[Volume 1 issue 8 Sep 2013]
Page No.204-208 ISSN :2320-7167

INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH

Domination Based Algorithms to Strong Centers of A Graph

A. Anto Kinsley, S. Somasundaram

Associate Professor, Department of Mathematics, St. Xavier's College (Autonomous), Palayamkottai, Tirunelveli - 627002, Tamil Nadu, India. antokinsley@yahoo.com

Professor, Department of Mathematics, Manonmaniam Sundaranar University, Tirunelveli - 627012, Tamil Nadu, India. somutvl@gmail.com

Corresponding Author :- A. Anto Kinsley, M.Sc., M.Phil., M.Tech., Ph.D.

Associate Professor, Department of Mathematics, St. Xavier's College (Autonomous), Palayamkottai-627002, Tirunelveli, Tamil Nadu, India.

antokinsley@yahoo.com

Abstract

In this paper we consider a strong center location problem, which is based on the dominating set problem. The computation of the minimum dominating sets of a graph is used as a basic step for the determination of strong centers of the graph. The strong centers are developed from strong domination sets. We first study reachable sets, link vectors of the vertices and then introduce two binary operations $\vee$ and $\wedge$ for finding the dominating sets. These are used as tools for designing an algorithm to find the strong centers. We investigate the dominating sets of a graph. Some required results for developing algorithms are also proved. Using these concepts we present an algorithm to find all strong centers of a graph.

Keywords: Graph, algorithm, dominating set, central structures.

1. Introduction

Graph theory has immense applications in the field of communication networks, interpersonal relations and several real life situations. In several cases choosing a convenient vertex is always an interesting problem. Such convenient vertices may be *center*, *median* or *centroid* of the graph[5]. P. J. Slater (1975) defined these three central structures and he generalized the concept of centrality. These centrality concepts have applications in facility location problems such as installation of a fire station, hospital etc. In the recent years there is an increasing interest among researchers in the design and analysis of algorithms to solve graph-theoretic problems.

The concept of center of a graph has applications in facility location problem: given *n* cities and the distances between them, we wish to select cities as centers so that the maximum distance of a city from its closest center is minimized. In this paper we study *reachable sets, link vectors* of the vertices and introduce two *binary operations* $\vee$ *and* $\wedge$ [2]. We construct dominating sets of a graph and search for strong centers. The computation of the strong dominating set is used as a basic step in the algorithm for finding the strong centers.

In this paper, we consider a finite, connected, undirected simple graph with n vertices. Terms not defined here are used in the sense of Harary[3].

2. The strong dominating sets

In this section we first see the definition of dominating set and then we prove some important theorems.

Dominating Sets

Definition 2.1

Let $G = (V, E)$ be a graph. A subset S of V is called a *dominating* set if every vertex in $V - S$ is adjacent to at least one vertex in S. A dominating set is called *minimal* if there is no other dominating set, which is contained in it. A dominating set is called *minimum* if it has minimum cardinality among all dominating sets.

The *domination number* γ of G is defined to be the cardinality of a minimum dominating set in G. Note that, if S is a dominating set of G then $S \cup T$ is also a dominating set of G for every $T \subseteq V - S$. For any graph G, $\gamma \geq n / (\nabla+1)$, where ∇ is the maximum degree of a vertices of G[1].

The strong domination number, denoted $s(G)$, was defined by B.D. Acharya[1]. He characterized the graphs G for which $s(G) = \gamma(G)$ and estimated suitable bounds for $s(G)$.

Definition 2.1

The *strong domination number* of a graph G is the least integer $s(G)$ for which every set of $s(G)$ vertices of G is a dominating set of G.

Some required results for developing algorithms are proved. Using these concepts we present an algorithm for finding the strong domination number of an arbitrary graph. Finally we present an algorithm to find the strong domination number of a graph. In all algorithms, we consider a graph with the distance matrix.

The strong domination number $s(G)$ of a graph G is the least integer s for which every s-subset of V is dominating. Then we have an immediate result [1].

Theorem 2.2 (B.D. Acharya)[1]

For any graph G with minimum degree δ, we have $\gamma \leq s(G) \leq (n\text{-}\delta)$ and the bounds are best possible.

Theorem 2.3

LV of each vertex of $\bigcap_{i=1}^{n} R_\lambda(x_i)$ is full.

Proof

Consider the vertex set $\{x_1, x_2,\ldots, x_n\}$ of a graph G. Let $x \in \bigcap_{i=1}^{n} R_\lambda(x_i)$.

Then $x \in R_\lambda(x_i)$ for every vertex x_i. That is, x is reachable to $x_1, x_2,\ldots, x_n$.

That is, LV of x is (1, 1, 1,…, 1). Hence x' is full. □

We see a theorem, which follows from the above theorem.

Example

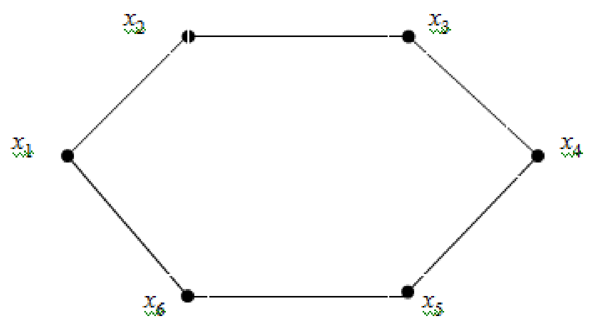

Figure 2.1 A graph illustrating Theorem 2.2

Consider $G = C_6$ where V = $\{x_1, x_2, x_3, x_4, x_5, x_6\}$.

The sets $\{x_1, x_4\}$, $\{x_2, x_5\}$, $\{x_3, x_6\}$ are minimum dominating sets and so $\gamma(G) = 2$.

But every subset of cardinality three is a dominating set and so s(G) = 3.

Also $(n\text{-}\delta) = 4$.

Theorem 2.4

Each vertex of $\bigcap_{i=1}^{n} R_1(x_i)$ dominates the graph G.

Proof: Let $x \in \bigcap_{i=1}^{n} R_1(x_i)$. Then x' is full by above theorem. Hence x is reachable to every vertex of

V. Since λ =1, x is adjacent to every vertex of V. Hence $\{x\}$ dominates G. □

We have presented all required results and then we concentrate in the development of algorithms.

Algorithms development

In this section we present three algorithms [6]. Now suppose that a proper subset D of the vertex set V of a graph G is given. We develop the following algorithm to check whether the set D is a dominating set.

Algorithm: 2.5 (Dominating set confirmation algorithm)

Input: A graph $G = (V, E)$ and a subset $D = \{x_1, x_2, \dots, x_k\}$ of vertices.

Output: D is a dominating set or not.

Step 1: For i: = 1 to k,
find the n-dimensional link vector LV of each vertex x_i ; say x_i'.

Step 2: Take $y' \leftarrow$ (o).

Step 3: For i: = 1 to k,
$y' := y' \vee x_i'$

Step 4: If y' is full then conclude that D is a dominating set.
Otherwise D is not a dominating set.

We prove the correctness and complexity of algorithm 2.5 in the following theorem.

Theorem 2.6

Algorithm 2.5 confirms a given proper subset D of vertices of a graph G, a dominating set or not. Also this algorithm requires O(kn) cost of time, where $k = |D|$ and $n = |V|$.

Proof

Let $D = \{x_1, x_2, \dots, x_k\}$ be a given proper subset of vertices. Step: 2 initialize y'as null for pair wise $\vee$ (cup) operation in step 3. Then $y' = \bigvee_{i=1}^{k} x_i'$ is found by step 4. If $\bigvee_{i=1}^{k} x_i'$ is full then by theorem 2.4, D is a dominating set of G. If $\bigvee_{i=1}^{k} x_i'$ is not full then by definition of link vector, at least one x_j ($k + 1 \le j \le n$) is not adjacent to any vertex of D. In this case, D is not a dominating set.

Regarding its complexity, the algorithm collects LVs x_i' ($1 \le i \le k$) from the distance matrix. By the for loop, step 3 requires $k\,n$ comparisons. Thus it requires O($k\,n$) cost of time[4].

□

Next we develop an algorithm to find all minimum-dominating sets of a graph G. In this algorithm, γ_0 denotes the integer closer to $n/(\Delta+1)$.

Algorithm 2.7
(Minimum-dominating set Algorithm)

Input: A graph $G = (V, E)$ with $V(G) = \{x_1, x_2, \dots, x_n\}$.

Output: D_j 's with γ vertices.

Step 1: For i: = 1 to n,
find the link vector LV of each vertex x_i ; say x_i'.

Step 2: Take $k \leftarrow \gamma_0$.

Step 3: Take all subsets D_j $\left(1 \le j \le \binom{n}{k}\right)$ of V with k vertices.

Step 4: Check whether any D_j is a dominating set, using algorithm 2.5
(Step 1 to step 4).

Step 5: If any D_j is a dominating set then find all dominating sets with k vertices, using
step 4 for all D_j .
5.1: Conclude that the above dominating sets D_j's are minimum. Stop.
Otherwise continue.

Step 6: Take $k = k+1$ and return to step 3.

We prove the correctness of algorithm 2.7 in the following theorem.

Theorem 2.8

Algorithm 2.7 correctly constructs all minimum-dominating sets of a graph G.

Proof

A graph $G = (V, E)$ is given with its distance matrix. By theorem 2.1, it is enough to investigate γ from the value γ_0, the closest integer to $n/(\Delta+1)$. Hence initialize $k = \gamma_0$ and collect the set F_k of all subsets of k vertices. Then the cardinality of F_k is $\binom{n}{k}$. Check whether any $D_j \in F_k$ is a dominating set. For this purpose we use algorithm 2.5. We have already proved that algorithm 2.5 confirms all dominating sets. If any

one of $D_j \in F_k$ is a dominating set then step 5 investigates all dominating sets of F_k. Then our process will be completed.
If no one of F_k is a dominating set then step 6 increases k by 1.
The process is repeated until we receive a dominating set. Since we start this algorithm with $k = \gamma_0$ we obtain all dominating sets with minimum cardinality.

By theorem 2.5, algorithm 2.7 correctly constructs all minimum-dominating sets of a graph G.
□

Finally we develop the main algorithm to find the strong domination number of a graph G.

Algorithm 2.9(Strong Domination Number Algorithm)
Input: A graph $G = (V, E)$ with $V(G) = \{x_1, x_2, \ldots, x_n\}$.
Output: k, the strong domination number.
Step 1: For i = 1 to n,
find the link vector x_i' of each vertex x_i ;
Step 2: For $k = \gamma$ to $(n\text{-}\delta\text{-}1)$
Repeat steps 2.1 to 2.2.
2.1: Take all subsets D_j 's $(1 \le j \le \binom{n}{k})$ of V with k vertices.
2.2: For $j = 1$ to $\binom{n}{k}$
Check whether D_j is a dominating set, using algorithm 2.5 (Step 1 to step 4).
If D_j is a dominating set, continue loop of step 2. 2
Otherwise continue loop of step 2
Step 3: Exit loop of step 2.
Step 4: Write the value of k.
End.

Theorem 2.10

Algorithm 2.9 correctly determines the strong domination number of a graph G.

Proof

By theorem 2.2, we can say that the value of k lies between γ and $(n - \delta)$. Using algorithm 2.5, Step 2.2 checks whether all D_j 's $(1 \le j \le \binom{n}{k})$ with k vertices are dominating sets. If all D_j 's are dominating sets (the required condition), then by definition the value of k is the strong domination number of the graph G. Otherwise, Step 2 to step 2.2 will be repeated by increasing the value of k by 1. This recursive call continues still the required condition is satisfied. If the required condition is not satisfied for all values of k between γ and $(n - \delta - 1)$ then we stop the investigation and conclude that the value of k is $(n - \delta)$. Hence algorithm 2.9 correctly determines the strong domination number of a graph G. □

Next we define a strong center and then we compute it using algorithms.

3. Strong centers

Definition 2.11

In a graph, a strong dominating set is a strong center.

Algorithm 2.12(Strong center Algorithm)
Input: A graph $G = (V, E)$ with $V(G) = \{x_1, x_2, \ldots, x_n\}$.
Output: The strong center.
Step 1: For $i = 1$ to n,
find the link vector x_i' of each vertex x_i;
Step 2: For $k = \gamma$ to $(n\text{-}\delta\text{-}1)$
Repeat steps 2.1 to 2.2.
2.1: Take all subsets D_j's $(1 \le j \le \binom{n}{k})$ of V with k vertices.
2.2: For $j = 1$ to $\binom{n}{k}$
Check whether D_j is a dominating set, using algorithm 2.5 (Step 1 to step 4).
If D_j is a dominating set,
then D_j is a strong k-center
continue loop of step 2. 2
Otherwise continue loop of step 2
Step 3: Exit loop of step 2.

Algorithm 2.12 correctly determines the strong domination number of a graph G, by theorem 2.10. The algorithm for finding strong centers is NP-complete[2].

Conclusion

We have designed algorithms to find the strong domination number of a graph G. We have defined a strong center of a graph and found by strong dominating set. The algorithm for finding strong center is NP-complete. A combinatorial algorithm can be developed for finding the strong center of a graph in future.

4. References

1. B. D. Acharya, H. B. Walikar and E. Sampathkumar**,** Recent developments in the theory of domination in graphs, Mehta Research Institute, Allahabad, MRI Lecture notes in Math.1(1979).

2. A. Anto Kinsley and S. Somasundaram, A domination based algorithm to k-center problem, Journal of Discrete Mathematical Sciences and Cryptography, Vol.9, No.3,(2006),403-416.

3. F. Buckley and F. Harary, Distance in Graphs, Addison-Wesley Publishing Company, New York, (1990).

4. N. Christofides, Graph Theory an algorithmic approach, Academic Press, London, (1975).

5. P. J. Slater, Maximin facility location, J. Res. Net Burstandards, 79B, (1975), 107- 115.

6. H. S. Wilf, Algorithms and Complexity, Prentice – Hall International, Inc., U. S. A. (1986).