International Journal of Contemporary Mathematical Sciences
Vol. 13, 2018, no. 3, 133 - 140
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ijcms.2018.8413

Algorithms for Weighted Domination Number and Weighted Independent Domination Number of a Tree

Min-Jen Jou and Jenq-Jong Lin

Ling Tung University, Taichung 40852, Taiwan

Abstract

A set S of vertices is a dominating set of G if every vertex not in S is adjacent to at least one member of S. An independent dominating set I of G is a dominating set of G if no two vertices of I are adjacent. The domination problem is NP-complete for an arbitrary graph. Here we focus on weighted trees. A weighted tree (T, w) is a tree together with a positive weight function on the vertex set $w : V(T) \to R^+$. The weighted domination number $\gamma_w(T)$ of (T, w) is the minimum weight $w(D) = \sum_{v \in D} w(v)$ of a dominating set D of T. The weighted independent domination number $i_w(T)$ of (T, w) is the minimum weight $w(I) = \sum_{v \in I} w(v)$ of an independent dominating set I of T. In this paper, we provide the liner-time algorithms for finding the weighted domination number and weighted independent domination number of a weighted tree.

Mathematics Subject Classification: 05C69, 05C85

Keywords: (independent) dominating set, weighted tree, weighted (independent) domination number

1 Introduction

A set S of vertices is a *dominating set* of G if every vertex not in S is adjacent to at least one member of S. The set $\mathcal{D}(G)$ is the collection of all dominating sets of G. An *independent dominating set* I of G is a dominating set of G if no two vertices of I are adjacent. The set $\mathcal{I}(G)$ is the collection of all independent dominating sets of G. A weighted tree (T, w) is a tree together with a positive weight function on the vertex set $w : V(T) \to R^+$. The weighted domination number $\gamma_w(T)$ of (T, w) is the minimum weight $w(D) = \sum_{v \in D} w(v)$ of a dominating set D of T. The weighted independent domination number $i_w(T)$ of (T, w) is the minimum weight $w(I) = \sum_{v \in I} w(v)$ of an independent dominating set I of T.

One of the fastest growing areas within graph theory is the study of domination and related subset problems. In 1962, Claude Berge [2] wrote a book on graph theory, in which he introduced the domination number of a graph. Oystein Ore [11] introduced the dominating set and domination number in his book on graph theory. Domination in graphs is now well studied in graph theory and the literature on this subject has been surveyed and detailed in the two books by Teresa W. Haynes, Stephen Hedetniemi, and Peter Slater ([8],[9]). The independent domination number and the notation $i(G)$ were introduced by Cockayne and Hedetniemi in [3]. Recently, it was then extensively studied for various classes of graphs in the literature (see [6], [5],[7],[10],[12]).

The domination problem is NP-complete for an arbitrary graph [4]. The domination problem has been well studied in the area of algorithmic graph theory. It has many applications, such as communication networks, requirements and games. Here we focus on weighted trees. In this paper, we provide the liner-time algorithms for finding the weighted domination number and weighted independent domination number of a weighted tree.

2 Notations

All graphs considered in this paper are finite, loopless, and without multiple edges. For a graph G, $V(G)$ and $E(G)$ denote the *vertex set* and the *edge set* of G, respectively. The cardinality of $V(G)$ is called the *order* of G, denoted by $|G|$. The order-zero graph is called the *null graph* and dented by K_0. The (open) neighborhood $N_G(v)$ of a vertex v is the set of vertices adjacent to v in G, and the close neighborhood $N_G[v]$ is $N_G[v] = N_G(v) \cup \{v\}$. For any subset $A \subseteq V(G)$, denote $N_G(A) = \bigcup_{v \in A} N_G(v)$ and $N_G[A] = \bigcup_{v \in A} N_G[v]$. The *degree* of v is the cardinality of $N_G(v)$, denoted by $\deg_G(v)$. A vertex x is said to be a *leaf* if $\deg_G(x) = 1$. For two sets A and B, the difference A-B is the set of all elements of A that are not elements of B. For a subset $A \subseteq V(G)$, the *deletion of A from G* is the graph $G - A$ obtained by removing all vertices in A and all edges incident to these vertices. A *forest* is a graph with no cycles, and a *tree* is a connected forest. The *leaf-first labeling* $[1, 2, \ldots, n]$ of a tree T is a vertex labeling such that the vertex 1 is a leaf of T and i is a leaf of the subgraph $T - \{1, \ldots, i-1\}$ for $i = 2, \ldots, n-1$. For $i \geq 1$, the subgraph T_i is the maximal connected subgraph of T including the vertex i and some vertices

$j < i$. The set $\widetilde{N}(i)$ is the collection of the vertices adjacent to the vertex i in T_i. For other undefined notions, the reader is referred to [1] for graph theory.

3 The weighted domination number of a tree

In this section, we provide a liner-time algorithm for finding the weighted domination number of a weighted tree. Let (T, w) be a weighted tree and $[1, 2, \ldots, n]$ be a leaf-first labeling of (T, w). For $i \geq 1$, we define the following sets and numbers.

$$\mathcal{D}^1(T_i) = \{D : D \in \mathcal{D}(T_i), i \in D\}.$$

$$\mathcal{D}^0(T_i) = \{D : D \in \mathcal{D}(T_i), i \notin D\}.$$

$$\mathcal{D}^-(T_i) = \{D : D \subset V(T_i), V(T_i - \{i\}) \subseteq N_T[D]\}.$$

$$a_i = \min\{w(D) : D \in \mathcal{D}^1(T_i)\}.$$

$$b_i = \min\{w(D) : D \in \mathcal{D}^0(T_i)\}.$$

$$c_i = \min\{w(D) : D \in \mathcal{D}(T_i)\}.$$

$$d_i = \min\{w(D) : D \in \mathcal{D}^-(T_i)\}.$$

Note that $\gamma_w(K_0) = 0$ and $\gamma_w(T_i) = c_i$ for $i = 1, 2 \ldots, n$. The correctness of the Algorithm 1 is based on the following theorem.

Theorem 3.1. *Let (T, w) be a weighted tree and $[1, 2, \ldots, n]$ be a leaf-first labeling of T, where $w(i) = w_i$ for $i = 1, 2 \ldots, n$.*
(i) *If i is a leaf of T, then $a_i = w_i$, $b_i = 0$, $c_i = w_i$ and $d_i = 0$.*
(ii) *If i is not a leaf of T, then we have the following results.*

$$\begin{cases} a_i = w_i + \sum\limits_{j \in \widetilde{N}(i)} d_j. \\ b_i = \min\limits_{j \in \widetilde{N}(i)} \{a_j + \sum\limits_{k \in \widetilde{N}(i), k \neq j} c_k\}. \\ c_i = \min\{a_i, b_i\}. \\ d_i = \min\{c_i, \sum\limits_{j \in \widetilde{N}(i)} c_j\}. \end{cases}$$

Proof. We prove it by induction on $i \geq 1$. If $i = 1$, then $\mathcal{D}^1(T_1) = \{\{1\}\}$, $\mathcal{D}^0(T_1) = \emptyset$, $\mathcal{D}(T_1) = \{\{1\}\}$ and $\mathcal{D}^-(T_1) = \{\emptyset, \{1\}\}$. So $a_1 = w_1$, $b_1 = 0$, $c_1 = w_1$ and $d_1 = 0$. Assume that it's true for all $i' < i$, where $i \geq 2$. If i is a leaf of T, then $\mathcal{D}^1(T_i) = \{\{i\}\}$, $\mathcal{D}^0(T_i) = \emptyset$, $\mathcal{D}(T_i) = \{\{i\}\}$ and $\mathcal{D}^-(T_1) = \{\emptyset, \{i\}\}$. So $a_i = w_i$, $b_i = 0$, $c_i = w_i$ and $d_i = 0$. Now we assume that i is not a leaf of T, then $\widetilde{N}(i) \neq \emptyset$.

(1). If $D \in \mathcal{D}^1(T_i)$, then $i \in D$. Let $D_j = D \cap V(T_j)$ for every $j \in \widetilde{N}(i)$. For every $j \in \widetilde{N}(i)$, then $N_T[D_j] = V(T_j)$ or $N_T[D_j] = V(T_j) - \{j\}$. That is $D_j \in \mathcal{D}(T_j)$ or $D_j \in \mathcal{D}^-(T_j)$. Note that $d_i \leq c_i$ for each i. By induction, $w(D_j) \geq d_j$. Hence

$$\begin{aligned} a_i &= \min\{w(D) : D \in \mathcal{D}^1(T_i)\} \\ &= w_i + \sum_{j \in \widetilde{N}(i)} d_j. \end{aligned}$$

(2). If $D \in \mathcal{D}^0(T_i)$, then $i \notin D$ and $D \cap \widetilde{N}(i) \neq \emptyset$. If $j \in D$ for some $j \in \widetilde{N}(i)$, by induction, we have

$$w(D) \geq a_j + \sum_{k \in \widetilde{N}(i), k \neq j} \gamma_w(T_k) = a_j + \sum_{k \in \widetilde{N}(i), k \neq j} c_k.$$

Hence

$$\begin{aligned} b_i &= \min\{w(D) : D \in \mathcal{D}^0(T_i)\} \\ &= \min_{j \in \widetilde{N}(i)} \{a_j + \sum_{k \in \widetilde{N}(i), k \neq j} c_k\}. \end{aligned}$$

(3). If $D \in \mathcal{D}(T_i)$, then $D \in \mathcal{D}^1(T_i)$ or $D \in \mathcal{D}^0(T_i)$. By (1) and (2), $c_i = \min\{a_i, b_i\}$.

(4). If $D \in \mathcal{D}^-(T_i)$, then $D \in \mathcal{D}(T_i)$ or $N_T[D] = V(T_i) - \{i\}$. If $D \in \mathcal{D}(T_i)$, by (3), then $w(D) \geq c_i$. If $N_T[D] = V(T_i) - \{i\}$, then $D_j \in \mathcal{D}(T_j)$ for every $j \in \widetilde{N}(i)$. By induction, $w(D) \geq \sum_{j \in \widetilde{N}(i)} c_j$. Hence

$$\begin{aligned} d_i &= \min\{w(D) : D \in \mathcal{D}^-(T_i)\} \\ &= \min\{c_i, \sum_{j \in \widetilde{N}(i)} c_j\}. \end{aligned}$$

By (1),(2),(3) and (4), it's true for i. We complete the proof. □

Note that $T_n = T$. By Theorem 3.1, then $\gamma_w(T) = \min\{w(D) : D \in \mathcal{D}(T_i)\} = c_n$. Base on above theorem, we have the Algorithm 1 for the weighted domination number in a weighted tree.

4 The weighted independent domination number of a tree

In this section, we provide a liner-time algorithm for finding the weighted independent domination number of a weighted tree. Let (T, w) be a weighted

Algorithm 1 The weighted domination $\gamma_w(T)$ of a tree T

Input: A weighted tree (T, w) and $[1, \ldots, n]$ is a leaf-first labeling of T.
Output: The weighted domination number $\gamma_w(T)$.

for $i = 1$ to n **do**
  if i is a leaf **then**
    $a_i \leftarrow w_i$
    $b_i \leftarrow 0$
    $c_i \leftarrow w_i$
    $d_i \leftarrow 0$
  else
    $a_i \leftarrow w_i + \sum_{j \in \widetilde{N}(i)} d_j$
    $b_i \leftarrow \min_{j \in \widetilde{N}(i)} \{a_j + \sum_{k \in \widetilde{N}(i), k \neq j} c_j\}$
    $c_i \leftarrow \ min\{a_i, b_i\}$
    $d_i \leftarrow \min\{c_i, \sum_{j \in \widetilde{N}(i)} c_j\}$
  end if
end for
$\gamma_w(T) \leftarrow c_n$

tree and $[1, 2, \ldots, n]$ be a leaf-first labeling of (T, w). Note that $i_w(K_0) = 0$. For $i \geq 1$, we define the following sets and numbers.

$$\mathcal{I}^1(T_i) = \{I : I \in \mathcal{I}(T_i), i \in I\}$$

$$\mathcal{I}^0(T_i) = \{I : I \in \mathcal{I}(T_i), i \notin I\}$$

$$\mathcal{I}^-(T_i) = \{I : I \in \mathcal{I}(T_i - \{i\})\}$$

$$\alpha_i = \min\{w(I) : I \in \mathcal{I}^1(T_i)\}$$

$$\beta_i = \min\{w(I) : I \in \mathcal{I}^0(T_i)\}$$

$$\delta_i = \min\{w(I) : I \in \mathcal{I}(T_i)\}$$

$$\lambda_i = \min\{w(I : I \in \mathcal{I}^-(T_i)\}$$

Note that $i_w(K_0) = 0$ and $i_w(T_i) = \delta_i$ for $i = 1, 2 \ldots, n$.The correctness of the Algorithm 2 is based on the following theorem.

Theorem 4.1. *Let (T, w) be a weighted tree and $[1, 2, \ldots, n]$ be a leaf-first labeling of T, where $w(i) = w_i$ for $i = 1, \ldots, n$.*
(i) *If i is a leaf of T, then $\alpha_i = w_i$, $\beta_i = 0$, $\delta_i = w_i$ and $\lambda_i = 0$.*

(ii) *If i is not a leaf of T, then we have the following results.*

$$\begin{cases} \alpha_i = w_i + \sum_{j\in\widetilde{N}(i)} \lambda_j. \\ \beta_i = \min_{j\in\widetilde{N}(i)} \{\alpha_j + \sum_{k\in\widetilde{N}(i),k\neq j} \delta_k\}. \\ \delta_i = \min\{\alpha_i, \beta_i\}. \\ \lambda_i = \sum_{j\in\widetilde{N}(i)} \delta_j. \end{cases}$$

Proof. We prove it by induction on $i \geq 1$. If $i = 1$, then $\mathcal{I}^1(T_1) = \{\{1\}\}$, $\mathcal{I}^0(T_1) = \emptyset$, $\mathcal{I}(T_1) = \{\{1\}\}$ and $\mathcal{I}^-(T_1) = \{\emptyset\}$. So $\alpha_1 = w_1$, $\beta_1 = 0$, $\delta_1 = w_1$ and $\lambda_1 = 0$. Assume that it's true for all $i' < i$, where $i \geq 2$. If i is a leaf of T, then $\mathcal{I}^1(T_i) = \{\{i\}\}$, $\mathcal{I}^0(T_i) = \emptyset$, $\mathcal{I}(T_i) = \{\{i\}\}$ and $\mathcal{I}^-(T_1) = \{\emptyset\}$. So $\alpha_i = w_i$, $\beta_i = 0$, $\delta_i = w_i$ and $\lambda_i = 0$. Now we assume that i is not a leaf of T, then $\widetilde{N}(i) \neq \emptyset$.

(1). If $I \in \mathcal{I}^1(T_i)$, then $i \in I$ and $\widetilde{N}(i) \cap I = \emptyset$. Let $I_j = I \cap V(T_j)$ for every $j \in \widetilde{N}(i)$. For every $j \in \widetilde{N}(i)$, note that $j \notin I_j$, then $I_j \in \mathcal{I}^-(T_j)$. By induction, $w(I_j) \geq \lambda_j$. Hence

$$\begin{aligned} \alpha_i &= \min\{w(I) : I \in \mathcal{I}^1(T_i)\} \\ &= w_i + \sum_{j\in\widetilde{N}(i)} \lambda_j. \end{aligned}$$

(2). If $I \in \mathcal{I}^0(T_i)$, then $i \notin I$ and $I \cap \widetilde{N}(i) \neq \emptyset$. If $j \in I$, by induction, we have

$$w(I) \geq \alpha_j + \sum_{k\in\widetilde{N}(i),k\neq j} i_w(T_k) = \alpha_j + \sum_{k\in\widetilde{N}(i),k\neq j} \delta_k.$$

Hence

$$\begin{aligned} \beta_i &= \min\{w(I) : D \in \mathcal{I}^0(T_i)\} \\ &= \min_{j\in\widetilde{N}(i)} \{\alpha_j + \sum_{k\in\widetilde{N}(i),k\neq j} \delta_k\}. \end{aligned}$$

(3). If $I \in \mathcal{I}(T_i)$, then $I \in \mathcal{I}^1(T_i)$ or $I \in \mathcal{I}^0(T_i)$. By (1) and (2), $\delta_i = \min\{\alpha_i, \beta_i\}$.

(4). If $I \in \mathcal{I}^-(T_i)$ and $I_j = I \cap V(T_j)$, where $j \in \widetilde{N}(i)$, then $I_j \in \mathcal{I}(T_j)$. By induction, $w(I) \geq \sum_{j\in\widetilde{N}(i)} \delta_j$. Hence

$$\begin{aligned} \lambda_i &= \min\{w(I) : I \in \mathcal{I}^-(T_i)\} \\ &= \sum_{j\in\widetilde{N}(i)} \delta_j. \end{aligned}$$

By (1),(2),(3) and (4), it's true for i. We complete the proof. $\square$

Base on the above theorem, we have the Algorithm 2 for the weighted independent domination number in a weighted tree.

Algorithm 2 The weighted independent domination $i_w(T)$ of a tree T

Input: A leaf-first ordering $[1, \ldots, n]$ and the weight function $w(T) = [w_1, \ldots, w_n]$.
Output: The weighted independent domination $i_w(T)$

```
for i = 1 to n do
    if i is a leaf then
        α_i ← w_i
        β_i ← 0
        δ_i ← w_i
        λ_i ← 0
    else
        α_i ← w_i + Σ_{j∈Ñ(i)} λ_j
        β_i ← min_{j∈Ñ(i)} {α_j + Σ_{k∈Ñ(i),k≠j} c_j}
        δ_i ← min{α_i, β_i}
        λ_i ← Σ_{j∈Ñ(i)} δ_j
    end if
end for
i_w(T) ← δ_n
```

References

[1] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, North-Holland New York, 1976.

[2] C. Berge, *Theory of Graphs and its Applications*, Methuen, London, 1962.

[3] E. J. Cockayne and S. T. Hedetniemi, Towards a theory of domination in graphs, *Networks*, **7** (1977), 247-261. https://doi.org/10.1002/net.3230070305

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[5] W. Goddard and M. Henning, Independent domination in graphs: A survey and recent results, *Discrete Math.*, **313** (2013), 839-854. https://doi.org/10.1016/j.disc.2012.11.031

[6] W. Goddard, M. Henning, J. Lyle and J. Southey, On the independent domination number of regular graphs, *Ann. Comb.*, **16** (2012), 719-732. https://doi.org/10.1007/s00026-012-0155-4

[7] J. Haviland, Independent domination in regular graphs, *Discrete Math.*, **143** (1995), 275-280. https://doi.org/10.1016/0012-365x(94)00022-b

[8] T. W. Haynes, S. Hedetniemi and P. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, Inc., New York, 1998.

[9] T. W. Haynes, S. Hedetniemi and P. Slater, *Domination in Graphs: Advanced Topics*, Marcel Dekker, Inc., New York, 1997.

[10] A. V. Kostochka, The independent domination number of a cubic 3-connected graph can be much larger than its domination number, *Graphs Combin.*, **9** (1993), 235-237. https://doi.org/10.1007/bf02988312

[11] O. Ore, *Theory of Graphs*, Amer. Math. Soc. Transl., Vol. 38, Colloquium Publications, 1962, 206-212. https://doi.org/10.1090/coll/038

[12] W. C. Shiu, X. Chen and W. H. Chan, Triangle-free graphs with large independent domination number, *Discrete Optim.*, **7** (2010), 86-92. https://doi.org/10.1016/j.disopt.2010.02.004

Received: May 14, 2018; Published: June 11 2018