ON APPROXIMATING A VERTEX COVER FOR PLANAR GRAPHS

R. Bar-Yehuda and S. Even
Computer Science Department
Technion - Israel Institute of Technology
Haifa, Israel

ABSTRACT

The approximation problem for vertex cover of n-vertex planar graphs is treated. Two results are presented:

(1) A linear time approximation algorithm for which the (error) performance bound is 2/3.

(2) An O(n log n) time approximation scheme.

1. INTRODUCTION

Let G(V,E) be a simple undirected graph. A subset of vertices is called a *vertex cover* if every edge has at least one endpoint in the subset. The vertex cover problem is, given G, find a vertex cover of minimum cardinality.

The vertex cover problem is known to be NP-hard, even if the graphs are restricted to be cubic or planar with maximum vertex degree 4 [1]. Thus, it is natural to look for efficient approximation algorithms.

Let A be an approximation algorithm. Denote by VC_A the vertex cover which A produces for G, and by VC^* a minimum vertex cover of G. Let $\varepsilon > 0$. We say that ε is a *performance bound* of A if for every graph

$$\frac{|VC_A| - |VC^*|}{|VC^*|} \leqslant \varepsilon \quad .$$

Gavril suggested a linear-time approximation algorithm for which $\varepsilon = 1$ (see [1] page 134). Hochbaum [2] showed a polynomial-time (linear programming) approximation algorithm for weighted graphs, for which $\varepsilon = 1$, and Bar-Yehuda and Even [3] achieved the same performance bound with a linear-time algorithm. Hochbaum [4] found an $O(n^2 \log n)$ approximation algorithm for planar graphs, for which $\varepsilon = 3/5$, and polynomial-time approximation algorithm for planar graphs, for which $\varepsilon = 1/2$; all her results are for weighted graphs.

In Section 2 we present a linear-time approximation algorithm for planar graphs, with $\varepsilon = 2/3$.

In Section 3 we prove some facts about graphs and planar graphs, which are useful in the following section. One of the more interesting facts is that for planar graphs, $|VC^*|$ is greater than one third of the number of vertices whose degree is greater than 2.

An *approximation scheme* is a scheme which for every $\varepsilon > 0$ yields a polynomial-time approximation algorithm with performance bound ε. In Section 4 we present an approximation scheme for vertex cover of planar graphs which is O(n log n) time. Here we use the result of Lipton and Tarjan [5], i.e. approximation scheme for finding a maximum independent set of planar graph.

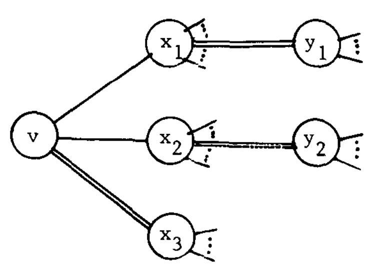
2. LINEAR-TIME APPROXIMATION ALGORITHM FOR PLANAR GRAPHS, WITH $\varepsilon = 2/3$

Gavril's approximation algorithm ($\varepsilon = 1$) is as follows: Find a maximal (not necessarily a maximum) matching of G. VC consists of all the endpoints of edges in the matching.

It is easy to see that this algorithm yields a VC which satisfies $|VC|/|VC^*| \leq 2$; each edge in the matching requires at least one of its endpoints to be in every vertex cover, and we use both. Another way of looking at this is as follows: Choose an edge $u - v$. This edge requires either u or v to be in the vertex cover; choose both and eliminate u, v and all their incident edger from G. The local performance bound is 1, and therefore the global performance bound is also 1.

We use a similar approach, but instead of relying on single edges we rely on subgraphs for which we get a better local performance bound. For example, a triangle requires at least two of its vertices to be in every vertex cover, and by using all three we get a local bound of 1/2.

The subgraph shown in the diagram below consists of a 3-edge matching.



Thus it requires at least 3 vertices in every vertex cover of the graph. We use x_1, x_2, x_3, y_1, y_2 and thus, the local performance bound is 2/3. As before, all the vertices $(x_1, x_2, x_3, y_1, y_2, v)$ are removed with all their incident edges, since all these edges are already covered.

Consider the following algorithm. The instructions in square brackets are not necessary, and are added for analysis of the algorithm only. Let $\tilde{G}(U)$ be the subgraph of $G(V,E)$, induced by $U \subseteq V$.

Algorithm A (Input: $G(V,E)$. Output: VC)

(0) $VC \leftarrow \emptyset$, $U \leftarrow V$, $[i \leftarrow 0]$.

(1) (Phase 1. During this phase the graph is reduced into a triangle-free subgraph.)

(1.1) While $\tilde{G}(U)$ contains triangles do $[i \leftarrow i+1, d_i \leftarrow 2]$.

(1.2) Let $X \subseteq U$ s.t. $\tilde{G}(U)$ is a triangle.

(1.3) $VC \leftarrow VC \cup X$, $U \leftarrow U-X$.

(1.4) End

(2) (Phase 2)

(2.1) While $U \neq \emptyset$ do

(2.2) Let v be a vertex of $\tilde{G}(U)$ of minimum degree d, $[i \leftarrow i+1, d_i \leftarrow d]$.

(2.3) Let $X = \{x | v - x \text{ in } \tilde{G}(U)\} = \{x_1, x_2, \ldots, x_d\}$,
$Y = \{y | y \in U-\{v\} \text{ and for some } x \in X, x - y \text{ in } \tilde{G}(U)\}$,
$E' = \{x - y | x \in X, y \in Y \text{ and } x - y \text{ in } \tilde{G}(U)\}$.

(2.4) In the bipartite graph (X,Y,E') we find a complete matching of $\{x_1, x_2, \ldots, x_{d-1}\}$ into Y. Let Y' be the set of $d-1$ vertices of Y which are used in the matching.

(2.5) $VC \leftarrow VC \cup X \cup Y'$, $U \leftarrow U - \{v\} - X - Y'$.

(2.6) End

(3) Stop.

Let e be the number of iterations (the final value of i) and $\tilde{d} = (\sum_{i=1}^{e} d_i)/e$.

Observe that in (X,Y,E'), of Step (2.4), the degree of each $x \in X$ is at least $d-1$. Thus, the task of finding a complete $\{x_1, \ldots, x_{d-1}\}$ into Y is trivial.

Theorem 2.1: The vertex cover, VC, produced by A, satisfies $|VC|/|VC^*| \leq 2 - (1/\tilde{d})$.

Proof: In Phase 1, each triangle requires 2 $(=d_i)$ vertices in any vertex cover, while A uses 3 $(=2 \cdot d_i - 1)$. In Phase 2, each $\tilde{G}(\{v\} \cup X \cup Y')$ requires d_i vertices in any vertex cover, since the matching of $\{x_1, \ldots, x_{d-1}\}$ to Y', plus the edge $v - x_d$ is a matching of cardinality d_i; A uses $2d_i - 1$ instead. We conclude that

$|VC^*| \geq \sum_{i=1}^{e} d_i$, while $|VC| = \sum_{i=1}^{e} (2 \cdot d_i - 1)$. Thus,

$$\frac{|VC|}{|VC^*|} \leq \frac{2\Sigma d_i - e}{\Sigma d_i} = 2 - \frac{1}{\tilde{d}} .$$

Q.E.D.

In order to analyze the complexity of Algorithm A, we use in Phase 1 the algorithm of Itai and Rodeh [6] for finding triangles in a graph.

Algorithm B

```
While there are edges in the graph do
    For every connected component do
        Find a spanning tree.
        Find a maximal set of disjoint triangles
        which contain at least one tree-edge.
        Delete all the edges of the tree from
        the component.
    End
End
```

They prove (Theorem 3) that for any graph this algorithm is $O(|E|^{3/2})$ time and $O(|V|^2)$ space. For Phase 2, one can use a priority queue [7] for the vertices, according to their degrees; each extraction of a vertex of minimum degree and each vertex deletion and insertion is $O(\log |V|)$, while the number of such operations is $O(|E|)$. Thus the complexity of Algorithm A is $O(|E|(\log |V|+|E|^{1/2}))$ time and $O(|V|^2)$ space. Obviously, A is $O(|E|^{3/2})$ time if there are no isolated vertices.

Theorem 2.2: For planar graphs, the vertex cover, VC, produced by A satisfies $|VC|/|VC^*| \leqslant 5/3$.

Proof: Let us show that for planar graphs $d_i \leqslant 3$ for $i = 1,2,\ldots,e$. This is obvious in Phase 1, since there $d_i = 2$.

The graph $\tilde{G}(U)$, when Phase 2 begins, contains no triangles. Let i be an iteration for which the minimum degree of the vertices, d_i, of the current $\tilde{G}(U)$, is greater than 3.

Since $\tilde{G}(U)$ is simple (no parallel edges and no self-loops) and triangle-free, the boundary of each of $\tilde{G}$'s faces contains at least 4 edges. Each edge is counted twice: If it is a bridge, it is counted twice in the face-boundary it belongs to. If it is not a bridge, it is counted once in each of the two face-boundaries it belongs to.

Denote by f the number of faces of $\tilde{G}(U)$. Thus,

$$2\cdot|E| \geqslant 4\cdot f . \qquad (*)$$

Euler's formula for planar graph states that

$$|V| + f - |E| \geqslant 2.$$

Since, by (*) $|E| \geqslant 2\cdot f$, Euler's formual implies that $2|V| - |E| \geqslant 4$, or

$$|E| < 2\cdot|V| .$$

Thus,

$$\sum_{v\in V} d(v) = 2\cdot|E| < 4\cdot|V| ,$$

and the average vertex degree, $\bar{d}$, satisfies

$$\bar{d} = \frac{\Sigma d(v)}{|V|} < 4 ,$$

contradicting the assumption that $d_i > 3$.

If follows that $\tilde{d} \leqslant 3$, and by Theorem 2.1, Theorem 2.2 holds.

Q.E.D.

Let us, next, consider the time complexity of Algorithm A, when applied to planar graphs.

For Phase 1, we use again Algorithm B. Itai and Rodeh prove (Theorem 2) that for planar graphs the algorithm is $O(|V|)$ time, and $O(|V|^2)$ space. Let us show that the space can be cut to $O(|V|)$. Specifically, we change the implementation of the search of a maximal set of disjoint triangles, by using a DFS tree [8]. This does not change the analysis of the running time.

Let us denote a tree-edge from vertex u to vertex v by $u \to v$, and a back-edge from v to u by $v \dashrightarrow u$. If $u \to v$ then $f(v) = u$. Also, for each vertex v we define an auxiliary label, $\lambda(v)$, which will be helpful in the detection of triangles.

There are 3 types of triangles which contain a tree-edge. If the vertices of the triangle are x, y and z the types are as follows:

Type 1: $x \dashrightarrow y$, $x \dashrightarrow z$, $z \to y$.

Type 2: $x \to y$, $x \dashrightarrow z$, $y \dashrightarrow z$.

Type 3: $x \dashrightarrow z$, $z \to y$, $y \to x$.

Algorithm C (Input: G(V,E) which has been scanned by DFS. Output: A set S of vertices which form a maximal vertex-disjoint set of triangles.)

```
S ← ∅, for every x ∈ V, λ(x) ← 0.
For every x ∈ V do
    for every x-→y, λ(y) ← x.
    [Type 1]
    for every x-→y do
        if λ(f(y)) = x and {x,y,f(y)} ∩ S = ∅
        then S ← S ∪ {x,y,f(y)}.
    end
    [Type 2]
    for every x→y do
        for every y-→z do
            if λ(z) = x and {x,y,z} ∩ S = ∅
            then S ← S ∪ {x,y,z}.
        end
    end
    [Type 3]
    if λ(f(f(x))) = x and {x,f(x),f(f(x))} ∩ S = ∅
    then S ← S ∩ {x,f(x),f(f(x))}.
End.
```

This algorithm is linear in the number of edges, since every edge is used no more than some constant (3) times. The algorithm can be made more efficient, at the cost of complicating its structure, by avoided repeated checks for a vertex which has already joined S. Also, by dropping the '<u>and</u> $\{\ldots\} \cap S = \emptyset$' parts, the algorithm produces all the triangles.

For Phase 2 of Algorithm A, the vertices are put on doubly-linked lists, one list for every possible degree. Also, a table is used to find the vertex occurrence and its degree, so that these data of a vertex can be determined in constant time. When an edge is eliminated from G, while one of its endpoints is eliminated, its other endpoint is found, removed from the list it is in, and added to the list of degree one less. By the proof of Theorem 2.2, the minimum degree of the vertices is at most 3. Thus, one can find a vertex of minimum degree in constant time. It follows that Phase 2 can also be implemented to run in $O(|V|)$ time and $O(|V|)$ space.

3. SOME GRAPH-THEORETIC FACTS

<u>Theorem 3.1</u>: Let (X,Y,E) be a non-empty planar bipartite graph. If the degree of all the vertices in X is 3 then $|X| < 2 \cdot |Y|$.

<u>Proof</u>: Clearly, the graph has at least 4 vertices, and contains no triangles. As in the proof of Theorem 2.2, $|E| < 2|X \cup Y|$. However, $|E| = 3 \cdot |X|$. Thus, $|X| < 2 \cdot |Y|$.

Q.E.D.

<u>Corollary 3.1</u>: Let (X,Y,E) be a non-empty planar bipartite graph. If the degree of all the vertices in X is greater than 2 then $|X| < 2 \cdot |Y|$.

<u>Proof</u>: As long as there is a vertex in x whose degree is greater than 3, eliminate one of its edges. The resulting graph is as in the premise of Theorem 3.1.

Q.E.D.

<u>Theorem 3.2</u>: If G(V,E) is a planar graph and $X = \{x | x \in V \text{ and } d(x) \geq 3\}$ is non-empty then the following 3 conditions hold:

(i) $|VC^*| > \frac{1}{3}|X|$,

(ii) $|IS^*| < |V| - \frac{1}{3}|X|$,

(iii) $|M| > \frac{1}{6}|X|$.

Here IS^* is a maximum independent set and M is any maximal matching. (Actually, (ii) and (iii) are not necessary for what follows.)

<u>Proof</u>: Let VC^* be a minimum vertex cover. The set $V-VC^*$ is therefore a maximum independent set, to be denoted IS^*. Let $X_1 = X \cap VC^*$ and $X_2 = X \cap IS^*$. Consider now the bipartite graph, $B(X_2, VC^*, E')$, induced by X_2 and VC^* as a subgraph of G; i.e.

$$E' = \{x - y | x \in X_2,\ y \in VC^* \text{ and } x - y \text{ in } G\}.$$

The degree of $x \in X_2$ in B is the same as in G, and thus is greater than 2. Now, Corollary 3.1 is applicable, and $|X_2| < 2 \cdot |VC^*|$. But $X_2 = X - X_1$, and $X_1 \subseteq VC^*$. Thus,

$$2|VC^*| > |X_2| \geq |X| - |VC^*|$$

and (i) follows.

Condition (ii) follows immediately from (i) and $|VC^*| + |IS^*| = |V|$.

If M is a maximal matching then $|VC^*| \leq 2 \cdot |M|$. By (i), (iii) follows.

Q.E.D.

<u>Theorem 3.3</u>: If B(X,Y,E) is a circuit-free bipartite graph and $d_X = \min_{x \in X} d(x)$, then

$$(d_X - 1) \cdot |X| < |Y| .$$

<u>Proof</u>: Delete edges from B until all vertices of X are of degree d_X. The resulting subgraph, $B'(X,Y,E')$ satisfies $|E'| = d_X \cdot |X|$. Also B and B' are forests. Thus, $|X| + |Y| > |E'|$ and the Theorem follows.

Q.E.D.

<u>Corollary 3.2</u>: If B(X,Y,E) is a circuit-free bipartite graph and for every $x \in X$, $d(x) = 2$, then $|X| < |Y|$.

A *rosary* is a simple circuit of G, $v_1 - v_2 - \ldots - v_{2\ell} - v_1$, such that for every $i = 1,2,\ldots,\ell$ $d(v_{2i}) = 2$.

A *bead* is a vertex of degree 2 which plays a role of v_{2i} in some rosary.

<u>Theorem 3.4:</u> Let G(V,E) be a simple planar graph. If the degree of every vertex is greater than 1 and if $\bar{Y}$ is the set of beads in G, then $|VC^*| > \frac{1}{5}(|V| - |\bar{Y}|)$.

Proof: Let $X = \{x | x \in V$ and $d(x) \geq 3\}$ and $Y = \{y | y \in V$ and $d(y) = 2\}$. Clearly, $X \cup Y = V$. Let VC^* be a minimum vertex cover and $IS^* = V - VC^*$. Define $Y_1 = Y \cap IS^*$ and $Y_2 = Y \cap VC^*$. Consider the bipartite graph $B(Y_1-\bar{Y}, VC^*, E')$, where $E' = \{y - v | y \in Y_1-\bar{Y}, v \in VC^*$ and $y - v$ in $G\}$. The degree of every vertex of $Y_1-\bar{Y}$, in B, is the same as in G, namely 2. Also, B is circuit-free; a circuit in B is a rosary of G, and its vertices in $Y_1-\bar{Y}$ are beads, contradicting the definition of $\bar{Y}$. Thus, by Corollary 3.2,

$$|VC^*| > |Y_1| - |\bar{Y}| .$$

By Theorem 3.2 (i)

$$3 \cdot |VC^*| > |X| .$$

Thus,

$$4 \cdot |VC^*| > |X| + |Y_1| - |\bar{Y}| = |V| - |Y_2| - |\bar{Y}|.$$

However

$$|VC^*| \geq |Y_2| .$$

We conclude that

$$5 \cdot |VC^*| > |V| - |\bar{Y}|.$$

Q.E.D.

Theorem 3.5: Let $B(X,Y,E)$ be a bipartite graph in which for every $x \in X$, $d(x) \geq 2$, and for every $y \in Y$, $d(y) \leq 2$. X is a minimum vertex cover of B.

Proof: By induction on $|Y|$. If $|Y| = 0$, then $|X| = 0$, $E = \emptyset$, and the theorem holds trivially.

Assume that the theorem holds for $|Y| < m$, and let $B(X,Y,E)$ be such that $|Y| = m$. Let y be a vertex in Y of minimum degree.

If $d(y) = 0$ then delete y from B. By the inductive hypothesis, the theorem holds on the resulting graph, and therefore is true also for the original graph.

If $d(y) = 1$, let $x - y$ be its incident edge. Put x in VC^* and delete x, y and all their incident edges from B. The resulting graph, $B'(X-\{x\}, Y-\{y\}, E')$ satisfies the premise of the theorem. Thus, by the inductive hypothesis, $X-\{x\}$ is a minimum vertex cover of B'. It follows that X is a minimum vertex cover of B.

If $d(y) = 2$ then the degrees of all vertices of B are greater than or equal to 2. Thus, B contains circuits. Choose a simple circuit C. C is a rosary. Clearly, the vertices of X in C are a minimum vertex cover of the edges of C. Put these vertices in VC^* and delete all the vertices of C and their incident edges from B. The resulting graph $B'(X',Y',E')$ satisfies the premise of the theorem. Thus, by the inductive hypothesis, X' is a minimum vertex cover of B'. It follows that X is a minimum vertex cover of B.

Q.E.D.

4. AN O(n log n)-TIME APPROXIMATION SCHEME OF THE VC PROBLEM FOR PLANAR GRAPHS

Lipton and Tarjan [5] describe an approximation scheme for the independent set problem of n-vertex planar graphs. The time complexity is $O(n \log n)$, while the performance bound is $O(1/\sqrt{\log \log n})$. Thus, for some constant c,

$$\frac{|IS^*|-|IS|}{|IS^*|} \leq \frac{c}{\sqrt{\log \log n}} . \qquad (*)$$

This algorithm produces an approximation for the VC problem by using $VC = V - IS$. Since $|IS^*|+|VC^*|=|V|=n$, inequality (*) yields

$$\frac{|VC|-|VC^*|}{|VC^*|} \leq \frac{c \cdot \left(\frac{n}{|VC^*|} - 1\right)}{\sqrt{\log \log n}} .$$

In order to show that this is an approximation scheme for the VC problem, it suffices to show that for some $\varepsilon_o > 1$ and n_o, if $n > n_o$, then $n/|VC^*| < \varepsilon_o$.

We shall achieve this condition by first applying a preparatory algorithm in which a vertex cover is derived for an induces subgraph, without any local performace loss, and the residual graph will satisfy the above. We shall see that the time complexity of this preparatory algorithm is $O(n)$. Thus, the combined approximation scheme is $O(n \log n)$-time and its performance bound is $O(1/\sqrt{\log \log n})$.

A graph $G(V,E)$ is called *proper* if it is simple, planar, the degree of its vertices is greater than 1 and the set $\bar{Y}$ of all its beads satisfies $|\bar{Y}| < \frac{1}{6}|V|$.

By Theorem 3.4, if G is proper then $|V|/|VC^*| < 6$. Thus, it suffices to ensure that the residual graph be proper.

The general structure of the preparatory algorithm is as follows: (Input: A simple planar graph $G(V,E)$. Output: A minimum vertex cover for $\tilde{G}(U)$,

$U \subseteq V$, and a residual graph, $\tilde{G}(V-U)$, which is proper.)

(0.1) $VC \leftarrow \emptyset$, $U \leftarrow \emptyset$, TEST $\leftarrow 1$

(0.2) While TEST = 1 do

(1.1) While there are vertices of degree less than 2 in $\tilde{G}(V-U)$ do

(1.2) If $v \in V-U$, $d(v) = 0$ in $\tilde{G}(V-U)$ then add v to U.

(1.3) If $v \in V-U$, $d(v) = 1$ in $\tilde{G}(V-U)$ and $v - u$ is its edge then add v, u to U and u to VC. End.

(2.1) While there are isolated simple circuits in $\tilde{G}(V-U)$ do

(2.2) Let $v_1 - v_2 - \ldots - v_\ell - v_1$ be a simple isolated circuit in $\tilde{G}(V-U)$. Add all vertices v_i, $1 \leq i \leq \ell$ and odd, to VC. Add all v_i $1 \leq i \leq \ell$ to U. End.

(3.1) Find $B(X,Y,E')$ a bipartite subgraph of $\tilde{G}(V-U)$. (Y contains all the beads of $\tilde{G}(V-U)$; if $y \in Y$ then $d(y) = 2$ in B; if $x \in X$ then $d(x) \geq 2$ in B. A detailed construction of such a B will be given shortly.)

(3.2) If $|Y| \geq \frac{1}{6}|V-U|$ then $VC \leftarrow VC \cup X$
$U \leftarrow U \cup X \cup Y$

(4.1) If $|Y| < \frac{1}{6}|V-U|$ or $V-U = \emptyset$ then TEST $\leftarrow 0$.

(4.2) End.

Let us discuss first the construction of $B(X,Y,E')$ of Step (3.1). This is done as follows: (Input: A graph $G(V,E)$; all its vertices are of degree greater than 1 and there are no isolated simple circuits. Output: $B(X,Y,E')$.)

(1) Label every $v \in V$ whose degree is greater than 2 by '+', and put it in a queue Q.

(2) While Q not empty do

(3) Remove the first vertex, v, from Q.

(4) Label all the unlabelled vertices, which are adjacent to v, with the opposite label (if v is labelled '+', label them '-'; if v is labelled '-', label them '+') and put them in Q. End.

(5) Let $X = \{x | x \in V \text{ and is labelled '+'}\}$
$Y = \{y | y \in V \text{ and is labelled '-'}\}$
$E' = \{x - y | x \in X,\ y \in Y \text{ and } x - y \text{ in } G\}$.

(6) While there are vertices of degree less than 2 in (X,Y,E'), delete them and their incident edges.

(7) The resulting bipartite graph is $B(X,Y,E')$.

(8) Stop.

Lemma 4.1: If y is a bead in G then $y \in Y$.

Proof: Let $y_1 - x_1 - y_2 - \ldots - y_\ell - x_\ell - y_1$ be a rosary in G. Since G contains no isolated simple circuit, some of the x-vertices are of degree greater than 2 and are labelled '+' in Step (1). All the other vertices of the rosary are of degree 2 and therefore their labeling cannot be effected by a vertex not in the rosary. It follows that every bead will eventually be labelled '-'. Also, none of the vertices of a rosary will be deleted in Step (6).

Q.E.D.

It is easy to see that the construction of B is $O(n)$ time. Also, the total time spent in each of the Steps (1.2), (1.3) and (3.2) is $O(n)$, while each time Step (2.2) or (3.1) is applied, the time spent is $O(n)$. Note that the number of vertices in V-U is reduced by a factor of at least 5/6 in each iteration, except possibly the last.

Let cn_i be an upper bound on the time complexity of the i-th iteration, where n_i is the number of vertices in the current graph. Since $n_i \leq \left(\frac{5}{6}\right)^i \cdot n$, the total time is bounded by

$$\sum_{i=0}^{\infty} cn_i \leq c \cdot n \sum_{i=0}^{\infty} \left(\frac{5}{6}\right)^i = c \cdot n \cdot 6 = O(n).$$

If the preparatory algorithm halts with $V-U = \emptyset$ then VC is a minimum vertex cover. The reason for this is as follows: Each induced subgraph which is delted in each of the Steps (1.2), (1.3) and (2.2) is covered in a minimum fashion. If $B(X,Y,E')$ satisfies $|Y| \geq \frac{1}{6}|V-U|$, then by Theorem 3.5, X is a minimum cover of $\tilde{G}(X \cup Y)$.

If the preparatory algorithm halts with $|Y| < \frac{1}{6}|V-U|$, then by Lemma 4.1, $\bar{Y}$, the set of beads of $\tilde{G}(V-U)$, satisfies $\bar{Y} \subseteq Y$. Thus, by Theorem 3.4, the minimum vertex cover of $\tilde{G}(V-U)$, VC^*, satisfies

$$|VC^*| > \frac{1}{5}(|V-U| - |\bar{Y}|) \geq \frac{1}{6}|V-U| ,$$

and the Lipton-Tarjan approximation scheme can be applied to the residual graph, $\tilde{G}(V-U)$, as discussed in the beginning of this section.

ACKNOWLEDGEMENT

The authors wish to thank Dr. Michael Rodeh for his comment which led to the linear-time complexity analysis of the algorithm of Section 4.

REFERENCES

[1] Garey, M.R., and Johnson, D.S., Computers and Intractability; A Guide to the Theory of NP-Completeness, Freeman, 1979.

[2] Hochbaum, D.S., "Approximation Algorithm for the Weighted Set Covering and Node Cover Problems", W.P. #64-79-80, GSIA, Carnegie-Mellon University, April 1980.

[3] Bar-Yehuda, R., and Even, S., "A Linear-Time Approximation Algorithm for the Weighted Vertex Cover Problems", J. of Algorithms, Vol. 2, 1981, pp. 198-203.

[4] Hochbaum, D.S., "Efficient Bounds for the Stable Set, Vertex Cover and Set Packing Problems", W.P. #50-80-81, GSIA, Carnegie-Mellon University, May 1981.

[5] Lipton, R.J., and Tarjan, R.E., "Applications of a Planar Separator Theorem", SIAM J. Comput., Vol. 9, No.3, August 1980, pp. 615-627.

[6] Itai, A., and Rodeh, M., "Finding a Minimum Circuit in a Graph", SIAM J. Comput., Vol. 7, No. 4, November 1978, pp. 413-423.

[7] Aho, A.V., Hopcroft, J.E., and Ullman, J.D., The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.

[8] Tarjan, R.E., "Depth-First Search and Linear Graph Algorithms," SIAM J. Comput., Vol. 1, 1972, pp. 146-160.