

On-line algorithms for the dominating set problem¹

Gow-Hsing King, Wen-Guey Tzeng*

Department of Computer and Information Science, National Chiao Tung University, Hsinchu 30050, Taiwan

Received 10 July 1996; revised 3 October 1996

Communicated by D. Gries

Keywords: Algorithms; On-line algorithms; Dominating set

1. Introduction

Let $G = (V, E)$ with $V = \{1, 2, \dots, n\}$ be an undirected graph. A dominating set of G is a subset $V' \subseteq V$ such that for each vertex $u \in V - V'$ there is a vertex $v \in V'$ so that $(u, v) \in E$. The minimum dominating set problem is to find a set V' of the minimum cardinality, which is known to be NP-complete [3]. In this paper we consider the on-line version of the problem in two settings. The first setting is that all vertices $1, 2, \dots, n$ in the graph are given in advance. At time interval i the adjacency condition of vertex i to the other vertices is presented. We propose an on-line algorithm of performance ratio $1.5\sqrt{n} + c_1$ and show that $\sqrt{n} - c_2$ is a lower bound for the performance ratio that an on-line dominating set algorithm can possibly achieve, where c_1 and c_2 are some positive constants. The second setting is that at time interval i the adjacency condition of vertex i to vertices $j, j < i$, is given. For this setting we show that performance ratio $n - 1$ is the tight bound.

2. The first on-line setting

Suppose all vertices of the graph G are given in advance. At time i the adjacency condition of a vertex i (say) to the other vertices is given. Therefore, without loss of generality, vertices are given in sequence $1, 2, \dots, n$. We present our on-line dominating set algorithm JUMP. Let D_i be the set of vertices selected by JUMP at time i , where $D_0 = \emptyset$. By the nature of the on-line algorithm, $D_i \subseteq D_{i+1}$ and $D = D_n$ is the dominating set returned by JUMP. Let $(i, i_1), (i, i_2), \dots, (i, i_{d_i})$ be the adjacent edges of vertex i given at time i , where d_i is the degree of vertex i and $1 \leq i_1 < i_2 < \dots < i_{d_i} \leq n$. After time i , vertex $u \in V$ is marked as “dominated” if $u \in D_i$ or there is a vertex $v \in D_i$ such that $(u, v) \in E$. A vertex that is not “dominated” is an “undominated” vertex. Also, the vertices $j, j \leq i$, that are presented before are called “visited”. There are four cases in the following by which the vertex i is put into D_i at time i . A vertex in D is classified into one of the three types: *j-vertex*, *r-vertex*, and *f-vertex*. Note that the current vertex is checked from case 1 to case 4 one by one, so a vertex cannot be of two types. We also define a function f on some vertices for later analysis.

* Corresponding author. Email: tzeng@cis.nctu.edu.tw.

¹ This research was supported in part by the National Science Council, Taiwan, R.O.C., under grant NSC 86-2213-E-009-024.

- (1) There are $\geq \lceil \sqrt{n} \rceil$ “undominated” vertices among i_1, i_2, \dots, i_{d_i} . For this case vertex i is called a j -vertex in D .
- (2) Vertex i dominates all the remaining “undominated” vertices. For this case vertex i is called an r -vertex.
- (3) All vertices i_s , $1 \leq s \leq d_i$ are less than i and not in D_{i-1} , that is, vertex i 's adjacent vertices are visited and not selected by JUMP such that vertex i is selected to dominate itself. For this case vertex i is called an f -vertex and we define $f(i) = i$.
- (4) There is a vertex j such that $j < i$, $i = j_{d_j}$, and all vertices $j, j_1, j_2, \dots, j_{d_j-1}$ are not in D_{i-1} , that is, vertex i is the last vertex that can dominate vertex j . For each such vertex j , we say that vertex j forces vertex i to be in D . For this case also vertex i is called an f -vertex, and we define $f(j) = i$.

Correctness. For each vertex i , $1 \leq i \leq n$, if it is put in D_i then it is dominated by itself. If it is not put in D_i then it is dominated either by a vertex $v \in D_{i-1}$ or by one of its neighbors i_p that is not visited yet, that is, $i_p > i$. Therefore, the final $D = D_n$ is a dominating set for the graph G .

Performance ratio. We now show that the performance ratio of JUMP is $1.5\sqrt{n} + c_1$ for some constant c_1 . Let C be a minimum dominating set for G .

Lemma 1. *There are at most $\lfloor \sqrt{n} \rfloor$ j -vertices in D .*

Proof. Since each j -vertex v in D dominates at least $\lceil \sqrt{n} \rceil$ “undominated” vertices at time v . Once a vertex is marked as “dominated”, it will no longer contribute as an “undominated” vertex in case one. Therefore, there are at most $\lfloor n / \lceil \sqrt{n} \rceil \rfloor = \lfloor \sqrt{n} \rfloor$ j -vertices in D . \square

Lemma 2. *There is at most one r -vertex in D .*

Proof. If vertex v is an r -vertex in D then all vertices of G are dominated after time v . No more vertices can be put into D thereafter. \square

Lemma 3. *If the minimum dominating set C consists of only one vertex then D contains no f -vertices.*

Proof. Let $C = \{c\}$. Vertex c is adjacent to all the other vertices. Therefore no f -vertices are put into D before time c . If vertex c is selected as a j -vertex in case one then all vertices are dominated by vertex c . No more vertices can be put into D . Otherwise, vertex c , or some other vertex c' that dominates all vertices, must be selected as an r -vertex. So, no f -vertices can be put into D . \square

We define opt as a mapping of a vertex into one of its dominating vertices in C :

$$opt(i) = \begin{cases} i & \text{if } i \in C, \\ \min\{j \mid j \in C, (i, j) \in E\} & \text{if } i \notin C. \end{cases}$$

We now compare the sets D and C . We define a charge scheme, which distributes weights w_1 of vertices in D to weights w_2 of vertices in C . Initially, $w_1(u) = 1$ if and only if $u \in D$ and $w_2(v) = 0$ if and only if $v \in C$. Then the performance ratio of JUMP is $\max\{w_2(v) \mid v \in C\}$ after the weight distribution. The charge scheme is as follows:

1. For each vertex $u \in C \cap D$, the weight of u is charged to itself. That is, $w_2(u)$ is increased by $w_1(u)$.
2. For each f -vertex $u \in D - C$, the weight of vertex u is evenly charged to each vertex $opt(v)$, where $f(v) = u$. That is, $w_2(opt(v))$ is increased by $w_1(u)/m$, where $m = |\{v \mid f(v) = u\}|$.
3. For each j - or r -vertex $u \in D - C$, the weight of vertex u is evenly charged to each vertex $v \in C$. That is, $w_2(v)$ is increased by $w_1(u)/|C|$.

Lemma 4. *For each $v \in C \cap D$, $w_2(v)$ is not charged by any other f -vertex in D .*

Proof. For an f -vertex $u \in D$ and $u \neq v$ charging to $w_2(v)$, vertex u must be in $D - C$. We consider only case four by which vertex u is put into D as an f -vertex. There must be a vertex v' with $opt(v') = v$ and vertex v' forces vertex u to be in D , that is, $f(v') = u$. However, if vertex v' forces vertex u to be in D , all its adjacent vertices must be visited and not in D_u . Therefore vertex v , which is adjacent to vertex v' , is not in D , which is a contradiction. \square

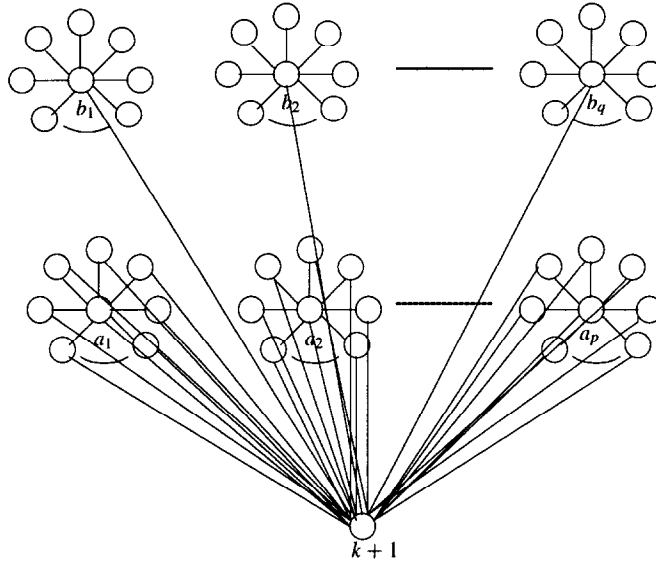


Fig. 1. The structure against on-line dominating set algorithms.

Lemma 5. For each vertex $v \in C - D$, $w_2(v)$ is charged by at most $\lceil \sqrt{n} \rceil - 1$ f -vertices.

Proof. Since vertex $v \notin D$, we let $I_v = \{u \mid (u, v) \in E, \text{ vertex } u \text{ is "undominated" after time } v - 1\}$. For an f -vertex u , $f(v') = u$ and $opt(v') = v$, to be charged to $w_2(v)$, vertex v' must be in I_v and f -vertex u is in $D - C$. Since vertex v is not put into D at time v , it has at most $\lceil \sqrt{n} \rceil - 1$ adjacent "undominated" vertices after time $v - 1$, that is, $|I_v| \leq \lceil \sqrt{n} \rceil - 1$. Therefore, $w_2(v)$ is charged at most $\lceil \sqrt{n} \rceil - 1$ times by f -vertices. \square

Lemma 6. If $|C| = 1$ then the performance ratio of JUMP is at most $\lceil \sqrt{n} \rceil + 1$.

Proof. By Lemma 3 the set D contains no f -vertices if $|C| = 1$. By Lemmas 1 and 2, $|D| \leq \lceil \sqrt{n} \rceil + 1$. The weight $w_2(c)$ is at most $\lceil \sqrt{n} \rceil + 1$, where $C = \{c\}$. Therefore, the performance ratio of JUMP is at most $\lceil \sqrt{n} \rceil + 1$ if $|C| = 1$. \square

Lemma 7. The performance ratio of JUMP is at most $1.5\sqrt{n} + c_1$ for some constant c_1 .

Proof. By the above lemma we consider only the case that $|C| \geq 2$. Since D consists of at most $\lceil \sqrt{n} \rceil$ j -

vertices and one r -vertex, for each vertex $v \in C$, $w_2(v)$ is charged at most $(\lfloor \sqrt{n} \rfloor + 1)/2$ by j - and r -vertices. Furthermore, by Lemmas 4 and 5, for each vertex $v \in C$, $w_2(v)$ is charged at most, for $n \geq 2$,

$$\lceil \sqrt{n} \rceil - 1 + (\lfloor \sqrt{n} \rfloor + 1)/2,$$

which is $1.5\sqrt{n} + c_1$ for some constant c_1 . \square

Lower bound. We now use a strong adversary to prove that $\sqrt{n} - c_2$ is a deterministic lower bound for the performance ratio of the on-line dominating set problem for general graphs in this setting, where c_2 is a constant.

Theorem 8. $\sqrt{n} - c_2$ is a lower bound for the performance ratio of the on-line dominating set problem for general graphs in this setting.

Proof. Let D be the dominating set selected by an on-line dominating set algorithm. For simplicity, we assume that $k = \sqrt{n} - 1$ is an integer. Let $S'_i = (V'_i, E'_i)$, where $V'_i = \{i, k + 1, i(k - 1) + 3, i(k - 1) + 4, \dots, i(k - 1) + k + 1\}$ and $E'_i = \{(i, v) \mid v \in V'_i - \{i\}\}$ for $1 \leq i \leq k$.

From time intervals 1 to k , vertices $1, 2, \dots, k$ are presented such that vertices a_1, a_2, \dots, a_p are included in D and vertices b_1, b_2, \dots, b_q are not

included in D by the on-line algorithm, where $\{a_1, a_2, \dots, a_p, b_1, b_2, \dots, b_q\} = \{1, 2, \dots, k\}$ and $p + q = k$. At time interval $k + 1$, vertex $k + 1$ is presented with adjacency to vertices 1 to k and to vertices in $V'_{a_1} \cup V'_{a_2} \cup \dots \cup V'_{a_p} - \{k + 1\}$. After time $k + 1$, all vertices in $(V'_{b_1} \cup V'_{b_2} \cup \dots \cup V'_{b_q}) - \{k + 1, b_1, b_2, \dots, b_q\}$ are forced to be included in D with a structure shown in Fig. 1. Therefore $D = \{a_1, a_2, \dots, a_p, k + 1\} \cup \{u \mid u \in (V'_{b_1} \cup V'_{b_2} \cup \dots \cup V'_{b_q}) - \{b_1, b_2, \dots, b_q\}\}$. Since $p \cdot k + q \cdot k + 1 = n$ and the minimum dominating set is $\{b_1, b_2, \dots, b_q, k + 1\}$, the lower bound for the performance ratio is

$$\begin{aligned} & \frac{q \cdot (k - 1) + p + 1}{1 + q} \\ &= \frac{k \cdot (1 + q) - 2q + 1}{1 + q} \\ &= \sqrt{n - 1} - \frac{2q - 1}{1 + q} \geq \sqrt{n} - c_2. \quad \square \end{aligned}$$

3. The second on-line setting

In this setting, at time interval i , the adjacency condition of vertex i to the vertices j , $1 \leq j < i$ is given. We show that $n - 1$ is the tight bound for this on-line setting.

First, we present an on-line algorithm for this setting for completeness although it is straightforward. At time i , the algorithm puts vertex i into D if vertex i is not dominated by other vertices j , $1 \leq j < i$. It is easy to check that the performance ratio of the algorithm is $n - 1$, in particular, for graphs with a single dominating vertex. We now show that the lower bound is $n - 1$.

Theorem 9. $n - 1$ is a deterministic lower bound for the performance ratio of the on-line dominating set problem in this setting.

Proof. An adaptive adversary is given as follows. Let A be an on-line algorithm. The adversary keeps providing a vertex that is connected to all previous vertices till it is not put into D . Let us say that this vertex is k . Then the remaining vertices j , $k + 1 \leq j \leq n$, are provided as being connected to vertex k only. Algorithm A has to choose the remained vertices so that D is $\{1, 2, \dots, k - 1, k + 1, k + 2, \dots, n\}$. However, the

minimum dominating set for this graph is $\{k\}$. Therefore, $n - 1$ is a deterministic lower bound. \square

From the above proof, we can see even if the given graph is known to be connected (or permutation, or interval) in advance, the lower bound remains the same.

Corollary 10. The deterministic lower bound for the performance ratio of the on-line dominating set problem for connected (even for permutation or interval) graphs in this setting is $n - 1$.

Corollary 11. $n - 1$ is the tight bound for the performance ratio of the on-line dominating set problem (even for connected, permutation, or interval graphs) in this setting.

References

- [1] M.J. Atallah, G.K. Manacher and J. Urrutia, Finding a minimum independent dominating set in a permutation graph, *Discrete Appl. Math.* **21** (1988) 177–183.
- [2] A. Bar-Noy, R. Motwani and J. Naor, The greedy algorithm is optimal for on-line edge coloring, *Inform. Process. Lett.* **44** (1992) 251–253.
- [3] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness* (Freeman, San Francisco, CA, 1979).
- [4] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs* (Academic Press, New York, 1980).
- [5] R.M. Karp, U.V. Vazirani and V.V. Vazirani, An optimal algorithm for on-line bipartite matching, in: *Proc. 22nd ACM Symp. on Theory of Computing* (1990) 352–358.
- [6] M.S. Manasse, L.A. McGeoch and D.D. Sleator, Competitive algorithms for server problems, *J. Algorithms* **11** (1990) 208–230.
- [7] A. Srinivasan and C.P. Rangan, Efficient algorithms for the minimum weighted dominating clique problem on permutation graphs, *Theoret. Comput. Sci.* **91** (1991) 1–21.