AN EFFICIENT g-CENTROID LOCATION ALGORITHM FOR COGRAPHS

V. PRAKASH

Received 19 August 2004 and in revised form 17 January 2005

In 1998, Pandu Rangan et al. proved that locating the g-centroid for an arbitrary graph is $\mathcal{NP}$-hard by reducing the problem of finding the maximum clique size of a graph to the g-centroid location problem. They have also given an efficient polynomial time algorithm for locating the g-centroid for maximal outerplanar graphs, Ptolemaic graphs, and split graphs. In this paper, we present an $O(nm)$ time algorithm for locating the g-centroid for cographs, where n is the number of vertices and m is the number of edges of the graph.

1. Introduction

In this introductory section, we present some basic terminology of graph theory and a strong motivation for the study of g-centroid location problem.

A graph G consists of a finite nonempty set $V = V(G)$ of *vertices* together with a set $E = E(G)$ of unordered pairs of distinct vertices of V. The pair $e = \{u,v\}$ of vertices in E is called an *edge* of G. We also write an edge $e = \{u,v\}$ as $e = uv$.

If $e = uv \in E$, then u and v are called *adjacent* vertices and e is *incident* with each of its two vertices u and v.

The *degree* of a vertex u, denoted by $d(u)$, is the number of edges incident with it.

For $i \geq 1$, the ith *neighbourhood* of $u \in V(G)$ is defined as $N_i(u) = \{v \in V(G) \mid d(u,v) = i\}$. We call $N_1(u)$ by simply $N(u)$.

The *eccentricity* of a vertex u, denoted by $e(u)$, is defined as $e(u) = \max\{d(u,v) : v \in V(G)\}$, where $d(u,v)$ is the distance between u and v.

A graph is *self-centered* if the eccentricity of a vertex is the same as that of every other vertices. The length of any longest geodesic in G is called the *diameter* of G.

Let $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$ be two graphs. The *join* of G_1 and G_2, denoted by $G = G_1 \vee G_2$, has the vertex set $V = V_1 \cup V_2$ and the edge set $E = E_1 \cup E_2 \cup \{uv \mid u \in V_1,\ v \in V_2\}$.

For further common terms not explicitly mentioned here, reference may be made (with suitable changes) to Bondy and Murty [1], Buckley and Harary [2], or Parthasarathy [12].

International Journal of Mathematics and Mathematical Sciences 2005:9 (2005) 1405–1413
DOI: 10.1155/IJMMS.2005.1405

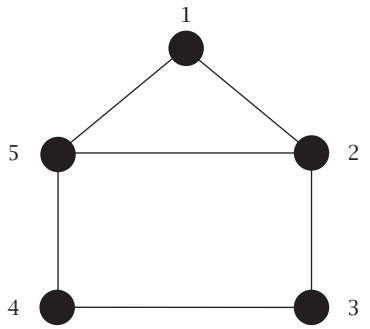

Figure 1.1

Several types of convexity in graphs have been studied in the past modeled on similar concepts in Euclidean space and topology. But the most important and the natural type of graph convexity is the *geodetic convexity* (g-convexity for short). This was introduced and studied by several researchers. See, for example, Mulder [9], Nieminen [10], Duchet [3], and Soltan and Chepoĭ [13].

We now give the definition of g-centroid through the g-convexity for graphs.

Definition 1.1. A set $S \subseteq V$ is *geodetic convex* (*g-convex* for short) if for every pair of vertices $u, v \in S$, all vertices on any $u - v$ shortest path (also called a *geodesic path*) belong to S.

From the above definition, it easily follows that a singleton set, vertex pair of an edge, and the whole vertex set $V(G)$ are g-convex sets of G. We call them as trivial g-convex sets. Also if S is a clique (S induces a complete subgraph of G), then S is a g-convex set of G.

Any communication network can be naturally modeled as a graph where the vertices represent either computers or intermediate message processors and the edges represent interconnections between them. In most of the communication networks, information between any two nodes tends to flow through any shortest path connecting them (e.g., paths with least hop count in case of distance vector protocols, and least weighted paths in case of link state routing protocols in the internetwork). Thus a g-convex set is basically a set of nodes that is *closed* with respect to the flow of information.

Definition 1.2. Let $G = (V, E)$ be any connected graph. For $v \in V$, the *g-weight* $w(v) = \max\{|S| : S \text{ is a } g\text{-convex set of } G \text{ not containing } v\}$. Let $gc(G) = \min\{w(v) : v \in V\}$. Then $gc(G)$ is called the *g-centroidal number* of G and the vertices v for which $w(v) = gc(G)$ are called the *g-centroidal* vertices. The *g-centroid* $C_g(G)$ is the set of all g-centroidal vertices of G (i.e., g-centroid is a set of vertices which satisfies the min-max relation).

For $v \in V(G)$, we denote by $S_v = S_v(G)$, any maximum g-convex set of G not containing v.

For the sake of clarity, we explain through an example.

Example 1.3. In Figure 1.1, $S_1 = \{2,3,4,5\}$; $w(1) = 4$, $S_2 = \{1,5,4\}$; $w(2) = 3$, $S_3 = \{1,5,4\}$ or $\{1,2,5\}$; $w(3) = 3$, $S_4 = \{1,2,3\}$ or $\{1,2,5\}$; $w(4) = 3$, and $S_5 = \{1,2,3\}$; $w(5) = 3$. Thus $C_g(G) = \{2,3,4,5\}$.

We now give a strong motivation for studying the g-centroid location problem.

In Mitchell [8], the g-centroid for trees plays the role of a "telephone center" through which maximum number of calls can pass at any given time for an optimal load balancing in a telephone network.

In [4], Gerstel and Zaks have discussed an application of g-centroid in distributed computing, in particular, to the message passing model of a distributed asynchronous networks. They have proved that given a network with a tree topology, choosing a g-centroidal vertex and then routing all the information through it is the best possible strategy in case of worst case complexity of any distributed sorting algorithm.

In [7], Kang and Ault gave some properties of the g-centroid for a tree and indicated some possible application in information retrieval.

However, in all the above three papers, the authors have considered only tree networks. These results can be extended for arbitrary graphs, which we present in our future papers.

Thus the problem of locating the g-centroid of a graph gained importance. In [11] Pandu Rangan et al. have proved that locating the g-centroid for an arbitrary graph is $\mathcal{NP}$-hard by reducing the problem of finding the maximum clique size of a graph to the problem of locating the g-centroid.

However, in practice, the underlaying topology of any application network will have some special graph properties like being acyclic or belong to a special classes of perfect graphs. Several $\mathcal{NP}$-hard or $\mathcal{NP}$-complete problems for arbitrary graphs have a nicer polynomial time algorithm on special classes of perfect graphs. For more detail on perfect graphs and the algorithmic aspects on perfect graphs, see Golumbic [5].

Among several classes of perfect graphs, chordal graphs, permutation graphs, and cographs are the most important ones due to their practical application. In this paper, we consider cographs and present a polynomial time algorithm for locating the g-centroid.

We now give the definitions of some special classes of perfect graphs considered in this paper. First we define the permutation graphs.

Let π be a permutation of the sequence $1,2,\ldots,n$. The graph $G(\pi) = (V,E)$ for the permutation π is defined as follows:

$V = \{1,2,\ldots,n\}$ and $ij \in E$ if and only if $(i-j)(\pi_i^{-1} - \pi_j^{-1}) < 0$, where π_i^{-1} is the pre-image of i under the permutation π.

An undirected graph G is called a *permutation graph* if there exists a permutation π such that G is isomorphic to $G(\pi)$ for some permutation π.

A graph G is a *distance hereditary graph* if for any two vertices u, v of G, the distance between u and v in G is the same as that of the distance between u and v in every connected induced subgraph containing both.

A graph is a *complement reducible graph* (popularly known as *cograph*) if it can be reduced to an empty graph by successively taking complements with in components (West [14]).

It can easily be verified that cographs are distance hereditary graphs and are properly contained in the class of permutation graphs.

2. Structural results for cographs

In this section, we present some structural results on cographs using some of the well-known earlier results. Using these results, we present an $O(nm)$ time algorithm for locating the g-centroid. Our structural characterizations can also used to provide polynomial time algorithm for other $\mathcal{NP}$-hard or $\mathcal{NP}$-complete problems on cographs.

Several characterizations of cographs are known. But the most important one is the following.

Proposition 2.1. *A graph G is a cograph if and only if it contains no P_4 (a path on 4 vertices) as an induced subgraph.*

As an immediate observation, we have the following results which specifies the maximum diameter and the properties of induced subgraphs of cographs.

Corollary 2.2. *Let G be a connected cograph. Then for any $u \in V$, the eccentricity $e(u) \leq 2$. That is, if a cograph is connected, then its diameter is at most two.*

Even though cographs can be disconnected, in this paper, we consider only connected cographs. Thus the graphs considered in this paper will have diameter at most two.

Proposition 2.3. *Any induced subgraph of a cograph is again a cograph.*

The following proposition is true for any arbitrary graph.

Proposition 2.4. *Let G be any connected graph and $u \in V(G)$. For any S_u, $S_u \cap N(u)$ is either empty or induces a complete subgraph of G. Further, if $e(u) = 1$, then S_u is a maximum clique of $G - \{u\}$.*

Proof. Let $x, y \in S_u \cap N(u)$. If x and y are nonadjacent, then x, u, y is a geodesic path joining x and y. Since S_u is g-convex, this requires $u \in S_u$, a contradiction. Thus $xy \in E$ and hence $S_u \cap N(u)$ induces a complete subgraph of G.

The second part of the proposition follows from the maximality of S_u. □

We now discuss about the g-centroid of G when G has a vertex of eccentricity one.

Proposition 2.5. *Let G be a cograph. Let $U = \{u_1, u_2, \ldots, u_r\}$ be the set of all vertices of G with eccentricity one (and therefore degree $n-1$). If $G' = G - U$ is disconnected, then $C_g(G) = U$.*

Proof. Let $M_1, M_2, \ldots, M_s$ be the maximum cliques of G' of size $\omega(G') = \omega'$ and $M = \cap_{i=1}^{s} M_i$. From Proposition 2.4, it follows that $w(u_i) = \omega' + r - 1$. We split our discussion into two cases depending upon whether M is empty or not.

Case 1 (M is empty). In this case for each vertex x of G', we can find an i, $1 \leq i \leq s$, such that $x \notin M_i$. Thus $U \cup M_i$ is a g-convex of G (as it is a clique) not containing x with cardinality $r + \omega'$. Therefore, $w(x) > w(u_i)$. By the arbitrariness of x, it follows that $C_g(G) = U$.

Case 2 (M is nonempty). Let $C_1, C_2, \ldots, C_t$ be the components of G'. Since G' is disconnected and M is a clique, M is contained in a component, say C_i of G'. Then for each x

not in M, as before $w(x) > w(u_i)$. If $x \in M$, then $S = (M_k - x) \cup U \cup_{j \neq i} C_j$ is a convex set not containing x with $|S| > w(u_i)$. Therefore, $C_g(G) = U$. □

Next we discuss the case when G' is connected. Before discussing about the structure of $C_g(G)$, we present some results on the structure of G' when G' is connected.

Proposition 2.6. *Let G be a cograph and G' be defined as in the previous proposition. If G' is connected, then G' is a self-centered cograph of diameter two.*

Proof. From Proposition 2.3, G' is again a cograph. Since G' is connected, by Corollary 2.2, $e_{G'}(u) \leq 2$ for every vertex $u \in V(G')$. We claim that G' has no vertex of eccentricity one. On the contrary, if u is a vertex of eccentricity one in G', then u is adjacent to every vertex of G'. Since u_i's are adjacent to every vertex of G, it follows that u is adjacent to every vertex of G. Thus $e_G(u) = 1$. This contradicts the definition of G'; hence $e_{G'}(u) = 2$ for every $u \in G'$. □

We now analyze the structure of self-centered cographs.

Proposition 2.7. *Let G be a self-centered cograph of diameter two and let u be a vertex of G. Then the following hold.*

(1) *If either $\langle N(u)\rangle$ is disconnected and $\langle N_2(u)\rangle$ is connected or if both are disconnected, then for every $x \in N(u)$ and $y \in N_2(u)$, $xy \in E$. That is, $G - u = \langle N(u)\rangle \vee \langle N_2(u)\rangle$.*

(2) *If both $\langle N(u)\rangle$ and $\langle N_2(u)\rangle$ are connected, then $G - u = \langle A\rangle \vee \langle (N_2(u) \cup B)\rangle$, where A is the set of vertices of $N(u)$ having a descendant in $N_2(u)$ and $B = N(u) - A$. Also $\langle A\rangle$ is an incomplete graph.*

(3) *If $\langle N(u)\rangle$ is connected and $\langle N_2(u)\rangle$ is disconnected, then there exists a pair of non adjacent vertices x_1, x_2 of $N(u)$ such that x_1 and x_2 are adjacent to every vertex of $N_2(u)$.*

Proof. (1) Let $\langle N(u)\rangle$ be disconnected and $\langle N_2(u)\rangle$ be connected. Let $x \in N(u)$ and $y \in N_2(u)$. Let C_1 be the component of $\langle N(u)\rangle$ containing x. Let x_1 be any parent of y in $N(u)$. If x_1 belongs to a component C_2 different from C_1, then by considering the path y, x_1, u, x, we have $xy \in E$ (as G has no induced P_4). Suppose $x_1 \in C_1$. Let C_2 be any other component of $\langle N(u)\rangle$ and $x_2 \in C_2$. Consider the path y, x_1, u, x_2. Since this is not an induced path, $yx_2 \in E$. But, then, as before $xy \in E$.

In the above proof, we have not used the fact that $\langle N_2(u)\rangle$ is connected. Therefore, even if $\langle N_2(u)\rangle$ is disconnected, $xy \in E$.

(2) Let A be the set of all vertices in $N(u)$ having a descendant in $N_2(u)$.

We prove this case through the following three claims:

(1) every vertex of A is adjacent to every other vertex of $N_2(u)$;

(2) every vertex of A is adjacent to every other vertex of $N(u) - A$;

(3) A induces an incomplete graph, that is, A has a pair of nonadjacent vertices.

Claim 2.8. Each vertex of A is adjacent to every vertex of $N_2(u)$.

Let $x \in A$. Let z be any arbitrary vertex of $N_2(u)$. If $xz \in E$, then we are through. If $xz \notin E$, we establish a contradiction. Let y be a descendant of x in $N_2(u)$. Let $y = y_0, y_1, y_2, \ldots, y_r = z$ be a path connecting y and z in $N_2(u)$ (see Figure 2.1). Such a path

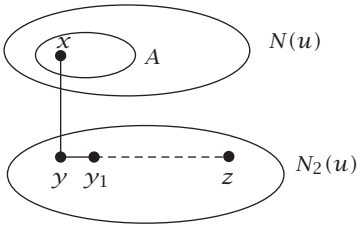

Figure 2.1

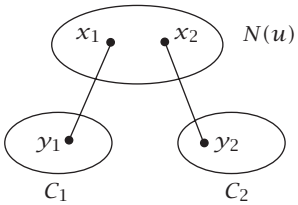

Figure 2.2

exists as $\langle N_2(u)\rangle$ is connected. Now u, x, y, y_1 is a path on 4 vertices implying $xy_1 \in E$. Extending the argument, we can show that $xz \in E$.

Claim 2.9. Let $B = N(u) - A$. Then for every $x \in A$ and $x_1 \in B$, $xx_1 \in E$.

This follows by considering a vertex y in $N_2(u)$ and the path x_1, u, x, y.

Claim 2.10. A induces an incomplete graph.

Assume the contrary. Let $x \in A$. By Claims 2.8 and 2.9, x is adjacent to every vertex of B as well as $N_2(u)$. Therefore, the degree of x, $d(x) = n - 1$ and hence the eccentricity $e(x) = 1$. This contradicts our assumption that G is a self centered cograph of diameter two. Thus $\langle A\rangle$ is an incomplete graph.

(3) Let $\langle N(u)\rangle$ be connected and $\langle N_2(u)\rangle$ be disconnected.

First we show that there exists a vertex x_1 of $N(u)$ adjacent to every vertex of $N_2(u)$. Let C_1 and C_2 be any two components of $\langle N_2(u)\rangle$ and let $y_i \in C_i$, $i = 1,2$. Let x_i be an ancestor of y_i in $N(u)$ (see Figure 2.2). As by the arguments in Claim 2.8, x_1 is adjacent to every vertex of C_1 and x_2 is adjacent to every vertex of C_2. If x_1 and x_2 are different, then consider the path y_1, x_1, u, x_2. Since G is a cograph, this path must have a chord. The possible chords are x_1x_2 and y_1x_2. If $y_1x_2 \in E$, then x_2 is adjacent to every vertex of C_1 and C_2. If $x_1x_2 \in E$, then by considering the path y_1, x_1, x_2, y_2, we can show that either x_1 or x_2 is adjacent to every vertex of C_1 and C_2. In a similar fashion, we can extend this to show that $N(u)$ contains a vertex x_1 adjacent to every vertex of $N_2(u)$.

Now $e(x_1) = 2$. Let x_2 be an eccentric vertex for x_1. Since x_1 is adjacent to every vertex of $N_2(u)$ and to u, $x_2 \in N(u)$. We now show that x_2 is also adjacent to every vertex of $N_2(u)$. Let $y \in N_2(u)$. By considering the path y, x_1, u, x_2, we have $yx_2 \in E$. □

Next we analyze the structure of S_u when G' is connected.

Proposition 2.11. *Let G be a cograph. If $G' = G - U$ is connected, where U is a set of vertices of G with eccentricity one, then for every vertex u of G, S_u induces a complete subgraph of G, and hence is a maximum clique of G not containing u.*

Proof. Let $u \in V(G)$. If $u \in U$, then by Proposition 2.4, S_u induces a complete subgraph of G.

Suppose $u \notin U$. Consider an S_u. If possible, contrary to our assumption, let x, y be a pair of nonadjacent vertices in S_u. From the remark succeeding Proposition 2.4, both x and y cannot belong to $N(u)$ as $\langle N(u) \cap S_u \rangle$ is complete. Thus, either both x and y are in $N_2(u)$, or one is in $N(u)$ and the other is in $N_2(u)$. We split our discussion into four cases and in every case, we deduce the contradiction that $u \in S_u$.

Case 1. $\langle N(u) \rangle$ is disconnected and $\langle N_2(u) \rangle$ is connected.

Then by (1) or Proposition 2.7, x, y belong to $\langle N_2(u) \rangle$. Let C_1 and C_2 be any two components of $\langle N(u) \rangle$. Let $x_i \in C_i$ for $i = 1,2$. From (1) of Proposition 2.7, xx_i and $yx_i \in E$ for $i = 1,2$. Thus x, x_i, y is an $x - y$ geodesic path for $i = 1,2$. Hence, $x_i \in S_u$ and subsequently $u \in S_u$ (since x_i's are nonadjacent and x_1, u, x_2 is a geodesic path). This is a contradiction to the definition of S_u.

Case 2. Both $\langle N(u) \rangle$ and $\langle N_2(u) \rangle$ are disconnected.

Then in a similar fashion we can show that $u \in S_u$.

Case 3. Both $\langle N(u) \rangle$ and $\langle N_2(u) \rangle$ are connected.

Suppose that $x \in N(u)$ and $y \in N_2(u)$. Then by (2) of Proposition 2.7, $x \in B$. Since A is incomplete, A has a pair of nonadjacent vertices, say x_1 and x_2. By Proposition 2.7, $xx_i, yx_i \in E$ for $i = 1,2$. Then as before $u \in S_u$, a contradiction.

Similarly if $x, y \in N_2(u)$, then also $x_i \in S_u$ for $i = 1,2$ and hence $u \in S_u$.

Case 4. $\langle N(u) \rangle$ is connected and $\langle N_2(u) \rangle$ is disconnected.

From (3) of Proposition 2.7, $N(u)$ has a pair x_1, x_2 of nonadjacent vertices adjacent to every vertex of $N_2(u)$. If $x, y \in N_2(u)$, then as before $u \in S_u$, which is a contradiction. If $x \in N(u)$ and $y \in N_2(u)$ and if x is not adjacent to x_1, then by considering the path y, x_1, u, x, we can show that $xy \in E$, which is a contradiction. Therefore, in this case, x is adjacent to both x_1 and x_2. Consequently, x, x_i, y for $i = 1,2$ are geodesics joining x and y. Therefore, $x_1, x_2 \in S_u$ and hence $u \in S_u$. □

The above proposition is true even if U is empty (i.e., when $G = G'$).

Combining all the above propositions, we have the following theorem.

Theorem 2.12. *Let G be a cograph. Let U be the set of all vertices of G with eccentricity one. If $G' = G - U$ is disconnected, then $C_g(G) = U$, otherwise $C_g(G) = U \cup M$ or $C_g(G) = V(G)$, depending upon whether the intersection M of all maximum cliques of G is nonempty or not. In particular, if G' is connected, then for each vertex u of G, $w(u) = \omega(G - u)$.*

We now present our algorithm (Algorithm 2.1) to locate the g-centroid for cographs.

```
Procedure GC_COGP(G)
    begin
        Find the set U
        Let G' = G − U
        If G' is disconnected, then output U as C_g(G) else
        begin
            for each vertex x ∈ V(G) do
                w(x) = ω(G − x)
        end;
        output the least weighted vertices;
    end;
```

Algorithm 2.1

Complexity analysis. Vertices in G with eccentricity one are precisely those vertices with degree $n-1$. Hence, U can be found in $O(n+m)$ time. If U is nonempty, then G' can be constructed in $O(n+m)$ time. To check whether G' is connected or not takes $O(n+m)$ time. This is done by executing a depth-first search traversal. If G' is connected, then for each vertex x of G, $G-\{x\}$ is again a cograph. Note that cographs form a proper subclass of permutation graphs. An $O(n+m)$ time algorithm to find the maximum clique size for permutation graph is given by Kamakoti and Pandu Rangan [6]. Using this if G' is connected, then we can find the weight of all the vertices in $O(nm)$ time. Outputting the least weighted vertices takes $O(n)$ time. Thus our algorithm takes $O(nm)$ time to output the g-centroid for cographs.

The proof of correctness of our algorithm follows from Theorem 2.12.

Theorem 2.13. *Let G be a cograph. Then the g-centroid can be located in $O(nm)$ time.*

Remark 2.14. Note that if G' is connected, then $C_g(G)$ is the intersection of the maximum cliques of G. Thus if all the maximum cliques of a cograph can found in $O(n+m)$ time, as for chordal graphs, our algorithm can be slightly modified to output $C_g(G)$ in $O(n+m)$ time.

Acknowledgments

The author is thankful to Professor K. R. Parthasarathy, Department of Mathematics, and Professor C. Pandu Rangan, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, for their encouragement and helpful discussions during the preparation of this paper.

References

[1] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, American Elsevier Publishing, New York, 1976.

[2] F. Buckley and F. Harary, *Distance in Graphs*, Addison-Wesley Publishing Company Advanced Book Program, California, 1990.

[3] P. Duchet, *Convexity in combinatorial structures*, Rend. Circ. Mat. Palermo (2) Suppl. (1987), no. 14, 261–293.

[4] O. Gerstel and S. Zaks, *A new characterization of tree medians with applications to distributed sorting*, Networks **24** (1994), no. 1, 23–29.

[5] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Computer Science and Applied Mathematics, Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1980.

[6] V. Kamakoti and C. Pandu Rangan, *Efficient transitive reduction of permutation graphs and its applications*, Journal of Computer Science and Informatics **23** (1993), no. 3, 52–59.

[7] A. N. C. Kang and D. A. Ault, *Some properties of a centroid of a free tree*, Information Processing Lett. **4** (1975/76), no. 1, 18–20.

[8] S. L. Mitchell, *Another characterization of the centroid of a tree*, Discrete Math. **24** (1978), no. 3, 277–280.

[9] H. M. Mulder, *The Interval Function of a Graph*, Mathematical Centre Tracts, vol. 132, Mathematisch Centrum, Amsterdam, 1980.

[10] J. Nieminen, *Characterizations of graphs by convex hulls*, Bull. Inst. Math. Acad. Sinica **10** (1982), no. 2, 177–184.

[11] C. Pandu Rangan, K. R. Parthasarathy, and V. Prakash, *On the g-centroidal problem in special classes of perfect graphs*, Ars Combin. **50** (1998), 267–278.

[12] K. R. Parthasarathy, *Basic Graph Theory*, Tata McGraw-Hill, New Delhi, 1994.

[13] V. P. Soltan and V. D. Chepoĭ, *d-convex sets in triangulated graphs*, Mat. Issled. (1984), no. 78, 105–124 (Russian).

[14] D. B. West, *Introduction to Graph Theory*, Prentice Hall, New Jersey, 1996.

V. Prakash: Department of Computer Science and Computer Engineering, La Trobe University, Bundoora, Melbourne, VIC 3086, Australia

E-mail address: prakash@cs.latrobe.edu.au