# Independent Domination on Graphs with Forbidden Vertices

**William Chung-Kung Yen**
Department of Information Management, Shih Hsin University, Taipei, Taiwan
Email: ckyen001@ms7.hinet.net

## Abstract

Let $G(V, E, W)$ be a graph with $n$ vertices and $m$ edges. Each vertex $v$ is associated with a cost $W(v)$. For any $Q \subseteq V$, the summation cost and the bottleneck cost of $Q$ are defined as $\delta(Q) = \sum_{v \in Q} W(v)$ and $\beta(Q) = \max_{v \in Q}\{W(v)\}$, respectively. Finding an independent dominating set $D$ such that $\delta(D)$ or $\beta(D)$ is minimized are two important fundamental graph problems, called the Weighted Minimum Independent Dominating Set problem (the WMIDS problem) and the Bottleneck Independent Dominating Set problem (the BIDS problem), respectively. This paper studies the WMIDS problem and the BIDS problem such that all vertices in $F$ cannot be included in any feasible solution, where $F$ is a given subset of $V$, called forbidden vertices. The resulting problems are called the RWMIDS problem and RBIDS problem, respectively. When $W(v) = 1$, for all $v \in V$, the WMIDS problem becomes the MIDS problem and RWMIDS problem becomes the RMIDS problem. The MIDS problem can be solved in linear-time on chordal graphs. We first prove that the RMIDS problem is NP-Hard on chordal graphs and the RWMIDS problem can be solved in $O(m)$-time on split graphs. An $O(n)$-time algorithm for the RWMIDS problem on trees is also proposed. Finally, we show that the RBIDS problem is linear-time equivalent to the BIDS problem on all classes of graphs.

**Keywords**: summation cost v.s. bottleneck cost, domination on graphs with forbidden vertices, independent dominating sets

## 1. Introduction

Let $G(V, E, W)$ be an undirected graph with $n$-vertex-set $V$ and $m$-edge-set $E$, and each vertex $v$ is associated with a cost $W(v)$. A subset $D$ of $V$ *dominates* $V$ if there exists $u \in D$ such that $(v, u) \in E$ for each $v \in (V - D)$ and $D$ is called a *dominating set* of $G$. If $(u, v) \notin E$, for all $u, v \in D$, then $D$ is called an *independent dominating set*. Domination on graphs has received extensive research efforts due to its theoretical importance and wide-area applications. Most of previous studies on domination problems are to find a certain kind of dominating set $D$ such that the *summation cost*, $\delta(D) = \sum_{v \in D} W(v)$, is minimized [2--6, 10--13, 15, 17, 18]. Meanwhile, we proposed an original study on finding a dominating set (an independent dominating set) $D$ such that the bottleneck cost, $\beta(D) = \max_{v \in D}\{W(v)\}$, is minimized [19, 20]. The problems of finding an independent dominating $D$ such that $\delta(D)$ ($\beta(D)$) is minimized are called *the Weighted Minimum Independent Dominating Set problem* (*the WMIDS problem*) and *the Bottleneck Independent Dominating Set problem* (*the BIDS problem*), respectively. We consider the following two more general problems.

**The Restricted Weighted Minimum Independent Dominating Set problem (the RWMIDS problem):** Given a graph $G(V, E, W)$ and $F \subseteq V$, find an independent dominating set $D$ such that $\delta(D)$ is minimized and $D \subseteq (V - F)$.

**The Restricted Bottleneck Independent Dominating Set problem (the RBIDS problem):** Given a graph $G(V, E, W)$ and $F \subseteq V$, find an independent dominating set $D$ such that $\beta(D)$ is minimized and $D \subseteq (V - F)$.

The set $F$ in the above two problems indicates that all vertices in $F$ cannot be included in any feasible solution and they are called *forbidden vertices*. Meanwhile, when $W(v) = 1$, for all $v \in V$, the WMIDS problem becomes MIDS problem and RWMIDS problem becomes the RMIDS problem. The WMIDS problem is a well-known NP-Hard problem. It has been known that the MIDS problem can be solved in linear-time on chordal graphs [5]. This paper will first prove that the RMIDS problem is NP-Hard on chordal graphs, but the RWMIDS problem is $O(m)$-time solvable on split graphs. Second, an $O(n)$-time algorithm for the RWMIDS problem on trees will be proposed. Finally, we will claim that the RBIDS problem is linear-time equivalent to the BIDS problem on all classes of graphs.

## 2. The RMIDS Problem on Chordal Graphs and Split Graphs

In the rest of this paper, an independent dominating set $D$ containing no forbidden vertex will be called a *restricted independent dominating set* (a *RID set*). An edge $(u, v)$ of a graph $G$ is called a *chord* of a cycle $\Omega$ of $G$ if $u$ and $v$ are two nonconsecutive vertices in $\Omega$. $G$ is called a *chordal graph* if each cycle with length greater than three has a chord [8, 9, 16]. To show that the RMIDS problem is NP-Hard on chordal graphs, the following NP-complete problem is used [7].

**The Satisfiability problem (the SAT problem):** Given a set $X = \{x_1, \ldots, x_n\}$ of Boolean variables and a collection $C = \{c_1, \ldots, c_r\}$ of clauses over $X$, is there a satisfying truth assignment for $C$?

Suppose that $X = \{x_1, \ldots, x_n\}$ and $C = \{c_1, \ldots, c_r\}$ is an instance of the SAT problem. Let $X' = \{x'_1, \ldots, x'_n\}$, $A = \{a_1, \ldots, a_n\}$, $P = \{p_1, \ldots, p_n\}$, $D = \{d_1, \ldots, d_n\}$, $B = \{b_1, \ldots, b_n\}$, and $U = \{u_1, \ldots, u_n\}$. A corresponding instance $G(V, E)$ and $F$ of the RMIDS problem can be constructed as follows: $V = C \cup U \cup X \cup X' \cup A \cup P \cup D \cup B$, $E = \{(c_j, x) \mid c_j$ contains $x$, for all $c_j \in C$ and $x \in X \cup X'\} \cup \{(x_j, u_j), (x_j, p_j), (a_j, p_j), (x'_j, u_j), (x'_j, b_j), (d_j, b_j) \mid 1 \le j \le n\} \cup \{(y, v) \mid$ for all $y, v \in C \cup U\}$, and $F = C \cup U$. On the constructed graph $G$, the following lemmas can be proved.

**Lemma 1.** *Let $Q$ be a RID set of $G$, i.e., $Q \subseteq (X \cup X' \cup A \cup P \cup D \cup B)$ and $G^{(j)}$ be the subgraph induced by $\{x_j, u_j, x'_j, p_j, b_j, a_j, d_j\}$, $1 \le j \le n$. Only one of the following cases could be true.* (1) $\{x_j, b_j, a_j\} \subseteq Q$. (2) $\{x'_j, p_j, d_j\} \subseteq Q$. (3) $\{x_j, x'_j, a_j, d_j\} \subseteq Q$.

**Lemma 2.** *$G$ can be constructed in $O((n + r)^2)$-time and it is a chordal graph.*

*Proof*: It is easy to see that there are $7n + r$ vertices in $G$. So, $G$ can be constructed in $O((n + r)^2)$-time. Next, let $\Omega = \{v_1, \ldots, v_\alpha\}$, $\alpha \ge 4$, be a cycle in $G$. The construction rule of $G$ implies that $\Omega \cap (A \cup D \cup P \cup B) = \varnothing$. Since all vertices in $F = C \cup U$ form a clique, if $\Omega$ contains at least three vertices in $F$, then $\Omega$ must have a chord. Thus, we only need to consider: (1) $\Omega$ contains no vertex in $F$, (2) $\Omega$ contains only one vertex in $F$, and (3) $\Omega$ contains exactly two vertices in $F$. It is easy to verify that a chord must exist in each case. Therefore, $G$ is a chordal graph.

**Lemma 3.** *Suppose that $Q$ is a minimum RID set of $G$. Then, the SAT problem has a solution iff $\left|Q\right| = 3n$.*

*Proof*: If the SAT problem has a solution and let $H$ be the set of TRUE literals, then $\left|H\right| = n$. Put $R_j = \{a_j, b_j\}$ if $x_j \in H$. Otherwise, $R_j = \{p_j, d_j\}$, $1 \le j \le n$. By Lemma 1, $Q = H \cup R_1 \cup R_2 \cup \ldots \cup R_n$ is a minimum RID set of $G$ and $\left|Q\right| = 3n$.

Assume that $\left|Q\right| = 3n$. Let $H = Q \cap (X \cup X')$. By Lemma 1, $\left|H\right| = n$ and either $x_j$ or $x'_j$ is included in $H$, $1 \le j \le n$. Setting the literals corresponding to the vertices in $H$ to be TURE must satisfy the input Boolean formula.

**Lemma 4.** *The SAT problem can be reduced to the RMIDS problem on chordal graphs in $O((n + r)^2)$-time.*

**Theorem 1.** *The RMIDS problem is NP-Hard on chordal graphs.*

We now consider the RWMIDS problem on split graphs. A graph is called a *split graph* if its vertices can be divided into two disjoint sets $K$ and $I$ such that $K$ is a clique and $I$ is an independent set [9, 21]. Let SG$(V, E)$ be a split graph with $V = K \cup I = \{k_1, ..., k_s\} \cup \{i_1, ..., i_t\}$. For any RID set $D$, if $\left|K \cap F\right| < \left|K\right|$, then either $K \cap D = \{k_j\}$, for some $k_j$, or $K \cap D = \varnothing$. Define $\delta_0(\mathrm{SG}) = \min\{\delta(D) \mid K \cap D = \varnothing$ and $D$ is a RID set of SG$\}$ and $\delta_1(\mathrm{SG}) = \min\{\delta(D) \mid K \cap D = \{k_j\}$, for some $k_j$, and $D$ is a RID set of SG$\}$. It is clear that $\delta(\mathrm{SG}) = \min\{\delta_0(\mathrm{SG}), \delta_1(\mathrm{SG})\}$.

To compute $\delta_0(\mathrm{SG})$, for $H \subset I$, the vertices in $I - H$ are not dominated by $H$. If $H$ is a RID set and $K \cap H = \varnothing$, then $H$ must be $I$ itself, i.e., there has no solution if $I \cap F \neq \varnothing$. Therefore, the first task is to check whether $I \cap F \neq \varnothing$. If $I \cap F = \varnothing$, then the second task is to check whether every vertex in $K$ is adjacent to at least one vertex in $I$ and the time-complexity for computing $\delta_0(\mathrm{SG})$ is $O(n)$.

Next, we show how to compute $\delta_1(\mathrm{SG})$. It is easy to see that $\delta_1(\mathrm{SG}) = \infty$, if $K \subseteq F$. Let $Q = K - F \neq \varnothing$. For any vertex $k_j$ in $K$, define Neighbor-I$(k_j) = \{i_q \in I \mid (i_q, k_j) \in E\}$. For any optimal solution $V^*$ containing some $k_j$ in $Q$, $V^*$ must contain all vertices belonging to $I$ - Neighbor-I$(k_j)$, i.e., if $(I$ - Neighbor-I$(k_j)) \cap F \neq \varnothing$ in this situation, then $\delta_1(\mathrm{SG}) = \infty$. It implies that $\delta_1(\mathrm{SG})$ can be obtained in $O(m)$-time.

**Theorem 2.** *The RWMIDS problem can be solved in $O(m)$-time on split graphs.*

## 3. The RWMID Problem on Trees

This section designs an $O(n)$-time algorithms for the RWMIDS problem on trees. The strategy used is the dynamic programming strategy [1, 14]. Given a tree $T$, we choose any vertex $r$ to be the root and the tree is denoted as $T(r)$. Define $\delta(T(r)) = \min\{\delta(D) \mid D$ is a RID set of $T(r)\}$. For any non-root vertex $x$, the following notations are defined. (1) Denote $T(x)$ as the subtree rooted at $x$ and par($x$) is the parent of $x$. (2) $\delta_{00}(x)) = \min\{\delta(D) \mid D$ is a RID set of $T(x)$ such that par($x$) $\notin D$ and $x \notin D\}$. (3) $\delta_{01}(x)) = \min\{\delta(D) \mid D$ is a RID set of $T(x)$ such that par($x$) $\notin D$ and $x \in D\}$. (4) $\delta_{10}(x)) = \min\{\delta(D) \mid D$ is a RID set of $T(x)$ such that par($x$) $\in D$ and $x \notin D\}$. For any vertex $u$, the following notations are defined. (1) $\delta_0(T(u)) = \min\{\delta(D) \mid D$ is a RID set of $T(u)$ such that $u \notin D\}$. (2) $\delta_1(T(u)) = \min\{\delta(D) \mid D$ is a RID set of $T(x)$ such that $u \in D\}$. (3) $\delta(T(u)) = \min\{\delta(D) \mid D$ is a RID set of $T(u)\}$.

**Case 1.** $r \in (V - F)$, i.e., $r$ is not a forbidden vertex.

Above definitions derive the following formula.

$\delta(T(r)) = \min\{\delta_0(T(r)), \delta_1(T(r))\}$ <1>

Among all children of $r$, suppose that $X = \{x_1, \ldots, x_k\} \subseteq F$ and $Y = \{y_1, \ldots, y_q\} \subseteq (V - F)$. To compute $\delta_0(T(r))$, if $Y$ is not empty, then at least one vertex in $Y$ must be selected.

$$\delta_0(T(r)) = \begin{cases} \infty, Y = \varnothing \\ \sum_{1\le j\le k} \delta_{00}(T(x_j)) + \min_{1\le j\le q}\left\{\delta_{01}(T(y_j)) + \sum_{1\le s\ne j\le q}\left(\delta(T(y_s))\right)\right\}, Y \ne \varnothing \end{cases}$$

<2>

The next task is to compute $\delta_1(T(r))$. It implies that $D \cap (X \cup Y) = \varnothing$, for all feasible solution $D$.

$$\delta_1(T(r)) = W(r) + \sum_{1\le j\le k} \delta_{10}(T(x_j)) + \sum_{1\le j\le q} \delta_{10}(T(y_j)) \quad \text{<3>}$$

**Case 2.** $r \in F$, i.e., $r$ is a forbidden vertex

If $Y$ is not empty, then at least one vertex in $Y$ must be selected. This case is just equivalent to compute $\delta_0(T(r))$ as Case 1.

Now, consider any non-root vertex $u$. Also, among all children of $u$, suppose that $X = \{x_1, \ldots, x_k\} \subseteq F$ and $Y = \{y_1, \ldots, y_q\} \subseteq (V - F)$.

**Case 1.** $u \in (V - F)$, i.e., $u$ is not a forbidden vertex

$$\delta_{00}(T(u)) = \begin{cases} \infty, Y = \varnothing \\ \sum_{1\le j\le k} \delta_{00}(T(x_j)) + \min_{1\le j\le q}\left\{\delta_{01}(T(y_j)) + \sum_{1\le s\ne j\le q}\left(\delta(T(y_s))\right)\right\}, Y \ne \varnothing \end{cases}$$

<4>

$$\delta_{01}(T(u)) = W(u) + \sum_{1\le j\le k} \delta_{10}(T(x_j)) + \sum_{1\le j\le q} \delta_{10}(T(y_j)) \quad \text{<5>}$$

To compute $\delta_{10}(T(u))$ means that $u$ is already dominated by par($u$).

$$\delta_{10}(T(u)) = \sum_{1\le j\le k} \delta_{00}(T(x_j)) + \sum_{1\le j\le q} \delta(T(y_j)) \quad \text{<6>}$$

**Case 2.** $u \in V$, i.e., $u$ is a forbidden vertex

In this case, $\delta_{01}(T(u)) = \infty$. The other two values, $\delta_{00}(T(u))$ and $\delta_{10}(T(u))$ are the same as Case 1.

The boundary condition occurs when $T(u)$ consists of the vertex $u$. Two possibilities could occur.

**Case 1.** $u$ is the root

$$\delta_0(T(u)) = \infty \quad \text{<7>} \qquad \delta_1(T(u)) = \begin{cases} \infty, u \in F \\ W(u), u \in (V - F) \end{cases} \quad \text{<8>}$$

$\delta(T(u)) = \min\{\delta_0(T(u)), \delta_1(T(u))\}$ <9>

**Case 2.** $u$ is non-root

$\delta_0(T(u)) = \delta_{00}(T(u)) = \infty$ <10>

$$\delta_1(T(u)) = \delta_{01}(T(u)) = \begin{cases} \infty, u \in F \\ W(u), u \in (V - F) \end{cases} \quad \text{<11>}$$

$\delta_{10}(T(u)) = 0$ <12>

$\delta(T(u)) = \min\{\delta_0(T(u)), \delta_1(T(u))\}$ <13>

**Lemma 5.** Formulas <1> to <13> can be implemented in either $O(1)$-time or $O(k + q)$-time, except Formula <2> and Formula <4>.

**Lemma 6.** Formula <2> and Formula <4> can be implemented in $O(k + q)$-time.

*Proof*: $\sum_{1\le j\le k} \delta_{00}(T(x_j))$ can be computed in $O(k)$-time. Let $\delta^{(j)} = \delta_{01}(T(x_j)) + \sum_{1\le s\ne j\le q} \delta(T(y_s))$, $1 \le j \le q$. It is easy to verify that $\delta^{(j+1)} = \delta^{(j)} - \delta_{01}(T(x_j)) + \delta_{01}(T(x_{j+1})) - \delta(T(x_{j+1})) + \delta(T(x_j))$, $2 \le j \le q$. Therefore, $\delta^{(1)}, \ldots \delta^{(q)}$ can be computed in $O(q)$-time. This implies that Formula <2> and Formula <4> can be implemented in $O(k + q)$-time.

**Theorem 3.** *The RWMIDS problem con be solved in $O(n)$-time on trees.*

# 4. The RBIDS Problem

**Theorem 4.** *The RBIDS problem is linear-time equivalent to the BIDS problem on any class of graphs.*

*Proof*: Let $G(V, E, C)$ and $F$ be an instance of the RBIDS problem. The corresponding instance $G'(V', E', C')$ of the BIDS problem can be derived as: $V' = V$ and $E' = E$, $C'(v) = C(v)$, for all $v \in (V - F)$, and $C'(v) = \eta + 1$, where $\eta = \max_{v\in(V-F)}\{C(v)\}$, for all $v \in F$. Let $D$ be an optimal solution of the BIDS problem on $G'$. **Case 1.** $\beta(D) \le \eta$. It is easy to verify that $D \subseteq (V - F)$ and $D$ is also an optimal solution of the RBIDS problem on $G$. **Case 2.** $\beta(D) = \eta + 1$. By the rules of assigning $C'(v)$, for all $v$, it implies that all subsets $Q$ of $(V - F)$ cannot form an independent dominating set of $G'$, i.e., the RBIDS problem on $G$ has no solution. Since we do not change the class of graphs, the proof is completed.

## 5. Conclusions

The MIDS problem can be solved in linear-time on chordal graphs. This paper has shown that the RMIDS problem is NP-Hard on chordal graphs and the RWMIDS problem is $O(m)$-time solvable on split graphs. An $O(n)$-time algorithm for the RWMIDS problem on trees has also been proposed. But, we have proven that the RBIDS problem is linear-time equivalent to the BIDS problem on any class of graphs.

In the future, it is worthy to identify the classes of graphs such that the time-complexities of the RMIDS problem and the MIDS problem fall into different complexity classes. Other types of domination issues, such as perfect domination, total domination, and connected domination, with forbidden vertices is other research direction deserving us to study in detail.

## Acknowledgement

This research was supported by National Science Council, Taiwan, R.O.C., under contract number NSC-91-2213-E-128-002.

## References

[1] R. E. Bellman and S. E. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, N.J., 1962.

[2] C. Bo and B. Liu, Some Inequalities about Connected Domination Number, Discrete Mathematics 159 (1996) 241-245.

[3] G. J. Chang, Total Domination in Block Graphs, Operations Research Letters 8 (1989) 53-57.

[4] Peter DAMASCHKE, Haiko MÜLLER, and Dieter KRATSCH, Domination in Convex and Chordal Bipartite Graphs, Information Processing Letters 36 (1990) 231-236.

[5] M. Farber, Independent Domination in Chordal Graphs, Operation Research Letters 1 (1982) 134-138.

[6] M. Farber, Domination, Independent Domination and Duality in Strongly Chordal Graphs, Discrete Applied Mathematics 7 (1984) 115-130.

[7] M. R. Garey and David S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Bell Laboratories, Murray Hill, Freeman & Co., N. J., 1978.

[8] F. Gavril, The Intersection Graphs of Subtrees in Tree Are Exactly the Chordal Graphs, Journal of Combinatorial Theory Series B, 16 (1974), pp. 47-56.

[9] M. C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, New York, Academic Press, 1980.

[10] T. W. Haynes, S. T. Hedetniemi, and P. J. Salter, Fundamentals of Domination in Graphs, Marcel Dekker, Inc., New York, Basel, Hong Kong, 1998.

[11] T. W. Haynes, S. T. Hedetniemi, and P. J. Salter, Domination in Graphs: Advanced Topics, Marcel Dekker, Inc., New York, Basel, Hong Kong, 1998.

[12] R. W. Irving, On Approximating the Minimum Independent Dominating Set, Information Processing Letters 37 (1991) 197-200.

[13] Y. C. Liu and M. S. Chang, Polynomial Algorithms for Various Weighted Perfect Domination Problems on Some Classes of Graphs, Master Theses, National Chung Cheng University, 1993.

[14] G. L. Nemhauser, Introduction to Dynamic Programming, Wiley, New York, 1966.

[15] J. Pfaff, R. Laskar, and S. T. Hedetniemi, Linear Algorithm for Independent Domination and Total Domination in Series-Parallel Graphs, Tech. Rept. 441, Clemson Univ., Clemson, SC, 1984.

[16] D. J. Rose, R. E. Tarjan, and G. S. Lueker, Algorithmic Aspects of Vertex Elimination on Graphs, SIAM Journal on Computing, 5 (1976), pp. 266-283.

[17] C. C. Yen and R. C. T. Lee, The Weighted Perfect Domination Problem, Information Processing Letters 35 (1990) 295-299.

[18] C. C. Yen and R. C. T. Lee, Linear Time Algorithms to Solve the Weighted Perfect Domination Problem in Series-Parallel Graphs, European Journal of Operations Research 73 (1994) 19-26.

[19] W. C-K Yen, Bottleneck Domination and Bottleneck Independent Domination on Graphs, Journal of Information Science and Engineering 18 (2002) 311-331.

[20] W. C-K Yen, The Bottleneck Independent Domination on the Classes of Bipartite Graphs and Block Graphs, Information Sciences 157 (2003) 199-215.

[21] W. C. K. Yen, Bottleneck Independent Domination on Some Classes of Graphs, in Proc. of International Computer Symposium, Workshop on Algorithms and Theory of Computation, National Chung Cheng University, Chiayi, Taiwan, 1-8 (2000).