BOTTLENECK INDEPENDENT DOMINATION ON SOME CLASSES OF GRAPHS[1]

William C. K. Yen
Dept. of Graphic Communications & Technology, Shih Hsin University,
Taipei, Taiwan, R.O.C.

ABSTRACT

Let $G(V, E, W)$ be a graph with vertex-set V and edge-set E, and each vertex v is associated with a positive cost $W(v)$. The bottleneck cost of any subset V' of V is defined to be $\max\{W(x) \mid x \in V'\}$. This paper resolves the Bottleneck Independent Dominating Set problem (the BIDS problem) which determines an independent dominating set of G with the minimum bottleneck cost.

The BIDS problem has been proven to be NP-hard on general bipartite graphs. This paper major discusses the problem on two hierarchies of graphs: chordal graphs and bipartite graphs. It first proves that the problem is NP-hard on chordal graphs, but linear-time solvable on weighted split graphs by the greedy approach. Second, it shows that the problem is still NP-hard on planar-bipartite graphs and a linear-time algorithm on weighted convex-bipartite graph is proposed. Finally, an $O(n)$ time algorithm of the BIDS problem on weighted cographs is designed. The later linear-time algorithms are designed by the dynamic programming strategy.

Keywords: bottleneck cost, independent dominating set, chordal graph, split graph, planar-bipartite graph, convex-bipartite graph, cographs

1. INTRODUCTION

Let $G(V, E, W)$ be a connected and undirected graph with vertex-set V and edge-set E, and each vertex v is associated with a positive cost $W(v)$. Assume that $|V| = n$ and $|E| = m$ hereafter. The concept of *dominating sets* of G arises naturally from various facilities locating problems in Operations Research and many practical applications such as communication systems and computer networks. A subset Q of V *dominates* V if there exists $u \in Q$ such that $(v, u) \in E$ for each $v \in (V - Q)$ and Q is called a *dominating set* of G [18, 19]. Many types of dominating sets on graphs have been proposed and studied for years, such as connected dominating sets [5, 20, 31], independent dominating sets [12, 13, 21, 23], total dominating sets [4, 6, 22, 29, 30], and perfect dominating sets [26, 32, 33]. Most of these previous studies emphasize to find a set D of certain type of dominating sets such that $\sum_{x \in D} W(x)$ is minimized. This paper considers another important cost measurement. For any $H \subseteq V$, the *bottleneck cost* is defined as $\max\{W(x) \mid x \in H\}$. This cost measurement is so important and practical when building services facilities or resources in real world. This paper concentrates on independent dominating sets (ID sets) of G. A subset H of V is *independent* if no two vertices in H are adjacent. An *independent dominating set* (*ID set*) of G is a subset of V which is both an independent set and a dominating set of G. The problem considered in this paper is defined precisely as follows:

The Bottleneck Independent Dominating Set problem (***The BIDS problem***): Given an undirected and connected graph $G(V, E, W)$, find an ID set $S \subseteq V$ such that $\max\{W(x) \mid x \in S\} \le \max\{W(x) \mid x \in H\}$, for all ID sets H.

Fig. 1 illustrates an input instance of the BIDS problem. The sets $\{a, e, h\}$ and $\{a, f\}$ are both ID sets of G. The set $\{a, f\}$ is an ID set with the minimum bottleneck cost, 2, which is equal to $\max\{W(a), W(f)\} = \max\{1, 2\} = 2$.

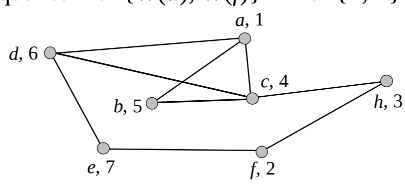

Fig. 1. An instance of the BIDS problem.

In [34], the BIDS problem has been proven to be NP-hard on general bipartite graphs. It is worthy to examine the borderline between polynomial time and NP-completeness on a given graph problem for various classes of graphs [11, 15]. This paper examines the hierarchies of chordal graphs and bipartite graphs. It first proves that the BIDS problem is NP-hard on chordal graphs, but linear-time solvable on weighted split graphs. This result is quite interesting since few problems have this property [26]. In [34], an $O(n\log^2 n)$ time algorithm was proposed for the BIDS problem on weighted permutation graphs. Therefore, the BIDS

[1]This research was supported by National Science Council, R.O.C., under contract number NSC-89-2213-E-159-001.

problem on weighted bipartite-permutation graphs can be solved at most in $O(n\log^2 n)$ time. This paper shows that the problem is still NP-hard on planar-bipartite graphs. Then, a linear-time algorithm on weighted convex-bipartite graphs is proposed. Finally, an $O(n)$ time algorithm of the BIDS problem on weighted cographs is designed.

2. SEPARATION IN COMPLEXITIES OF SPLIT GRAPHS FROM CHORDAL GRAPHS

2.1 NP-hardness on Chordal Graphs

To analyze the complexity about the BIDS problem, a decision problem corresponding to the BIDS problem and a variant of it are defined, respectively.

The Bottleneck Independent Dominating Set decision problem (***The BIDS decision problem***): Given a graph $G(V, E, W)$ and a real constant η, determine whether an independent dominating set $S \subseteq V$ exists such that $\max\{W(x) \mid x \in S\} \le \eta$.

The Constrained Independent Dominating Set decision problem (***The CIDS decision problem***): Given an undirected and connected graph $G(V, E)$ and a set of vertices $V' \subseteq V$, determine whether there exists an independent set $S \subseteq (V - V')$ which also dominates V'.

The following lemma has been established in [34].

Lemma 1: The BIDS decision problem is polynomially equivalent to the CIDS decision problem.

Now, to prove that the BIDS decision problem is NP-complete [15], it suffices to show that the CIDS decision problem is NP-complete. The technique used here is similar to that used in [25]. First, a known NP-complete problem is introduced [15].

The Monotone Three Satisfiability problem (***The M3SAT problem***): Given a set of Boolean clauses with the conjunctive normal form in which each clause can contain either only positive literals, say x_i′s, or only negative literals, say $\overline{x_i}$′s, and each literal contains exactly three literals, determine whether the given Boolean formula is satisfiable or not.

For any graph $G(V, E)$, an edge is called a *chord* of a cycle if it connects two nonconsecutive vertices of the cycle. G is called a *chordal graph* [16] if each cycle with length greater than three has a chord.

Lemma 2: The CIDS decision problem is NP-complete on chordal graphs.

Proof. It is clearly observable that the CIDS decision problem belongs to the NP class. We now show that the M3SAT problem can be polynomially reduced to the CIDS decision problem on chordal graphs. Suppose that there is an instance of the M3SAT problem with h variables $x_1, ..., x_h$ and r clauses $c_1, ..., c_r$. Assume that P represents the set of clauses that contain only positive literals and N represents the set of clauses that contain only negative literals, respectively. Meanwhile, in the case that will not cause any ambiguity, the literal-set of any clause c_i, $1 \le i \le r$, will also be denoted by c_i hereafter. Now, a graph $G(V, E)$ is constructed as follows: $V = \{c_1, ..., c_r\} \cup \{x_1, ..., x_h\} \cup \{a_1, ..., a_h, b_1, ..., b_h, d_1, ..., d_h, q_1, ..., q_h\}$; $E = \{(c_s, x_i) \mid c_s \text{ contains } x_i\} \cup \{(c_t, \overline{x_i}) \mid c_t \text{ contains } \overline{x_i}\} \cup \{(c_s, c_t) \mid$ for all clauses c_s and $c_t\} \cup E'$, where E' is obtained by the following loop.

E' = empty set; /* initialization */
for i = 1 to h do
$E' = E' \cup \{(x_i, a_i), (x_i, b_i), (a_i, b_i),$
$(a_i, d_i), (a_i, q_i), (a_i, \overline{x_i}), (\overline{x_i}, q_i),$
$(b_i, d_i), (d_i, q_i)\}$;
$E' = E' \cup \{(b_i, q_i)\}$;
endfor
for each pair $c_s \in P$ and $c_t \in N$ do
let $c_t = \{\overline{x_{t_1}}, \overline{x_{t_2}}, \overline{x_{t_3}}\}$;
for j = 1 to 3 do
if $x_{t_j} \in c_t$ $E' = E' \cup \{(a_i, c_t)\}$;
endfor
endfor

Fig. 2 and Fig. 3 depict the edges added by each iteration of the first loop and the second loop, respectively.

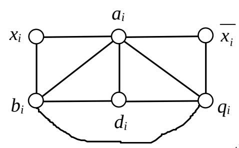

Fig. 2. The edges added by the i^{th} iteration.

The time-complexity of the above construction procedure can be easily proved to be $O(h + |P| * |N|)$. To show that the graph G is a chordal graph, let $X = \{x_1, ..., x_h\}$ and $\overline{X} = \{\overline{x_1}, ..., \overline{x_h}\}$. Suppose that a cycle Ω with length greater than three exists. Assume that the cycle is $v_1 — v_2 — ... — v_p — v_1$, $p \ge 4$, i.e., its vertex-set is $\{v_1, v_2, ..., v_p\}$. For the simplicity on explanation, the cycle Ω will be represented by $v_1 — v_2 — ... — v_p — v_1$ and $\{v_1, v_2, ..., v_p\}$ alternatively hereafter. Based on the construction procedure of G, it is easily verifiable that one of the following cases could occur.

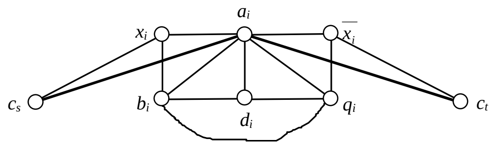

Fig. 3. Two edges (c_s, a_i) and (a_t, c_t) are added if c_s contains x_i and c_t contains $\overline{x_i}$.

Case 1. $\Omega \subseteq \{c_j\} \cup \{x_i, a_i, \overline{x_i}, b_i, d_i\ q_i\}$, for some c_j: In this case, Ω must have the form either c_j — a_i — ... — b_i — x_i — c_j or c_j — a_i — ... — q_i — $\overline{x_i}$ — c_j. The edge (x_i, a_i) and (a_i, $\overline{x_i}$) is a chord of the two cases, respectively.
Case 2. $\Omega \subseteq V - (P \cup N)$: In this case, Ω must be a cycle within a subgraph induced by $\{x_i, a_i, \overline{x_i}, b_i, d_i, q_i\}$, for some i. Checking that a chord exists in Ω is simple.
Case 3. $\Omega \subseteq P \cup X$: Since all vertices in X are independent, Ω must contain at least one subpath as c_s — x_i — c_t such that $c_s, c_t \in P$. This case is depicted in Fig. 4. In this situation $c_s \cap c_t = \{x_i\} \neq \emptyset$. From the construction rule of G, we can make sure that (c_s, c_t) is a chord of this cycle.
Case 4. $\Omega \subseteq \overline{X} \cup N$: This case is symmetrical to Case 3.
Case 5. $\Omega - (P \cup N) \neq \emptyset$: In this case, three possibilities must be considered.
Case 5.1. Ω contains a subpath c_s — x_i — c_t, where $c_s, c_t \in P$: As the discussions in Case 3, $c_s \cap c_t = \{x_i\} \neq \emptyset$, ($c_s$, c_t) is a chord of this cycle.
Case 5.2. Ω contains a subpath c_s — $\overline{x_i}$ — c_t, $c_s, c_t \in N$: It is symmetrical to Case 5.1.
Case 5.3. All subpaths in Ω have the form neither c_s — x_i — c_t, where $c_s, c_t \in P$ nor c_s — x_i — c_t, where $c_s, c_t \in N$: Since $\Omega - (P \cup N) \neq \emptyset$, $\Omega \cap P$ could contain exactly one vertex, say c_s, and $\Omega \cap N$ also contain exactly one vertex, say c_t. All other vertices in Ω must belong to $V - (P \cup N)$. It is easy to see that (c_s, c_t) is a chord of Ω.

Let $V' = \{c_1, ..., c_r, a_1, ..., a_h, d_1, ..., d_h\}$. Then, $(V - V') = \{x_1, ..., x_h, \overline{x_1}, ..., \overline{x_h}, b_1, ..., b_h, q_1, ..., q_h\}$. The remaining task is to show that there exists an independent set of $(V - V')$ which dominates V' in G iff the given Boolean formula $c_1 \bullet ... \bullet c_r$ is satisfiable.

1. Assume that there is an assignment which satisfies the Boolean formula. Let it be $x_{z_1} = ... = x_{z_\alpha}$ = True and $x_{w_1} = ... = x_{w_\varepsilon}$ = False, where $\alpha + \varepsilon = h$. Then, $z_i \neq w_j$, for any i and j. Put $S = \{ x_{z_1}, q_{z_1} ..., x_{z_\alpha}, q_{z\alpha}, \overline{x_{w_1}}, b_{w_1} ..., \overline{x_{w_\varepsilon}}, b_{w_\varepsilon} \} \subseteq (V - V')$. Verifying that S is an independent set of $(V - V')$ and S dominates all of the vertices in V' is an easy task.

2. Assume that $S \subseteq (V - V')$ is an independent set and also dominates V'. First, we claim that for each pair $\{x_i, \overline{x_i}\}$, either $x_i \in S$ or $\overline{x_i} \in S$, i.e., the case that both x_i and $\overline{x_i}$ belong to S can not occur. If both x_i and $\overline{x_i}$ belong to S, then based on the construction rule of G, $b_i \notin S$ and $q_i \notin S$ and no vertex in S can dominate d_i. A contradiction occurs. Now, let $Q = \{ x_{z_1}, ..., x_{z_\alpha}, \overline{x_{w_1}}, ..., \overline{x_{w_\varepsilon}} \}$ be the set of vertices corresponding to literals contained in S. Assign the literals corresponding to the vertices in Q to be True, i.e., assign $x_{z_1} = ... = x_{z_\alpha}$ = True and $x_{w_1} = ... = x_{w_\varepsilon}$ = False. Ascertaining that this assignment satisfies the given Boolean formula is simple.

From the discussions so far, there exists an independent set of $(V - V')$ dominating V' in the chordal graph G if and only if the given Boolean formula is satisfiable. Therefore, the CIDS decision problem is NP-complete on chordal graphs. ■

Theorem 1: The BIDS problem is NP-hard on chordal graphs.

2.2 An $O(m)$ Time Algorithm on Weighted Split Graphs

A graph $G(V, E)$ is called a *split graph* [14] if V can be separated into two disjoint sets K and I such that K forms a clique and I is an independent set. Let $SG(V, E)$ denote a split graph with the vertex-set $V = K \cup I = \{k_1, ..., k_s\} \cup \{i_1, ..., i_t\}$. For a vertex set H of SG, let $\beta(H)$ represent the bottleneck cost of H, that is, $\beta(H)$ can be expressed as $\max_{v\in H}\{W(v)\}$. For the sake of clear presentation, denote $\delta(SG)$ to be the value of the bottleneck cost of an optimal solution of the BIDS problem on SG, i.e., $\delta(SG) = \min\{\beta(H) \mid H \text{ is an ID set of SG}\}$.

Since K is a clique, for any ID set of SG, at most one vertex in K can be included. Let V^0 be any ID set of SG. Then, either $K \cap V^0 = \{k_j\}$, for some k_j, or $K \cap V^0 = \emptyset$. The following two new problems are introduced and Formula (2.1) can be easily obtained.

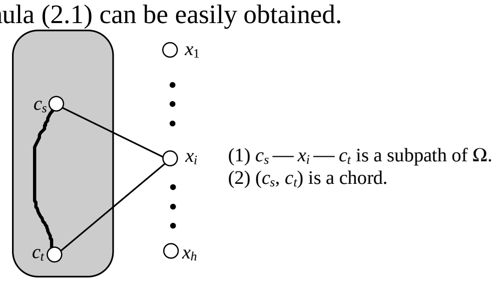

Fig. 4. The cycle $\Omega \subseteq P \cup X$.

(P0) Compute $\delta_K(\mathrm{SG}) = \min\{\beta(H) \mid K \cap H = \{k_j\}$, for some k_j, and H is an ID set$\}$.
(P1) Compute $\delta_{K'}(\mathrm{SG}) = \min\{\beta(H) \mid K \cap H = \varnothing$ and H is an ID set$\}$.

$$\delta(\mathrm{SG}) = \min\{\delta_K(\mathrm{SG}), \delta_{K'}(\mathrm{SG})\} \text{ -- (2.1)}$$

Let us first consider the problem (P0). For any vertex k_j in K, define Neighbor_I(k_j) = $\{i_q \in I \mid (i_q, k_j) \in E\}$. The set Neighbor_K($i_q$) can be defined similarly, for any vertex $i_q \in I$. The following property can be easily obtained.

Property 1: For any optimal solution V^* of the problem (P0), it must contain all vertices belonging to I – Neighbor_I(k_j), when $k_j \in V^*$.

Property 1 indicates that an optimal solution of the problem (P0) can be found by examining each vertex k_j in K, all vertices in I, as well as each edge once. Verifying that the time-complexity needed is $O(m)$ is a simple task.

Lemma 3: The problem (P0) can be solved in $O(m)$ time.

Now, turn to the problem (P1), i.e., find a subset V^* of I with the minimum bottleneck cost. If H is a proper subset of I, then the vertices in I - H are not dominated by H. Thus, if H is an ID set of SG and $K \cap H = \varnothing$, then H must be equal to I. Therefore, in order to solve the problem (P1), the only task is to check whether every vertex in K is adjacent to at least one vertex in I and the time-complexity is only $O(n)$. Now, the following lemma can be easily derived.

Lemma 4: The problem (P1) can be solved in $O(n)$ time.

Theorem 2: The BIDS problem can be solve in $O(m)$ time on weighted split graphs.

3. COMPLEXITIES ON PLANAR-BIPARTITE AND CONVEX-BIPARTITE GRAPHS

3.1 NP-hardness on Planar-Bipartite Graphs

As stated in Section 2.1, we also show that the CIDS decision problem is NP-complete here. The NP-Complete problem for reduction is also the M3SAT problem.

A graph $G(V, E)$ is called a *bipartite graph* if V can be partitioned into two disjoint sets X and Y such that both X and Y are independent sets. A bipartite graph will be denoted by $G(X \cup Y, E)$ herein. Meanwhile, any graph G is said to be *planar* if we can draw it into the plane such that all edges intersect only at end vertices [17]. A planar-bipartite graph is a bipartite graph which is also planar. The following theorem states the necessary and sufficient conditions for testing the planarity of graphs [17].

Theorem 3 (Kuratowski′s Theorem): A graph G is planar if and only if G contains no subgraph homeomorphic with K_5 or $K_{3,3}$.

Lemma 5: The CIDS decision problem is NP-complete on planar-bipartite graphs.
Proof. We now show that the M3SAT problem can be polynomially reduced to the CIDS decision problem on planar-bipartite graphs. Suppose that there is an instance of the M3SAT problem with h variables $x_1, \ldots, x_h$ and r clauses $c_1, \ldots, c_r$. In Boolean algebra, $c \bullet c = c$, for any clause c, so we can assume that $c_i \neq c_j$ for all $i \neq j$, i.e., c_i and c_j can not contain the same three literals. Let $P = \{c_{i_1}, \ldots, c_{i_a}\}$ be the set of clauses that contain only positive literals and $N = \{c_{j_1}, \ldots, c_{j_b}\}$ be the set of clauses that contain only negative literals. For the reason of clear presentation hereafter, denote $X_1 = \{x_1, \ldots, x_h\}$ and $X_2 = \{\overline{x_1}, \ldots, \overline{x_h}\}$. First, a bipartite graph $G(X \cup Y, E)$ is constructed as follows: $X = X_2 \cup P$; $Y = X_1 \cup N$; $V = X \cup Y$; $E = \{(x_i, c_s) \mid c_s$ contains $x_i.\} \cup \{(\overline{x_i}, c_t) \mid c_t$ contains $\overline{x_i}.\} \cup \{(x_i, \overline{x_i}) \mid 1 \leq i \leq h\}$. It is easy to see that no subgraph in G is homeomorphic with $K_{3,3}$ or K_5. Based on **Kuratowski′s Theorem**, the constructed graph is a planar-bipartite graph.

It is easy to ascertain that the time-complexity of the above transformation procedure is polynomial. Now, let $V' = \{c_1, \ldots, c_r\}$. Then, $(V - V')$ is the set $\{x_1, \ldots, x_h, \overline{x_1}, \ldots, \overline{x_h}\}$. The task left is to show that there exists an independent set of $(V - V')$ which dominates V' if and only if the given Boolean formula $c_1 \bullet \ldots \bullet c_r$ is satisfiable.
1. Assume that there is an assignment which satisfies the Boolean formula. Let it be $x_{p_1} = \ldots = x_{p_k}$ = True and $x_{q_1} = \ldots = x_{q_s}$ = False, where $k + s = h$. Then, $p_i \neq q_j$, for any i and j. Put $S = \{x_{p_1}, \ldots, x_{p_k}, \overline{x_{q_1}}, \ldots, \overline{x_{q_s}}\} \subseteq (V - V')$. Verifying that S is an independent set of $(V - V')$ and S dominates all of the vertices $c_1, \ldots, c_r$ is an easy task.
2. Assume that $S = \{x_{p_1}, \ldots, x_{p_k}, \overline{x_{q_1}}, \ldots, \overline{x_{q_s}}\} \subseteq (V - V')$ is an independent set and also dominates V'. A true assignment can be easily derived by assigning the literals corresponding to the vertices in S to be True.

From the discussions so far, an independent set of $(V - V')$ dominating V' exists in the planar-bipartite graph iff the given Boolean formula is satisfiable. Therefore, the CIDS decision problem is NP-complete on planar-bipartite graphs. ■

Theorem 4: The BIDS problem is NP-hard on planar-bipartite graphs.

3.2 A Linear-Time Algorithm on Weighted Convex-Bipartite Graphs

This section proposes a linear-time algorithm for the BIDS problem on weighted convex-bipartite graphs by the dynamic programming strategy [3, 28]. This technique has been applied to solve some other dominating set problems [27, 32].

A bipartite graph $G(X \cup Y, E)$ is called *convex* if the vertices in Y can be arranged to a linear ordering $(Y, <)$ such that $N(x)$ consists of consecutive vertices of Y, for each $x \in X$ [11]. Given a convex-bipartite graph $G(X \cup Y, E)$, assume that $X = \{x_1, ..., x_\varepsilon\}$ and $Y = \{y_1, ..., y_\eta\}$ hereafter. Some definitions are made in the following.

Definition 1: For any $x \in X$, let $N(x) = \{y_j, ..., y_{j+\alpha}\}$. Define $L(x) = j$ and $U(x) = j + \alpha$.

Definition 2: For any $x_1, x_2 \in X$, the two new relations on X, $<^r$ and $=^r$, are defined as follows, respectively. (1) $x_1 <^r x_2$ iff $L(x_1) < L(x_2)$ or ($L(x_1) = L(x_2)$ and $U(x_1) < U(x_2)$); (2) $x_1 =^r x_2$ iff $L(x_1) = L(x_2)$ and $U(x_1) = U(x_2)$.

Definition 3: Let $R^X \subseteq X$ and $R^Y \subseteq Y$. If the graph obtained from any convex-bipartite graph by removing the vertices belonging to $R^X \cup R^Y$ and all edges incident with them is still a convex-bipartite graph, then reindex the vertices in $X - R^X$ and $Y - R^Y$, respectively, and the graph obtained is denoted by $\mathrm{CB}_{R^X}^{R^Y}$. Based on this definition, the original convex-bipartite graph is just $\mathrm{CB}_{\varnothing}^{\varnothing}$ and simply denoted by CB.

Without a loss of generality, we will assume that all vertices in X are sorted into non-decreasing order under the relations defined in Definition 2 in the rest of this paper. In another, R^X and R^Y denote any two subsets of X any Y, respectively, satisfying the property stated in Definition 3 hereafter. For any R^X and R^Y, let $\delta(\mathrm{CB}_{R^X}^{R^Y})$ be the value of the minimum bottleneck cost of an optimal solution of the BIDS problem on the subgrarph $\mathrm{CB}_{R^X}^{R^Y}$, i.e., $\delta(\mathrm{CB}_{R^X}^{R^Y}) = \min\{\beta(D) \mid D$ is an ID set of $\mathrm{CB}_{R^X}^{R^Y}\}$. The BIDS problem is to compute $\delta(\mathrm{CB})$.

For each x_i, any optimal solution of the BIDS problem either includes x_i or not. This leads us to introduce the following two new related problems (P2) and (P3) and Formula (4.1) holds directly.

(P2) Compute $\delta_{\overline{x_i}}(\mathrm{CB}) = \min\{\beta(D) \mid x_i \notin D$ and D is an ID set of CB$\}$.
(P3) Compute $\delta_{x_i}(\mathrm{CB}) = \min\{\beta(D) \mid x_i \in D$ and D is an ID set of CB$\}$.

$$\delta(\mathrm{CB}) = \min\{\delta_{\overline{x_i}}(\mathrm{CB}), \delta_{x_i}(\mathrm{CB})\} \quad (4.1)$$

First, consider the boundary case in which $X = \{x_1\}$ and $Y = \{y_1, ..., y_\eta\}$. Since the input graph is assumed to be connected, x_1 is adjacent to all vertices $y_1, ..., y_\eta$ in this situation. Therefore, $\{x_1\}$ and $\{y_1, ..., y_\eta\}$ are the two only ID sets of CB. It is easy to verify that $\delta_{\overline{x_1}}(\mathrm{CB}) = \sum_{j=1}^{\eta} W(y_j)$ and $\delta_{x_1}(\mathrm{CB}) = W(x_1)$. $\delta(\mathrm{CB})$ is now equal to $\min\{W(x_1), \sum_{j=1}^{\eta} W(y_j)\}$ in this boundary condition.

Consider the cases in which $X = \{x_1, ..., x_\varepsilon\}$ and $Y = \{y_1, ..., y_\eta\}$, $\varepsilon, \eta \geq 2$. A useful property about $N(\{y_q, ..., y_{q+\beta}\})$ for all q and β can be easily verified as follows:

Property 2: If $N(\{y_q, ..., y_{q+\beta}\}) = \{x_{\pi_1}, ..., x_{\pi_t}\}$, then $x_{\pi_1} \leq^r ... \leq^r x_{\pi_t}$.

Consider the problem (P2) in which no feasible solution includes x_1. Assume that $N(x_1) = \{y_j, ..., y_{j+\alpha}\}$. From the definition of ID sets, an optimal solution of the problem (P2) must contain at least one vertex belonging to $\{y_j, ..., y_{j+\alpha}\}$. Suppose that some y_i, $j \leq i \leq \alpha$, must be included. In this situation, all vertices in $N(y_i)$ can not be included. The following lemma can be easily established.

Lemma 6: $\mathrm{CB}_{R_i^X}^{R_i^Y}$ are still convex-bipartite graphs for all $R_i^Y = \{y_i\}$ and $R_i^X = N(y_i)$.

Now, the correctness of the following formula can be easily ascertained.

$$\delta_{\overline{x_1}}(\mathrm{CB}) = \min_{j \leq i \leq \alpha}\{\delta(\mathrm{CB}_{R_i^X}^{R_i^Y})\} \quad (4.2)$$

Formula (4.2) indicates that the cases where y_i, $j \le i \le \alpha$, must be included are dealt with, respectively.

Next, consider the problem (P3) in which x_1 must be included in any optimal solution. In this case, all vertices in $N(x_1)$ can not be included in any optimal solution. Also, each vertex x_h in which $U(x_h) \le j + \alpha$ must be included because $N(x_h) \subseteq N(x_1)$. The following lemma can be directly derived from Property 2.

Lemma 7: $\mathrm{CB}_{R^X}^{R^Y}$ is still a convex-bipartite graph, where $R^X = \{x_h \mid U(x_h) \le j + \alpha\}$ and $R^Y = N(x_1)$.

The following formula can then be obtained.

$$\delta_{x_1}(\mathrm{CB}) = \max\{\max\{W(x) \mid x \in R^X\}, \delta(\mathrm{CB}_{R^X}^{R^Y})\} \quad (4.3)$$

Let T(P2, CB) and T(P3, CB) represent the time-complexities for solving the problems (P2) and (P3) on CB with $X = \{x_1, ..., x_\varepsilon\}$ and $Y = \{y_1, ..., y_\eta\}$, respectively. From the discussions so far, the following formula can be derived.

$T(\mathrm{BIDS}, \mathrm{CB}) = T(\mathrm{P2}, \mathrm{CB}) + T(\mathrm{P3}, \mathrm{CB})$ (4.4)

$T(\mathrm{P2}, \mathrm{CB}) = \sum_{i=j}^{\alpha} T(\mathrm{BIDS}, \mathrm{CB}_{R_i^X}^{R_i^Y})$, where $R_i^Y = \{y_i\}$ and $R_i^X = N(y_i)$. (4.5)

$T(\mathrm{P3}) = O(|R^X|) + T(\mathrm{BIDS}, \mathrm{CB}_{R^X}^{R^Y})$, where $R^X = \{x_h \mid U(x_h) \le j + \alpha\}$ and $R^Y = N(x_1)$. (4.6)

The boundary conditions yields the following formula, where CB^b denotes the convex-bipartite graph with the vertex-set $X \cup Y$ in which $X = \{x_1\}$ and $Y = \{y_1, ..., y_\eta\}$.

$T(\mathrm{P2}, \mathrm{CB}^b) = T(\mathrm{P3}, \mathrm{CB}^b) = T(\mathrm{BIDS}, \mathrm{CB}^b) = O(\eta + 1)$ (4.7)

It is easy to check that each edge is examined in constant time. Now, from Formula (4.1) to (4.7), it is simple to derive that $T(\mathrm{BIDS}, \mathrm{CB}) = O(m)$.

Theorem 5: The BIDS problem can be solved in $O(m)$ time on weighted convex-bipartite graphs.

4. AN $O(n)$ TIME ALGORITHM ON WEIGHTED COGRAPHS

The final class of graphs considered is the class of *cographs* [1, 2, 7, 37], which arises in a wide spectrum of applications. A cograph is defined recursively in the following way [7, 37]: (1) A single vertex is a cograph. (2) If $G_1, G_2, ..., G_k$ are cographs, then so is their union $G_1 \cup G_2 ... \cup G_k$. (3) If G is a cograph, then so is its complement $\overline{G}$. A cograph has a tree representation called *cotree* [7, 10]. The leaves of a cotree represent the vertices of its corresponding cograph, and its internal nodes are labeled with either 0 or 1. The root is labeled with 1 if the cograph is connected, and 0 otherwise. Two vertices x and y in a cograph are adjacent iff the lowest common ancestor of their corresponding nodes in the cotree is a 1-node. Since a general tree can be easily interpreted as a binary tree [24], only binary cotrees are considered herein. An $O(m + n)$ time algorithm has been developed to recognize a cograph and to construct its cotree representation [9, 10]. In the rest of the paper, given a cograph G and its corresponding cotree T, let r be any internal node of T. The subcotree rooted at r is denoted to be $T(r)$ and the cograph induced by the leaves of $T(r)$ is denoted to be $G(r)$.

Many efficient algorithms solve many problems on cographs, such as isomorphism, coloring, clique-detection, minimum weight dominating set, maximum matching, searchlight guarding [7, 8, 9, 35, 37]. Indeed, the class of cographs is a proper subset of the class of permutation graphs [36]. By the result of [34], an $O(n\log^2 n)$ time algorithm exists for the problem on weighted cographs. This section improves the result to $O(n)$ by working on the cotree of the original cograph. The strategy used is the dynamic programming strategy.

Given a cograph G with costs on vertices and its corresponding cotree T, for any non-leaf node r of T, denote the subcotree rooted at r to be $T(r)$ and the subcograph induced by the leaves of $T(r)$ to be $G(r)$. Denote $\delta(r)$ to be the value of the bottleneck cost of an optimal solution of the BIDS problem on the cograph $G(r)$, i.e., $\delta(r) = \min\{\beta(H) \mid H$ is an independent dominating set of $G(r)$, where $\beta(H)$ is the bottleneck cost of $H\}$.

From the definition of cographs, any non-leaf node r should be either a 1-node, denoted by r^1, or a 0-node, denoted by r^0. Let u_1 and u_2 be its two children. The following considers the two cases for any non-leaf node r.

Case 1. 0-node, r^0: In this case, no vertex in $G(u_1)$ is adjacent to any vertex in $G(u_2)$, i.e., $G(r^0)$ is just the union of $G(u_1)$ and $G(u_2)$. An optimal solution on $G(r^0)$ is merely the union of any optimal solutions on $G(u_1)$ and $G(u_2)$, respectively, i.e., the BIDS problem on $G(u_1)$ and $G(u_2)$ can be solved independently and recursively. Therefore, the minimum bottleneck cost under this case, denoted by $\delta(r^0)$, is equal to $\max\{\delta(u_1), \delta(u_2)\}$.

Case 2. 1-node, r^1: In this case, all vertices in $G(u_1)$ are adjacent to all vertices in $G(u_2)$. Let the vertices of $G(u_1)$,

$V(G(u_1)) = X = \{x_1, ..., x_m\}$ and the vertices of $G(u_2)$, $V(G(u_2)) = Y = \{y_1, ..., y_n\}$. Based on the definition of cographs, $G(r^1)$ is constructed from the union of $G(u_1)$ and $G(u_2)$ with additional new edges connecting all pairs (x_i, y_j). The subgraph induced by the new edges connecting X and Y forms a complete-bipartite graph.

If any vertex in $G(u_1)$, say x_i, is included in any optimal solution of $G(r^1)$, then all vertices in $G(u_2)$ must be excluded since they are all dominated by x_i. Similarly, if any vertex in $G(u_2)$, say y_j, is included in any optimal solution of $G(r^1)$, then all vertices in $G(u_1)$ must be excluded. Therefore, the minimum bottleneck cost under this case, denoted by $\delta(r^1)$, can be easily proved to be equal to $\min\{\delta(u_1), \delta(u_2)\}$.

From the above discussions, an optimal solution can be identified by examining each internal node once from the root r after $\delta(r)$ has been computed, and its time-complexity is $O(n)$.

Theorem 6: The BIDS problem can be solved in $O(n)$ time on weighted cographs.

5. THE CONCLUSIONS

This paper discusses the Bottleneck Independent Dominating Set problem on graphs with positive costs on vertices. The results achieved in this paper can be summarized in Table 1.

Some directions are worthy to continue in the future.

1. The approach used in this study can be easily applied to solve this problem on other classes of graphs, such as interval graphs and block graphs.
2. Identify other types of dominating sets, e.g., perfect dominating sets and connected dominating set, etc., with minimum bottleneck costs on weighted graphs.
3. Find out the relationships between bottleneck problems and summation problems on weighted graphs. This is a very interesting and practical research direction.

Table. 1. The complexities of the BIDS problem achieved in this paper.

Class of graphs	**Complexity**	**Strategy**
Chordal Graphs	NP-hard	--
Weighted Split Graphs	$O(n + m)$	Greedy
Planar-Bipartite Graphs	NP-hard	--
Weighted Convex-Bipartite Graphs	$O(m)$	Dynamic Programming
Weighted Cographs	$O(n)$	Dynamic Programming

REFERENCES

[1] K. Abrahamson, N. Dadoun, D. G. Kirkpatrick and T. Przytycka, ″A Simple Parallel Tree Contraction Algorithm,″ J. Algorithms, Vol. 10, pp. 282-302, 1989.

[2] G. S. Adhar and T. Peng, ″Parallel algorithm for Path Covering, Hamiltonian Path, and Hamiltonian Cycle in Cographs,″ Proc. Internat. Conf. on Parallel Processing, Vol. III 364-365 (1990).

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

[4] A. A. Bertossi, ″Total Domination in Interval Graphs,″ Information Processing Letters, Vol. 23, pp. 131-134, 1986.

[5] C. Bo and B. Liu, ″Some Inequalities about Connected Domination Number,″ Discrete Mathematics, Vol. 159, pp. 241-245, 1996.

[6] G. J. Chang, ″Total Domination in Block Graphs,″ Operations Research Letters, Vol. 8, pp. 53-57, 1989.

[7] D. G. Corneil, H. Lerchs and L. S. Burlingham, ″Complement Reducible Graphs,″ Discrete Appl. Math., Vol. 3, pp. 163-174, 1981.

[8] D. G. Corneil and Y. Perl, ″Clustering and Domination in Perfect Graphs,″ Discrete Appl. Math., Vol. 9, pp. 27-39, 1984.

[9] D. G. Corneil, Y. Perl and L. K. Stewart, ″Cographs:Recognition, Applications, and Algorithms, Proc. 15th Southeastern Conf. on Combinatorics, Graph Theory and Computing (1984).

[10] D. G. Corneil, Y. Perl and L. K. Stewart, ″A Linear Recognition Algorithm for Cographs,″ SIAM J. Comput., Vol. 14, pp. 926-934, 1985.

[11] Peter DAMASCHKE, Haiko MÜLLER, and Dieter KRATSCH, ″Domination in Convex and Chordal Bipartite Graphs,″ Information Processing Letters, Vol. 36, pp. 231-236, 1990.

[12] M. Farber, ″Independent Domination in Chordal Graphs,″ Operation Research Letters, Vol. 1, pp. 134-138, 1982.

[13] M. Farber, ″Domination, Independent Domination and Duality in Strongly Chordal Graphs,″ Discrete Applied Mathematics, Vol. 7, pp. 115-130, 1984.

[14] S. Folds and P. L. Hammer, ″Split Graphs,″ Proc. 8th Southeastern Conf. on Combinatorics, Graph Theory and Computing (F. Hoffman et al., eds), Louisiana State University, Baton Rouge, Louisiana, 311-315 (1977).

[15] 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.

[16] F. Gavril, ″The Intersection Graphs of Subtrees in Tree Are Exactly the Chordal Graphs,″ Journal of Combinatorial Theory, Vol. 16, pp. 47-56, 1974.

[17] R. Gould, Graph Theory, The Benjamin/Cummings Publishing Company, Inc., Menlo Park, California, 1988.

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

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

[20] S. T. Hedetniemi and R. Laskar, ″Connected Domination in Graphs,″ In B. Bollobas, editor, Graph Theory and Combinatorics, pp. 209-218, Academic Press, London, 1984.

[21] R. W. Irving, ″On Approximating the Minimum Independent Dominating Set,″ Information Processing Letters, Vol. 37, pp. 197-200, 1991.

[22] J. K. Keil, ″Total Domination in Interval Graphs,″ Information Processing Letters, Vol. 22, pp. 171-174, 1986.

[23] H. Kim, ″Finding a Maximum Independent Set in a Permutation Graph,″ Information Processing Letters, Vol. 36, pp. 19-23, 1990.

[24] D. E. Knuth, The Art of Computer Programming, Vol. 1, Fundamental Algorithms, Addison-Wesley, Reading, MA (1968).

[25] D. Lichtenstein, ″Planar Formulae and Their Uses,″ SIAM Journal on Computing, Vol. 11, No. 2, pp. 329-342, 1982.

[26] 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.

[27] K. S. Natarajan and L. J. White, ″Optimum Domination in Weighted Trees,″ Information Processing Letters, Vol. 7, pp. 261-265, 1987.

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

[29] 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.

[30] G. Ramalingam and C. P. Rangan, ″Total Domination in Interval Graphs Revisited,″ Information Processing Letters, Vol. 27, pp. 17-21, 1988.

[31] E. Sampathkumar and H. B. Walikar, ″The Connected Domination Number of A Graph,″ Journal of Mathematical Physical Sciences, Vol. 13, pp. 607-613, 1979.

[32] C. C. Yen and R. C. T. Lee, ″The Weighted Perfect Domination Problem,″ Information Processing Letters, Vol. 35, pp. 295-299, 1990.

[33] 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, Vol. 73, pp. 19-26, 1994.

[34] W. C. K. Yen and C. Y. Tang, ″The Bottleneck Independent Dominating on Permutation Graphs,″ Proc. International Computer Symposium′92, Vol. I, 455-462 (1992).

[35] W. C. K. Yen and C. Y. Tang, ″The Searchlight Guarding Problem on Weighted Split Graphs and Weighted Cographs,″ Networks, Vol. 35, pp. 195-206, 2000.

[36] C. W. Yu and G. H. Chen, Properties and Algorithms on Permutation Graphs and Doubly Convex-Bipartite Graphs, Ph.D. Dissertation, National Taiwan University, (1993).

[37] M. S. Yu and C. H. Yang, ″An $O(n)$ Time Algorithm for Maximum Matching on Cographs,″ Information Processing Letters, Vol. 47, pp. 89-93, 1993.