v_p \in D'$ and $v_p$ $k$-dominates $v_i$. As $L[v_i] = 0$, $p \neq i$. Since $D'$ is a distance total $k$-dominating set of $G$, there is a vertex $v_{p'} \in N_G(v_p)$ such that $v_{p'} \in D'$. If $v_p = F_k(v_i)$ or $v_{p'} = F_k(v_i)$, then we are done. So assume that $v_p \neq F_k(v_i)$ or $v_{p'} \neq F_k(v_i)$. Since $F_k(v_i) \neq v_i$, $i \neq n$. Let $F_k(v_i) = v_l$.

**Algorithm 2:** DistTot-$k$-Interval

**Input**: An interval graph $G = (V, E)$;
**Output**: A distance total $k$-dominating set of $G$;
Obtain an interval ordering $\sigma = (v_1, v_2, \ldots, v_n)$;
Initialize the arrays $D$ and $L$ such that $D[v_i] = 0$ and $L[v_i] = 0$ for all $v_i; 1 \leq i \leq n$;
**for** $i = 1$ **to** $n$ **do**
    **if** $(D[v_i == 0])$ **then**
        **if** $(F_k(v_i) \neq v_i$ *and* $L[v_i] == 0)$ **then**
            $L[F_k(v_i)] = 1$;
            $D[u] = 1$ for every $u \in N^k(F_k(v_i))$;
        **else if** $(F_k(v_i) \neq v_i$ *and* $L[v_i] == 1)$ **then**
            $L[F_k(v_i)] = 2$, $L[v_i] = 2$;
            $D[u] = 1$ for every $u \in N^k[F_k(v_i)]$;
        **else if** $(F_k(v_i) = v_i$ *and* $L[v_i] == 0)$ **then**
            $L[v_i] = 2$ and $L[w] = 2$, where $w \in N^k(v_i)$ such that $L[w] = 0$;
        **else if** $(F_k(v_i) = v_i$ *and* $L[v_i] == 1)$ **then**
            $L[v_i] = 2$ and $L[w] = 1$, where $w \in N^k(v_i)$ such that $L[w] = 0$;
        **end**
    **end**
**end**
Return $TD_k = \{v \in V | L[v] = 2\}$;

First assume that $p < i$. We first prove that $N^k_{G_i}(v_p) \subseteq N^k_{G_i}(F_k(v_i))$. Let $v_r \in N^k_{G_i}(v_p)$. Let $r'$ be the first index of the path between $v_r$ and $v_p$ such that $r' > i$. Then $v_i$ is adjacent to $v_{r'}$ and hence $d_{G_i}(v_i, v_r) \leq d_G(v_r, v_p) \leq k$. By Corollary 4.3, $v_r$ must be adjacent to one of the vertex from $\{F_1(v_i), F_2(v_i), \ldots, F_{k-1}(v_i)\}$. So $v_r \in N^k_{G_i}(F_k(v_i))$. Recall that since $D'$ is a distance total $k$-dominating set of $G$, there is a vertex $v_{p'} \in N^k_G(v_p)$ such that $v_{p'} \in D'$ and $v_{p'} \neq F_k(v_i)$. If $p' = i$, then $(D' \setminus \{v_p\}) \cup \{F_k(v_i)\}$ is a minimum distance total $k$-dominating set of $G$. If $p' < i$, then by Lemma 4.4, $v_{p'} \notin TD_{i-1}$. Again $d_{G_i}(v_s, F_k(v_i)) \leq d_G(v_{p'}, v_s)$ for every $v_s \in N^k_{G_i}(v_{p'})$. Also, $v_i \notin D'$; otherwise $(D' \setminus \{v_p, v_{p'})\cup \{F_k(v_i)\}\}$ is a smaller distance total $k$-dominating set of $G$. Now $(D' \setminus \{v_p, v_{p'}\}) \cup \{v_i, F_k(v_i)\}$ is a minimum distance total $k$-dominating set of $G$. If $p' > i$, then $v_{p'} \in N^k_{G_i}(F_k(v_i))$ and hence $(D' \setminus \{v_p\}) \cup \{F_k(v_i)\}$ is a minimum distance total $k$-dominating set of $G$.

Next assume that $p > i$. By Corollary 4.3, $v_p$ is adjacent to one of the vertex of $\{F_1(v_i), F_2(v_i), \ldots, F_{k-1}(v_i)\}$. This means $d_{G_i}(v_p, F_k(v_i)) \leq k$. We now have to prove that $N^k_{G_i}(v_p) \subseteq N^k_{G_i}(F_k(v_i))$. Let $v_r \in N^k_{G_i}(v_p)$. If $i \leq r \leq l$ (we assume $v_l = F_k(v_i)$ for better understanding of indices), then by Corollary 4.1, $v_r \in N^k_{G_i}(F_k(v_i))$. So If $r > l$ and $p > l$, then by Lemma 4.2, $d_{G_l}(v_r, v_{l'}) \leq k$, where $v_{l'} = F_k(F_k(v_i))$. If $r > l$ and $p < l$, then let $r'$ be the first index of the path between $v_r$ and $v_p$ such that $r' > l$. Then $F_k(v_i)$ is adjacent to $v_{r'}$ and hence $d_{G_l}(F_k(v_i), v_r) \leq d_G(v_r, v_p) \leq k$. By Corollary 4.3, $v_r$ must be adjacent to one of the vertex from $\{F_1(F_k(v_i)), F_2(F_k(v_i)), \ldots, F_{k-1}(F_k(v_i))\}$. So $v_r \in N^k_{G_i}(F_k(v_i))$. Now if $p' = i$, then $(D' \setminus \{v_p\}) \cup \{F_k(v_i)\}$ is a minimum distance total $k$-dominating set of $G$. If $p' < i$, then by Lemma 4.4, $v_{p'} \notin TD_{i-1}$. Again $d_{G_i}(v_s, F_k(v_i)) \leq d_G(v_{p'}, v_s)$ for every $v_s \in N^k_{G_i}(v_{p'})$. Also, $v_i \notin D'$; otherwise $(D' \setminus \{v_p, v_{p'})\cup \{F_k(v_i)\}\}$ is a smaller distance total $k$-dominating set of $G$. Now $(D' \setminus \{v_p, v_{p'}\}) \cup \{v_i, F_k(v_i)\}$ is a minimum distance total $k$-dominating set of $G$. If $p' > i$, then $v_{p'} \in N^k_{G_i}(F_k(v_i))$ and hence $(D' \setminus \{v_p\}) \cup \{F_k(v_i)\}$ is a minimum distance total $k$-dominating set of $G$. □

**Lemma 4.6.** *Assume that $TD_{i-1}$ is contained in some minimum distance total $k$-dominating set $D'$ of $G$. If $D[v_i] = 0, L[v_i] = 1$ and $F_k(v_i) \neq v_i$, then there is a minimum distance total $k$-dominating set $D^*$ of $G$ containing $TD_{i-1} \cup \{F_k(v_i)\}$.*

*Proof.* Since $L[v_i] = 1$, $v_i \in TD_{i-1}$ and hence $v_i \in D'$. If $F_k(v_i) \in D'$, then we are done. So assume that $F_k(v_i) \notin D'$. Since $D'$ is a distance total $k$-dominating set of $G$, there is a vertex $v_{i'} \in N_G^k(v_i)$ such that $v_{i'} \in D'$. Clearly $v_{i'} \neq F_k(v_i)$. Since $L[v_i] = 1$, by Lemma 4.4, $v_{i'} \notin TD_{i-1}$.

First assume that $i' < i$. As we proved in proof of Lemma 4.5, we can prove that $N_{G_i}^k(v_{i'}) \subseteq N_{G_i}^k(F_k(v_i))$. Now assume that $i' > i$. By Lemma 4.2, $N_{G_i}^k(v_{i'}) \subseteq N_{G_i}^k(F_k(v_i))$. So in any case, $(D' \setminus \{v_{i'}\}) \cup \{F_k(v_i)\}$ is a minimum distance total $k$-dominating set of $G$. □

**Lemma 4.7.** *Assume that $TD_{i-1}$ is contained in some minimum distance total $k$-dominating set $D'$ of $G$. If $D[v_i] = 0, L[v_i] = 0$ and $F_k(v_i) = v_i$, then there is a minimum distance total $k$-dominating set $D^*$ of $G$ containing $TD_{i-1} \cup \{v_i, w\}$, where $w \in N^k(v_i)$ such that $L[w] = 0$.*

*Proof.* Since $F_k(v_i) = v_i$, we have $i = n$. Let $v_p$ be the minimum indexed vertex in $D'$ such that $v_p$ $k$-dominates $v_i$. As $L[v_i] = 0$, by Lemma 4.4, $v_p \notin TD_{i-1}$. Since $D'$ is a distance total $k$-dominating set of $G$, there is a vertex $v_{p'} \in N_G^k(v_p)$ such that $v_{p'} \in D'$. By Lemma 4.4, $v_{p'} \notin TD_{i-1}$. As $i = n$, $N_{G_i}^k(v_p) \cup N_{G_i}^k(v_{p'}) = \{v_i\}$. Let $D^* = (D' \setminus \{v_p, v_{p'}\}) \cup \{v_i, w\}$, where $w \in N_G^k(v_i)$ such that $L[w] = 0$. Note that such a vertex exists as $L[v_i] = 0$ and $D[v_i] = 0$. So $D^*$ is a minimum distance total $k$-dominating set of $G$. □

**Lemma 4.8.** *Assume that $TD_{i-1}$ is contained in some minimum distance total $k$-dominating set $D'$ of $G$. If $D[v_i] = 0, L[v_i] = 1$ and $F_k(v_i) = v_i$, then there is a minimum distance total $k$-dominating set $D^*$ of $G$ containing $TD_{i-1} \cup \{w\}$, where $w \in N^k(v_i)$ such that $L[w] = 0$.*

*Proof.* Since $F_k(v_i) = v_i$, we have $i = n$. As $L[v_i] = 1$, $v_i \in TD_{i-1}$ and hence $v_i \in D'$. Since $D'$ is a distance total $k$-dominating set of $G$, there is a vertex $v_{i'} \in N_G^k(v_i)$ such that $v_{i'} \in D'$. By Lemma 4.4, $v_{i'} \notin TD_{i-1}$. As $i = n$, $N_{G_i}^k(v_{i'}) = \{v_i\}$. Let $D^* = (D' \setminus \{v_{i'}\}) \cup \{w\}$, where $w \in N_G^k(v_i)$ such that $L[w] = 0$. Note that such a vertex exists as $L[v_i] = 1$. So $D^*$ is a minimum distance total $k$-dominating set of $G$. □

**Theorem 4.9.** *Given an interval graph $G = (V, E)$ with $n$ vertices and $m$ edges, the algorithm* DistTot-$k$-Interval *correctly computes a minimum distance total $k$-dominating set in $O(n+m)$ time.*

*Proof.* Recall that $TD_i = \{v | L[v] > 0\}$, $1 \leq i \leq n$ is the set computed by the algorithm DistTot-$k$-Interval at the end of $i$-th iteration. To show that algorithm DistTot-$k$-Interval correctly computes a minimum distance total $k$-dominating set of an interval $G$, it is sufficient to prove that $TD = TD_n$ is a minimum distance total $k$-dominating set of $G$. By Lemma 4.4, $TD_n$ is a distance total $k$-dominating set of $G$. To prove that $TD_n$ is minimum, we show by induction that the statement that there is a minimum distance total $k$-dominating set of $G$ containing $TD_i$ is true for all $i, 0 \leq i \leq n$. So at the termination of the algorithm $TD_n$ is a minimum distance total $k$-dominating set of $G$. Since $TD_0 = \emptyset$, the base case, i.e. there is a minimum distance total $k$-dominating set of $G$ containing $TD_0$ is trivially true. Assume that the statement that there is a minimum distance total $k$-dominating set of $G$ containing $TD_{i-1}, i > 1$.

The algorithm DistTot-$k$-Interval processes the vertex $v_i$ at the $i$-th iteration.

If $D[v_i] = 0, L[v_i] = 0$ and $F_k(v_i) \neq v_i$, then $L[F_k(v_i)]$ is made 1 by the algorithm DistTot-$k$-Interval. In this case, $TD_i = TD_{i-1} \cup \{F_k(v_i)\}$. By Lemma 4.5, $TD_i$ is contained in some minimum distance total $k$-dominating set of $G$. So the induction is true in this case.

If $D[v_i] = 0, L[v_i] = 1$ and $F_k(v_i) \neq v_i$, then $L[v_i]$ and $L[F_k(v_i)]$ are made 2 by the algorithm DistTot-$k$-Interval. In this case, $TD_i = TD_{i-1} \cup \{F_k(v_i)\}$. By Lemma 4.6, $TD_i$ is contained in some minimum distance total $k$-dominating set of $G$. So the induction is true in this case.

If $D[v_i] = 0, L[v_i] = 0$ and $F_k(v_i) = v_i$, then $L[v_i]$ and $L[w]$, where $w \in N_G^k(v_i)$ such that $L[w] = 0$ are made 2 by the algorithm DistTot-$k$-Interval. In this case, $TD_i = TD_{i-1} \cup \{v_i, w\}$. By Lemma 4.7, $TD_i$ is contained in some minimum distance total $k$-dominating set of $G$. So the induction is true in this case.

If $D[v_i] = 0, L[v_i] = 1$ and $F_k(v_i) = v_i$, then $L[v_i]$ and $L[w]$, where $w \in N_G^k(v_i)$ such that $L[w] = 0$ are made 2 by the algorithm DistTot-$k$-Interval. In this case, $TD_i = TD_{i-1} \cup \{w\}$. By Lemma 4.8, $TD_i$ is contained in some minimum distance total $k$-dominating set of $G$. So the induction is true in this case.

Now we discuss the running time of the algorithm DistTot-$k$-Interval. An interval ordering of an interval graph $G$ can be computed in $O(n+m)$ time [17]. If $F(u)$ for each $u \in V$ is computed, then we can find $F_k(u)$ for each $u \in V$ by using the recursive definition of $F_k(u)$ which can be done in constant number of steps (as $k$ is fixed positive integer). Since $\{F(u)|u \in V\}$ can be computed in at most $O(n + m)$ time, the algorithm DistTot-$k$-Interval can be executed in at most $O(n + m)$ time. □

## References


[1] A. A. Bertossi, Dominating sets for split and bipartite graphs. Inform. Process. Lett. 19 (1984) 37–40.

[2] K. S. Booth and J. H. Johnson, Dominating sets in chordal graphs, SIAM J. Comput. 11 (1982) 191–199.

[3] K. S. Booth and G. S. Lueker, Testing for consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms, J. Comput. Syst. Sci. 13 (1976) 335-379.

[4] G. J. Chang, Algorithmic aspects of domination in graphs, in: D. Z. Du and P. M. Pardalos (Eds.), Handbook of combinatorial optimization, Vol 3., Kluwer Acad. Publ., Boston, MA, 1998.

[5] G. J. Chang, Labeling algorithms for domiantio problem in sun-free chordal graphs, Discrete Appl. Math. 22 (1988/89) 21–34.

[6] D. R. Fulkerson and O. A. Gross, Incidence matrices and interval graphs, Pacific J. Math. 15 (1965) 835–855.

[7] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, editors, Domination in graphs: Advanced topics, Volume 209 of Monographs and Textbooks in Pure and Applied Mathematics, Marcel Dekker Inc., New York, 1998.

[8] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, Fundamentals of domination in graphs, Volume 208 of Monographs and Textbooks in Pure and Applied Mathematics, Marcel Dekker Inc., New York, 1998.

[9] M. A. Henning and A. Yeo, Total domination in graphs, Springer Monograph in Mathematics, Springer, 2013.

[10] M. A. Henning, A survey of selected recent results on total domination in graphs, Discrete Math. 309 (2009) 32–63.

[11] R. E. Jamison and R. Laskar, Elimination orderings of chordal graphs, In Combinatorics and applications (Calcutta, 1982), ISI, Calcutta, (1984), 192–200.

[12] J. K. Lan and G. J. Chang, On the algorithmic complexity of $k$-tuple total domination, Discrete Appl. Math. (2014) DOI: http://dx.doi.org/10.1016/j.dam.2014.04.007

[13] R. Laskar, J. Pfaff, S. M. Hedetniemi and S. T. Hedetniemi, On the algorithmic complexity of total domination, SIAM J. Alg. Disc. Math. 5 (3) (1984) 420–425.

[14] B. S. Panda and S. K. Das, A linear time recognition algorithm for proper interval graphs, Inform. Process. Lett. 87 (2003) 153–161.

[15] D. Pradhan, Algorithmic aspects of $k$-tuple total domination in graphs, Inform. Process. Lett. 112 (2012) 816–822.

[16] D. Pradhan, Complexity of certain functional variants of total domination in chordal bipartite graphs, Discrete Math. Algorithms Appl. 4 (2012) 1250045, 19pp.

[17] G. Ramalingam and C. Pandu Rangan, A unified approach to domination problems in interval graphs, Inform. Process. Lett. 27 (1998) 271-274.

[18] Y. Zhao and E. Shan, An efficient algorithm for distance total domination in block graphs, J. Comb. Optim. 2014