Galaxy Cutsets in Graphs

NICOLAS SONNERAT* and ADRIAN VETTA†

Abstract. Given a network $G = (V, E)$, we say that a subset of vertices $S \subseteq V$ has radius r if it is spanned by a tree of depth at most r. We are interested in determining whether G has a cutset that can be written as the union of k sets of radius r. This generalizes the notion of k-vertex connectivity, since in the special case $r = 0$, a set spanned by a tree of depth at most r is a single vertex.

Our motivation for considering this problem is that it constitutes a simple model for virus-like malicious attacks on G: An attack occurs at a subset of k vertices and begins to spread through the network. Any vertex within distance r of one of the initially attacked vertices may become infected. Thus an attack corresponds to a subset of vertices that is spanned by k trees of depth at most r. The question we focus on is whether a given network has a cutset of this particular form.

The main results of this paper are the following. If $r = 1$, an attack corresponds to a subset of vertices which is the union of at most k stars. We call such a set a *galaxy* of order k. We show that it is NP-hard to determine whether a given network contains a cutset which is a galaxy of order k, if k is part of the input. This is in stark contrast to the case $r = 0$, since testing whether a graph is k-vertex connected can be done in polynomial time, using standard maxflow-mincut type results.

On the positive side, testing whether a graph can be disconnected by a single attack (i.e. $k = 1$) can be done efficiently for any r. Such an attack corresponds to a single set of vertices spanned by a tree of depth at most r. We present an $O(rnm)$ algorithm that determines if a given network contains such a set as a cutset.

Keywords: graph connectivity, star-cutsets, complexity

1. INTRODUCTION

Graph connectivity is a fundamental concept in network design and has been extremely well studied from a complexity viewpoint. The motivation for desiring highly connected networks is to provide resilience against network failures at vertices and/or edges. These failures may arise due to equipment malfunctions or due to malicious attacks on the network. Here we generalize the form in which network failures may occur: Instead of just

*Department of Mathematics and Statistics, McGill University. Email: sonnerat@math.mcgill.ca

†Department of Mathematics and Statistics, and School of Computer Science, McGill University. Email: vetta@math.mcgill.ca

affecting isolated vertices/edges, we allow them to spread through the network. This models a variety of scenarios, e.g. the spread of a virus through a social network; subversion in a spy network, where subverting some agents compromises the reliability of the spies with whom the subverted agents had been communicating; the spread of a fire through a neighbourhood, where the fire can jump from a building to other buildings within a certain distance; Denial of Service (DoS) attacks, where a (malicious) server attempts to disrupt servers it is connected to by sending a high volume of requests.

In addition to being useful for designing highly connected networks that are resilient to attacks, the notions of galaxy cutsets and cutsets spanned by trees of depth r are natural extensions of the usual vertex- and edge-connectivity of graphs and of star-cutsets, and therefore interesting from a purely graph theoretic point of view. Moreover, star-cutsets play an important role in the theory of perfect graphs and as such have received a great deal of attention in the literature.

Throughout this paper, $G = (V, E)$ will always denote a simple, undirected graph. A subset $X \subseteq V$ of vertices is called a cutset if the graph induced on $V - X$ is disconnected. We will abuse notation slightly and say that $G - X$ is disconnected. We are interested in cutsets that have a particular form, namely cutsets X that can be written as $X = \cup_{i=1}^{k} S_i$, where each S_i is a set spanned by a tree of depth at most r. Recalling our motivating examples, the roots of the trees spanning the sets S_i can be thought of as being infected by a virus, which can then spread to the other vertices of the sets S_i. The question we focus on in this paper is whether we can determine efficiently if a given graph is vulnerable to such a virus attack. This corresponds to deciding whether or not a given graph has a cutset which is spanned by k trees of depth at most r.

Note that our model does not require that *every* vertex within distance at most r of an initially infected vertex v become infected. However, for a vertex u to become infected, there must be a path of length at most r from v to u consisting entirely of infected vertices. Specifying the model like this allows for greater generality. In terms of the motivating examples, it is conceivable that a malicious attacker has control over the direction into which the virus spreads, or that there is an element of chance involved, e.g. some vertices could be less vulnerable to infection than others.

In Section 2 we show that if k is part of the input, it is NP-hard to determine whether a graph G contains a cutset X which is the union of k sets spanned by trees of depth at most $r = 1$. This is in stark contrast to the case $r = 0$, since testing whether a given graph is k-vertex connected can be achieved in polynomial time if k is part of the input, using standard results on network flows. The hardness proof proceeds by reducing the Vertex Cover problem to our problem.

In Section 3, we give a positive result, showing that we can determine efficiently whether G contains a cutset X spanned by a single tree of depth r. Our algorithm has run-time $O(rnm)$, which is a significant improvement over a straightforward $O(n^3m)$ algorithm.

In Section 4 we place our work into context by discussing related results, and also present some open problems.

2. A Hardness Result

Given a graph $G = (V, E)$, a *star* is a vertex v together with a subset of its neighbours. We emphasize that we do not require the star to contain all the neighbours of v. Any subset, including the empty set (in which case the star is just $\{v\}$), is permitted. We shall call a set X which can be written as the union of at most k stars a *galaxy of order* k. Our main result is the following theorem.

Theorem 2.1. *Determining whether a graph G contains a cutset which is a galaxy of order k is NP-hard if k is part of the input.*

We prove this result by reducing the Vertex Cover problem to the problem in the theorem. So suppose we are given an integer k and a graph H with vertices $\{v_1, \ldots, v_n\}$ and edges $\{e_1, \ldots, e_m\}$. We will construct an auxiliary graph G such that H has a vertex cover of size k if and only if G has a cutset S which is a galaxy of order k. Note that we may assume that $m > k$, as otherwise picking one endpoint of each edge yields a vertex cover of size at most k. If $k \leq 2$ or $k \geq n - 2$, we can find a vertex cover of size at most k in polynomial time, so we shall also assume that $2 < k < n - 2$.

In order to understand the construction of the reduction graph G better, it is helpful to observe two things about graphs without cutsets that are galaxies of order k. Firstly, every vertex must have degree at least $k + 1$, for otherwise we can disconnect that vertex by taking S to consist of all its neighbours. Secondly, if for some vertex v there are vertices whose neighbourhoods have large intersections with the neighbourhood of v, then it is intuitively more likely that there will be a galaxy cutset of order k disconnecting v from the rest of the graph. Our reduction graph consists, roughly speaking, of two trees whose leaves are connected in a specific way. The reason why trees are helpful is that to disconnect two vertices s_1 and s_2 of a tree, it doesn't really make a difference whether we remove a star intersecting the path from s_1 to s_2 or just a vertex on that path. The tree-like structure and the way the leaves of the trees are joined by edges will allow us to control how the neighbourhoods of different vertices intersect and as a consequence to place restrictions on which sets of k vertices can possibly form galaxy cutsets.

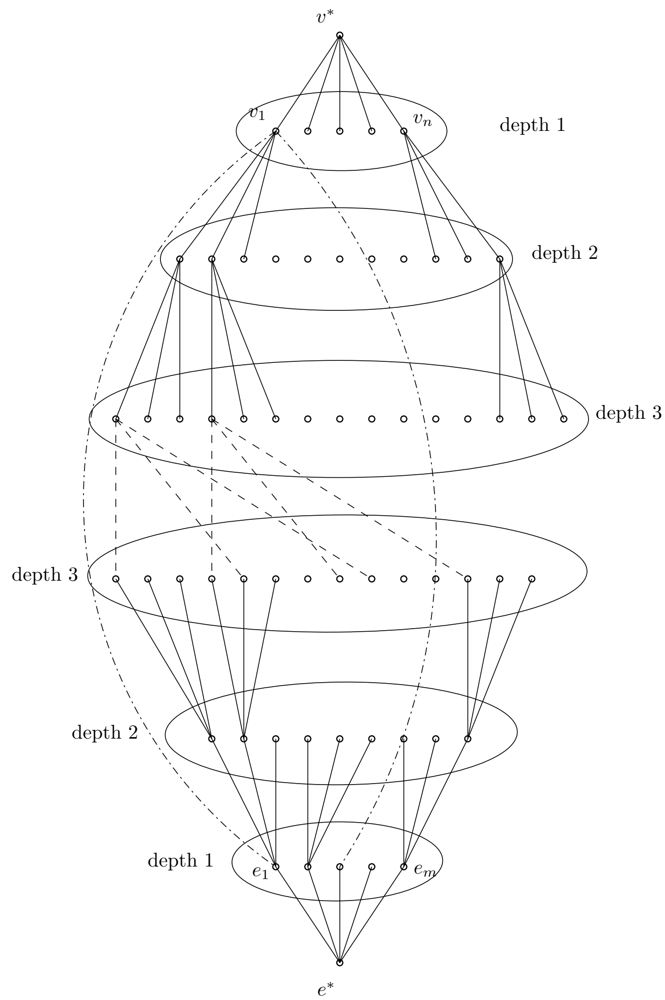

FIGURE 1. The reduction graph G

Figure 1 shows the high-level structure of G. There are two vertices v^* and e^*. These will be the roots of trees of depth 3. We will connect these trees by adding edges from leaves of the first tree to leaves of the second tree in a carefully chosen way. The vertex e^* has m children $\{t_1, \ldots, t_m\}$ (corresponding to the edges of H), and the vertex v^* has n children $\{s_1, \ldots, s_n\}$ (corresponding to the vertices of H). If $n \leq 2k$, we add $2k - n + 1$ dummy vertices $\{s_{n+1}, \ldots, s_{2k+1}\}$ as additional children of v^* and set $n' := 2k + 1$ (otherwise $n' = n$). We call these vertices of *depth* 1. Next, each vertex s_i has a_i children, and each vertex t_j has b_j children, where the a_i and b_j are chosen as follows. Pick a prime $p \geq \max(n'+1, m+1, 2k+4)$. Such a prime can be found in polynomial time, since there is always a prime between $N := \max(n' + 1, m + 1, 2k + 4)$ and $2N$, and PRIMES is in P ([1]). It will become clear in the proof of Claim 2.5 why we choose p to be a prime number. Now let each a_i equal either $\lfloor \frac{p^2}{n'} \rfloor$ or $\lceil \frac{p^2}{n'} \rceil$ and let each b_j equal either $\lfloor \frac{p^2}{m} \rfloor$ or $\lceil \frac{p^2}{m} \rceil$, so that they satisfy $\sum_i a_i = \sum_j b_j = p^2$. Note that we will have $a_i \geq k+1$ for all i and $b_j \geq k+1$ for all j. We call the children of the s_i grandchildren of v^*, and the children of the t_j grandchildren of e^*. Finally, for all i each child of s_i has $k+1$ children. These children are numbered and said to have *type* $1, 2, \ldots, k+1$ according to their number. Similarly, for all j, each child of t_j has $k+1$ children numbered 1 to $k+1$. The grandchildren of v^* and e^* are said to have depth 2, and the great-grandchildren are said to have depth 3.

To complete G, we add a few more edges. If v_i is an endpoint of e_j in H, we add the edge $s_i t_j$ to G. (We do not add edges from the dummy vertices to children of e^*.) The great-grandchildren of v^* are connected to the great-grandchildren of e^* as follows. Consider the great-grandchildren of v^* of type 1. There are p^2 of them, and we label them using the elements of the Abelian group $\mathbb{Z}_p \times \mathbb{Z}_p$ in lexicographic order, i.e.

$$(0,0), (0,1), \ldots, (0, p-1), (1,0), \ldots, (p-1, p-1).$$

The great-grandchildren of e^* of type 1 are labeled in the same way. For a vertex (i,j), we add edges to the great-grandchildren of e^* of type 1 which are labeled $(i+x, j+x^2)$ for $x = 0, \ldots, p-1$, where the labels are all taken modulo p. We do the same thing for the great-grandchildren that have type $2, 3, \ldots, k+1$. This concludes our description of the graph G.

The hardness of our decision problem is established by the following theorem.

Theorem 2.2. *The graph H has a vertex cover of size at most k if and only if G has a cutset S which is a galaxy of order k.*

Proof. One direction is easy. Suppose H has a vertex cover C of size at most k. Without loss of generality, assume $C = \{v_1, v_2, \ldots, v_l\}$ with $l \leq k$. Let S be the set consisting of $\{s_1, \ldots, s_l\}$ together with their neighbours among the children of e^*. Then S is the union of at most k stars. Since C is a vertex cover, S contains all the children of e^*, and so e^* and v^* are in different components of $G - S$.

For the other direction, we need two lemmas.

Lemma 2.3. *Let S be a galaxy of order k in G. Then the following hold:*

(1) *If $v^* \notin S$, then v^* has a neighbour of depth 1 in $G - S$.*
(2) *Any vertex of depth 1 in $G - S$ has a neighbour of depth 2 in $G - S$.*
(3) *If $e^* \notin S$, then e^* has a neighbour of depth 1 in $G - S$ unless H has a vertex cover of size at most k.*
(4) *Any vertex of depth 2 in $G - S$ has a neighbour of depth 3 in $G - S$.*

Lemma 2.4. *Let S be a galaxy of order k in G. Then the following hold:*

(1) *There exists an i such that any vertex of depth 3 and type i in $G-S$ is connected to every other vertex of type i in $G-S$.*
(2) *Any vertex of depth 3 and type i in $G-S$ is connected to a vertex of depth 3 and type j for all $j \neq i$ in $G-S$.*

These two lemmas together imply the theorem. If H has no vertex cover of size at most k, any vertex not in S is connected to some vertex of depth 3 by Lemma 2.3. But then Lemma 2.4 implies that all the vertices of depth 3 not in S are in the same component of $G-S$. So $G-S$ must be connected. □

The remainder of this section is devoted to proving Lemmas 2.3 and 2.4.

Proof of Lemma 2.3.

(1) Suppose $v^* \notin S$. A star centred at a grandchild of v^* contains at most one of the vertices s_i, namely its parent, and a star centred at a vertex t_j contains at most two vertices s_i, s_l. A star centred at a vertex s_i contains only one child of v^*, namely itself. If a star is centred at a vertex that is neither v^*, a child of v^*, a grandchild of v^*, nor a child of e^*, it cannot contain any of the vertices s_i. So S contains at most $2k$ of the children of v^*. Since we added dummy vertices $\{s_{n+1}, \ldots, s_{2k+1}\}$ in the case $n \leq 2k$, it follows that v^* will have a neighbour of depth 1 in $G-S$.
(2) Now take a vertex w of depth 1 in $G-S$. By construction, any star (except of course one centred at w) contains at most one of the children of w. But w has at least $k+1$ children, so there must be one in $G-S$.
(3) A star centred at a grandchild of e^* can contain at most one child of e^*, namely its parent. A star centred at a vertex s_i can contain at most $d(v_i)$ children of e^*, where $d(v_i)$ is the degree of v_i in H. A star centred at a vertex t_j contains only one child of e^*, namely itself. If a star is centred at a vertex which is neither e^*, a child of e^*, a grandchild of e^* nor a child of v^*, then it cannot contain any of the vertices t_j. So S contains at most M of the vertices t_j, where M is the maximum number of edges covered by a vertex cover of size at most k in H. It follows that if $M < m$, i.e. if H has no vertex cover of cardinality at most k, then e^* will have a neighbour of depth 1 in $G-S$.
(4) For the last case, take a vertex $w \in G-S$ of depth 2, and suppose w is a grandchild of v^*. By construction, w has $k+1$ children of depth 3. We claim that at most k of them can be contained in S. To see this, observe that a star that contains any of the children of w has to be centred at a vertex u of depth 3 (or at w, but we are assuming that $w \notin S$). If u is a great-grandchild of v^*, it has to be a child of w, and then the star contains no other children of w other than

u itself. If u is a great-grandchild of e^*, say of type i, then the star can only contain a child of w that is also of type i, and there is only one such child. Hence S contains at most k of the children of w, so w will have a neighbour of depth 3. A similar argument shows that a vertex of depth 2 which is a grandchild of e^* will have a neighbour of depth 3 in $G - S$.

□

The proof of Lemma 2.4 relies on the following claim.

Claim 2.5. *For each l, let F_l be the bipartite subgraph of G induced by the vertices of depth 3 and type l. Then F_l has the following properties:*

(a) *F_l is p-regular*
(b) *If u, w are in the same stable set of F_l, then $|\Gamma(u) \cap \Gamma(w)| \leq 1$* [1].
(c) *F_l is $(k+1)$-vertex connected.*

Proof. Fix l, and let X be the stable set of F_l whose vertices are the great-grandchildren of v^*, and let Y be the stable set of F_l whose vertices are the great-grandchildren of e^*.

(a) A vertex (i, j) in X has an edge to $(i + x, j + x^2)$ for $x = 0, 1, \ldots, p - 1$, and a vertex (i', j') in Y has an edge to $(i' - x, j' - x^2)$ for $x = 0, 1 \ldots, p-1$. So F_l is a p-regular bipartite graph. (Addition and subtraction are taken mod p.)

(b) Suppose $u = (i, j)$ and $w = (i', j') \neq (i, j)$ are in X, and that they have two (distinct) common neighbours. So there exist $x, y, \tilde{x}$, and $\tilde{y}$ such that

$$\begin{aligned} &(1) & i + x &= i' + \tilde{x} \\ &(2) & j + x^2 &= j' + \tilde{x}^2 \\ &(3) & i + y &= i' + \tilde{y} \\ &(4) & j + y^2 &= j' + \tilde{y}^2 \end{aligned}$$

Subtracting (3) from (1) and (4) from (2), we see that

$$\begin{aligned} &(5) & x - y &= \tilde{x} - \tilde{y} \\ &(6) & x^2 - y^2 &= \tilde{x}^2 - \tilde{y}^2 \end{aligned}$$

If $x - y = \tilde{x} - \tilde{y} = 0$, we obtain a contradiction because the two neighbours of (i, j) and (i', j') are not distinct. If $x - y = \tilde{x} - \tilde{y} \neq 0$, we may divide (6) by $x - y$ to obtain $x + y = \tilde{x} + \tilde{y}$, which together with (5) now implies that $x = \tilde{x}$ and $y = \tilde{y}$. It follows that $i = i'$ and $j = j'$, which is also a contradiction. (Note that this argument relied on the fact that if p is prime, every non-zero element in $\mathbb{Z}_p$ has a multiplicative inverse.) An analogous proof shows that (b) holds for two vertices u, w in Y.

[1] $\Gamma(u)$ denotes the set of neighbours of u.

(c) The $(k+1)$-vertex connectivity follows from the first two properties. Again, suppose first that u, w are two vertices in X. Let

$$U := \left(\bigcup_{u' \in \Gamma(u)} \Gamma(u') \right) - \{u\},\ W := \left(\bigcup_{w' \in \Gamma(w)} \Gamma(w') \right) - \{w\},$$

i.e. $U \subset X$ is the set of neighbours of neighbours of u (excluding u itself), and $W \subset X$ is the set of neighbours of neighbours of w (excluding w itself). Let $Z_1 := U \cap W$. Since the neighbourhoods of two neighbours of u only have u in their intersection by (b), we have that $|U| = (p-1)p$. By the same argument, $|W| = (p-1)p$. But $|X| = p^2$, so we must have $|Z_1| \geq p^2 - 2p \geq 0$. Now we claim that we can find $k+1$ internally vertex disjoint paths of length 4 between u and w. Pick a vertex u_1 in Z_1. So u_1 is adjacent to a neighbour z_1 of u and to a neighbour z_1' of w, by definition of U and W. This gives the first path $P_1 = \{u, z_1, u_1, z_1', w\}$ of length 4 between u and w. (The degenerate case where $z = z'$ can only happen once because $|\Gamma(u) \cap \Gamma(w)| \leq 1$; in that case we get a path of length 2.) Now remove z_1, z_1' and all their neighbours except u and w from the graph. So we remove at most $2(p-1)$ vertices from Z_1. Call the resulting set Z_2. Pick a vertex $u_2 \in Z_2 \subset U \cap W$. Then u_2 is adjacent to a neighbour z_2 of u and a neighbour z_2' of w. This gives a path P_2 of length 4 between u and w, and P_2 is internally vertex disjoint from P_1. Doing this $k+1$ times yields $k+1$ vertex disjoint paths $P_1, \ldots, P_{k+1}$. Since $|U \cap W| \geq p(p-2) \geq 2(p-1)(k+1)$ because $p \geq 2k+4$, all of the sets $Z_1, Z_2, \ldots, Z_k$ will be non-empty. A similar argument shows that we can find $k+1$ vertex disjoint paths between u and w if they both lie in Y. Finally, any vertex u has $p > k+1$ neighbours in Y, so no cutset of size k in F_l can contain all of the neighbours of u. Hence F_l is $(k+1)$-vertex connected. □

Proof of Lemma 2.4. We now proceed to prove part (1) of Lemma 2.4. The proof is based on two simple observations. Firstly, if a star T is centred at a vertex u of depth 3, then T can contain at most $p+1$ vertices of depth 3, namely u itself and all its neighbours. But the crucial point is that T will only contain depth 3 vertices of one type, the same type as the centre u. Secondly, if a star T is centred at a vertex w of depth 2, it can contain at most $k+1$ vertices of depth 3, namely the children of w. The crucial point here is that among the children of w, for each $i = 1, \ldots, k+1$ there will be at most one vertex of type i. A star centred at a vertex which is not of depth 2 or 3 cannot contain any vertices of depth 3.

So now let S be a galaxy of order k, say $S = \bigcup_{j=1}^{k} T_j$, where each T_j is a star centred at w_j. We need to show that there exists an i_0 such that all the vertices of type i_0 which are not in S are in the same component of $G - S$. Now if a centre w_j is of depth 3 and type i, ignore all vertices of that type completely. Since there are $k+1$ types, there is an i_0 such that no centre w_j is of type i_0. By Claim 2.5, the subgraph F_{i_o} spanned by the vertices of type

i_0 is $(k+1)$-connected. Since there are no stars with a centre of type i_0, each star contains at most one vertex of type i_0 by the second observation above. So S contains at most k vertices of type i_0, and thus $F_{i_0} - S$ is connected, so all vertices of type i_0 are in the same component of $G - S$, as claimed.

For the proof of (2), let $u \in G - S$ be a vertex of type i and depth 3, and assume u is a great-grandchild of v^*. u has p neighbours among the great-grandchildren of e^*. Any two of these only have u as a common neighbour, so u is joined to $p(p-1)$ great-grandchildren of v^* through vertex disjoint paths of length 2. It follows that u is joined to $p(p-1)$ grandchildren of v^* through vertex disjoint paths of length 3. We think of the grandchildren of v^* as *hubs* that let us reach vertices of type $j \neq i$. We claim that $G - S$ contains at least one hub and all its children.

Call a hub z *useless* if z or one of its children is contained in S, or if a vertex on one of the $u - z$ paths described above is contained in S. We shall bound the maximum number of useless hubs. To this end, let x_1 be the number of stars in S centred at vertices of depth 1, let x_2 be the number of stars centred at vertices of depth 2, let x_3 be the number of stars centred at vertices of depth 3 and type i, and let x_3' be the number of vertices centred at vertices of depth 3 and type $j \neq i$.

Now note that a star centred at a vertex of depth 1 can render at most $\lceil \frac{p^2}{n'} \rceil$ hubs useless. A vertex of depth 2 can render at most one hub useless. A vertex of depth 3 and type i can render at most p hubs useless, and a vertex of type 3 and type $j \neq i$ can render at most p hubs useless. Thus the total number of useless hubs is bounded above by

$$x_1 \lceil \frac{p^2}{n'} \rceil + x_2 + x_3 p + x_3' p.$$

Note that $x_1 + x_2 + x_3 + x_3' \leq k$. Clearly the worst case occurs when $x_1 = k$, and then there are $k \lceil \frac{p^2}{n'} \rceil$ useless hubs. Since $n' - k > 2$, we have that $\frac{n'-k}{n'} > \frac{2}{n'} \geq \frac{1}{p} + \frac{k}{p^2}$, and so $p^2(1 - \frac{k}{n'}) > p + k$ and thus

$$p(p-1) > k \lceil \frac{p^2}{n'} \rceil.$$

So u will be connected to a vertex of depth 2 and all of its children in $G - S$, as required.

A analogous argument works when u is a great-grandchild of e^*, and we have thus proved Lemma 2.4 □

3. The recognition algorithm

In this section, we restrict our attention to cutsets that are spanned by one tree only. However, we relax the restriction that this tree be a star, and instead consider trees of depth r. Instead of saying that a set S is spanned by a tree of depth at most r, we shall often say S has radius r. Since this definition does not yield a unique radius for a given set (if S has radius r, it also has radius $r+1, r+2, \ldots$), it should not be viewed as a strict definition, but rather as a useful shorthand for the cumbersome expression "S is spanned by a tree of depth at most r".

We begin by observing that testing whether a graph G has a cutset of radius r can easily be done in time $O(n^3 m)$ as follows. Suppose that v and w are vertices of G, and that there exists a set S of radius r such that v and w are in different components of $G-S$. Let u be the root of a tree T of depth at most r that spans S. Then letting T' be a BFS-tree of depth r in $G-\{v,w\}$ rooted at u, it is clear that T' also separates v and w in G. Thus, for every triple $\{u,v,w\}$ we grow a BFS-tree of depth r rooted at u in $G-\{v,w\}$ and check whether v and w are in the same component of $G-T$. As there are $O(n^3)$ triples and both BFS and checking for connectivity requires $O(m)$ time, the total run-time of this algorithm is $O(n^3 m)$.

We will now show that we can significantly improve on this trivial algorithm in the special case where $k=1$. In other words, we are now concerned with finding a single cutset of radius r. The idea of the algorithm is to decide, for each vertex v, whether v centres a cutset of radius r. We will be able to achieve this in time $O(rm)$ using a modified BFS-algorithm, thus obtaining a total run-time of $O(rnm)$.

So, let the graph G and the integer $r \geq 1$ be given. For a vertex $v \in G$, let $D_i(v)$ denote the vertices at distance i from v, and $\tilde{D}(v)$ the vertices at distance at least $r+1$. We will just write D_i and $\tilde{D}$ if the vertex v is fixed and there is no risk of confusion. Note that $\bigcup_{i=0}^{r} D_i$ is spanned by a tree of depth r rooted at v, e.g. a BFS-tree.

The algorithm relies on the following structural results: If G itself is a set of radius r, then G has no cutset of radius r if and only if it is a cycle or a clique. On the other hand, if the set of vertices $\tilde{D}(v)$ is non-empty for all vertices v, we will show that G has a cutset of radius r centred at v unless the set $\tilde{D}(v)$ is a connected subgraph and every vertex w in $D_i(v)$ can "escape" to $\tilde{D}(v)$ via a safe path consisting of vertices that are at distance strictly greater than r from v in $G-\{w\}$.

We begin with the case where G itself is spanned by a tree of depth at most r.

Lemma 3.1. *Let G be a graph such that $\tilde{D}(v)$ is empty for some $v \in G$, and let $r \geq 1$ be an integer. Then G has no cutset of radius r if and only if G is a cycle or a clique.*

Proof. Suppose G has no cutset of radius r, and let v be such that $\tilde{D}(v) = \emptyset$. Let $t \leq r$ be the largest integer such that $D_t \neq \emptyset$. The case $t=0$ is trivial, since then $G=\{v\}$ is obviously a clique. If $t=1$, G must be a clique as well, because if D_1 contained two non-adjacent vertices x and y, then $V(G) - \{x, y\}$ would be a cutset of radius 1. So suppose $t \geq 2$. It is easy to see that for $1 \leq i \leq t-1$ every vertex $w \in D_i$ must have a neighbour in D_{i+1}, otherwise we get a cutset of radius i. It is also easy to see that no D_i for $1 \leq i \leq t-1$ can be a clique, since otherwise D_i would be a cutset of radius 1 separating v from the vertices in D_t. On the other hand, D_t must be a clique, because $G - \{x, y\}$ is a set of radius t for any $x, y \in D_t$.

We claim that D_i must consist of two non-adjacent vertices for each $1 \leq i \leq t-1$. To see this, suppose that some D_i, $1 \leq i \leq t-1$, contains three vertices $\{x, y, z\}$. Since D_i is not a clique, we may without loss of generality assume that x and y are not adjacent. If $i > t-i$, let w be a vertex in D_{t-i} that lies on a shortest path from v to z. If $i < t-i$, pick $w \in D_{t-i}$ such that z lies on a shortest path from v to w. Such a w must exist, since z has a neighbour in D_{i+1}, which in turn has a neighbour in D_{i+2}, and so on up to D_{t-i}. If $i = t-i$, i.e. if $i = \frac{t}{2}$, pick $w = z$. Note that in all three cases, we have $w \in D_{t-i}$.

In order to derive a contradiction, we will show that in $G - \{x, y\}$, every vertex is at distance at most t from w. To see this, observe that since $(t-i)+i = t$, every vertex of $\{v\} \cup D_1 \cup \cdots \cup D_i$ lies within distance t of w, via a path through v. Moreover, such a path can be chosen to go through z if $i < t-i$ and thus contains neither x nor y. Also, since D_t is a clique, every vertex of $D_{i+1} \cup \cdots D_t$ lies within distance at most $i+1+(t-(i+1)) = t$ from w, via a path through D_t. Again, such a path can be chosen to go through z if $i > t-i$ and thus contains neither x nor y. It follows that the non-adjacent vertices x and y could be separated by removing $G - \{x, y\}$, a cutset of radius $t \leq r$ centred at w.

Let u and w be the two vertices of D_{t-1}. To complete the proof, we must show that either $D_t = \{x\}$ for some x adjacent to both u and w, or $D_t = \{x, y\}$ for some x and y such that x is adjacent to u but not to w, and y is adjacent to w but not to u (this completes the cycle). Observe that if, for some vertex $x \in D_t$, each neighbour of x in D_{t-1} is also adjacent to some other vertex $y \in D_t$, then the set $D_t - \{x\} \cup D_{t-1}$ is a cutset of radius 2 separating x from v. It follows immediately that $|D_t| \leq 2$, since we know that $|D_{t-1}| = 2$. If $D_t = \{x\}$, it is clear that x must be adjacent to both vertices of D_{t-1}, and if $D_t = \{x, y\}$, then x and y cannot have common neighbours in D_{t-1}, i.e. either x is adjacent to u and y to w, or vice-versa. Recall that we argued earlier that D_t had to be a clique, so (x, y) must be an edge, which completes the cycle.
For the converse, simply observe that cliques and cycles have no cutsets of radius r, for any r. □

So now assume that for every $v \in G$, there is at least one vertex at distance at least $r+1$ from v, i.e. $\tilde{D}(v) \neq \emptyset$. Given v, denote by $E(D_i)$ the set of edges between vertices in D_i, by W_i the set of vertices $D_i \cup \ldots \cup D_r$, and by G_i the induced subgraph on W_i. Let G_i' be the graph obtained from G_i by removing the edges $E(D_i)$. We say that $w \in D_i$ has an *exclusive* neighbour $u \in D_{i+1}$ if $(w, u) \in E(G)$ and $d_{G_i'}(u, w') > r - i$ for all vertices $w' \in D_i$, $w' \neq w$. Another way of saying this is that u is at distance strictly greater than r from v in $G - \{w\}$. The intuition behind this definition is that these exclusive neighbours provide a safe path to the set $\tilde{D}$ of vertices at distance greater than r from v, if they exist. We formalise this in the following lemma:

Lemma 3.2. *G has no cutset of radius r centred at v if and only if all of the following hold:*

(A) *$\tilde{D}$ is connected,*
(B) *every vertex in D_r has a neighbour in $\tilde{D}$,*
(C) *for every $t \leq r - 1$, every vertex in D_t has an exclusive neighbour in D_{t+1}.*

Proof. Suppose G has no cutset of radius r centred at v. It is clear that $\tilde{D}$ must be connected. If there were a vertex $w \in D_r$ without a neighbour in $\tilde{D}$, then $v \cup \bigcup_i D_i - \{w\}$ would be a cutset of radius r, contradiction. If some $w \in D_t$ did not have an exclusive neighbour in D_{t+1}, we could find a cutset of radius r as follows: The set of vertices $\bigcup_{i=1}^{t} D_i - \{w\}$ is spanned by a tree of depth t rooted at v. Since all of w's neighbours in D_{t+1} are at distance at most $r - t$ from some other vertex of D_t, there is a tree rooted at v containing $\bigcup_{i=1}^{t+1} D_i - \{w\}$, and this tree spans a cutset of radius r.

To prove the converse, it suffices to show that if we remove a tree T of depth at most r rooted at v, there exists a path from every $w \notin T$ to $\tilde{D}$ using only vertices of $V(G) - T$. Clearly this is true if $w \in \tilde{D}$. If $w \in D_r$, then w has a neighbour in $\tilde{D}$ by assumption. If $w \in D_t$ for some $t \leq r - 1$, note that its exclusive neighbour w' cannot be contained in any tree of depth r rooted at v that does not also contain w. But then w's exclusive neighbour has an exclusive neighbour w'' in D_{t+2} which cannot be reached by a tree rooted at v not containing w', and so on. This gives a path from w all the way to $\tilde{D}$. □

Lemmas 3.1 and 3.2 lead to a fast algorithm for checking whether a graph has a cutset of radius r. We proceed to show that the conditions on the graph stated in the lemmas can be tested for in polynomial time.

We write D_t and $\tilde{D}$ instead of $D_t(v)$ and $\tilde{D}(v)$ as there is no risk of confusion, and begin by observing that we can check in time $O(m)$ if G is a clique or a cycle. If that is the case, we know that G is has no cutset of radius r and we are done.

If G is neither a clique nor a cycle, we do the following for every vertex v: First we grow a BFS-tree rooted at v. This gives us the sets $D_1, \ldots, D_r$ and $\tilde{D}$. If $\tilde{D}$ is empty, we know G has a cutset of radius r by Lemma 3.1, and we are done. If $\tilde{D} \neq \emptyset$, we proceed to check whether Case (A) holds, i.e. whether $\tilde{D}$ is connected. This can be achieved in time $O(m)$. Next, we verify the condition given in Case (B), i.e. whether every vertex in D_r has a neighbour in $\tilde{D}$. This can also be done in time $O(m)$.

The final and most complex step is to check that, for each i such that $1 \leq i \leq r-1$, every vertex in D_i has an exclusive neighbour in D_{i+1} (Case (C)). Doing this requires $r-1$ independent phases, each of which is a modified BFS-algorithm. In phase i, we will process the level D_i. Phase i in turn will consist of $r-i$ steps. Let i be such that $1 \leq i \leq r-1$, and suppose that $D_i = \{w_1, \ldots w_p\}$.

For a vertex u, denote by $\mathcal{L}^i(u)$ the set of *labels* of u in phase i. At the start of phase i, all vertices in W_{i+1} are unlabelled, i.e. $\mathcal{L}^i(u) = \emptyset$ for all $u \in W_{i+1}$. We use the superscript i to emphasise that the labels in the $r-1$ phases are independent of one another.

We shall label all the vertices of $W_{i+1} = D_{i+1} \cup \ldots \cup D_r$ using the indices of the vertices in D_i in such a way that at the end of the process, we can simply read off the exclusive neighbours (in D_{i+1}) of the vertices in D_i from the labels. Labelling all the vertices in W_{i+1} will require $r-i$ steps.

The first of those $r-i$ steps is to scan the neighbours of the vertices $w_j \in D_i$ in D_{i+1}, proceeding in a breadth-first-search fashion. When we scan a vertex $u \in D_{i+1}$ adjacent to $w_j \in D_i$, either u has at most one label, or u already has two distinct labels. We update $\mathcal{L}^i(u)$ according to the following rules:

(1) If $\mathcal{L}^i(u) = \emptyset$ or $\mathcal{L}^i(u) = \{j'\}$ with $j' \neq j$, set $\mathcal{L}^i(u) := \mathcal{L}^i(u) \cup \{j\}$.
(2) If $\mathcal{L}^i = \{j\}$ or $|\mathcal{L}^i(u)| \geq 2$, we do not modify $\mathcal{L}^i(u)$.

We observe that after this first step, all the vertices in D_{i+1} will have one label if they are adjacent to only one vertex of D_i, and two distinct labels if they are adjacent to at least two vertices of D_i. Note that if $i = r-1$, there is only one step to be performed, so the algorithm terminates here.

In steps 2 to $r-i$, we only work in the graph G_{i+1}, the subgraph of G induced on the vertex set $W_{i+1} = \bigcup_{l=i+1}^{r} D_l$. At each step, we scan the neighbours of vertices in W_{i+1} whose set of labels was modified in the previous step. Suppose we scan a neighbour u of a vertex x. We add labels to u according to the following rules:

(1) If $\mathcal{L}^i(u) = \emptyset$ we set $\mathcal{L}^i(u) := \mathcal{L}^i(x)$.
(2) If $\mathcal{L}^i(u) = \{j\}$ and $\mathcal{L}^i(x)$ contains some $j' \neq j$, set $\mathcal{L}^i(u) := \mathcal{L}^i(u) \cup \{j'\}$.

(3) If $\mathcal{L}^i(u) = \mathcal{L}^i(x) = \{j\}$ or $|\mathcal{L}^i(u)| \geq 2$ we do not modify $\mathcal{L}^i(u)$.

The following claim establishes the connection between our labelling scheme and exclusive neighbours of the vertices $w_j \in D_i$.

Claim 3.3. *After t steps, the unlabelled vertices are the vertices at distance greater than t from D_i, the vertices with one label, say j, are the vertices at distance at most t from some $w_j \in D_i$, but at distance greater than t from all the other vertices in D_i, and the vertices with two labels j, l are the vertices at distance at most t from at least two vertices in D_i, namely w_j and w_l.*

Proof. We use induction on t. We have already observed that the base case is true just after the description of step 1 of the algorithm. So now suppose we just completed step t, and let u be any vertex in W_{i+1}.

If u is unlabelled, it must be at distance greater than t from D_i. Otherwise it would be adjacent to a vertex x at distance at most $t-1$ from D_i, which by induction hypothesis would have a least one label. But then u would have been labelled in step t.

Suppose u has two labels, say j and j'. If u already had label j after step $t-1$, then it is at distance at most $t-1 < t$ from $w_j \in D_i$. If it did not have label j, it inherited it from some vertex x that is at distance at most $t-1$ from w_j, by the induction hypothesis. The same is true for label j'. So after step t, u must be at distance at most t from w_j and $w_{j'}$.

Suppose u only has label j. So it is at distance at most t from w_j, either because it already had label j the step before, or because it inherited it from a vertex at distance at most $t-1$ during step t. If u were at distance at most t from a second vertex $w_{j'}$, then it would also have inherited the label j' by step t. □

For each i with $1 \leq i \leq r-1$, we run the labelling process described above for $t := r - i$ steps. Claim 3.3 then implies that if $u \in D_{i+1}$ is a neighbour of $w_j \in D_i$, then u is an exclusive neighbour if and only if $\mathcal{L}^i(u) = \{j\}$ after $r - i$ steps.

To summarise, each iteration of the labelling process identifies the exclusive neighbours of all the vertices in D_i for some i. Hence, we need $r-1$ iterations of the labelling process to handle all the vertices in $D_1 \cup \cdots \cup D_{r-1}$. It remains to determine the run-time of each iteration of the labelling algorithm.

Claim 3.4. *Each edge of G_i is considered at most four times during the labelling process.*

Proof. Let $e = (u, \bar{u})$ be an edge in G_i. We only consider e when either u's or $\bar{u}$'s label changes. The label of any vertex can change at most twice, so we consider e at most four times. □

It follows that we can process each level D_i in time $O(m)$. Since there are r levels, we can determine if all the vertices in $W_1 = D_1 \cup D_2 \cup \cdots \cup D_r$ have an exclusive neighbour in time $O(rm)$. Each vertex $v \in V$ could be the centre of a cutset of radius r, so the total running time is $O(rnm)$.

4. Related Work and Open Questions

As mentioned in the introduction, star-cutsets have received a great deal of attention because of their connection with the theory of perfect graphs. Chvátal's Star-cutset Lemma ([2]) asserts that no minimal imperfect graph contains a star-cutset (see also Cornuéjol's survey on the Strong Perfect Graph Theorem [3]).

In [6], Gunther gives a characterization of minimal k-regular graphs containing a k-clique which remain connected if k closed neighbourhoods are removed from the graph. In our notation, a set of k closed neighbourhoods is a galaxy of order k. Note however that we don't require that *all* the neighbours of the centres of the stars must be removed.

Another problem concerned with virus-like spreading behaviour through a network is the Firefighter problem. This was proposed by Hartnell in [7]. Finbow, King, MacGillivray and Rizzi give results for the special case of graphs of maximum degree 3 in [4].

In some situations, it may be desirable to maximize the spread of a virus through a network, rather than minimizing it. This is the case in the context of viral marketing, see e.g. the work by Kempe, Kleinberg and Tardos ([8]).

The present work focuses on recognizing whether a given network contains a cutset of radius r or a galaxy cutset. Another set of problems arises from trying to design networks without cutsets of radius r or galaxy cutsets. In [12], the authors present an algorithm that finds a spanning subgraph without star-cutsets in a graph with no cutsets of radius 3, and show that for $r \geq 4$, it is NP-hard to determine whether a given graph contains a spanning subgraph with no cutset of radius r. The problem of designing low cost graphs with high vertex- or edge-connectivity has been studied extensively, see Khuller ([9]) or Kortsarz and Nutov ([11]) for surveys on the literature.

It remains open whether the recognition algorithm presented in Section 3 can be improved. The authors conjecture that it should be possible to remove the dependence on r and obtain an $O(nm)$ algorithm. In view of the hardness result in Section 2, bi-criteria problems such as the following become interesting: Given a graph G and an integer k, determine in polynomial time that either G has no galaxy cutset of order k, or exhibit a galaxy cutset of order αk (for some constant $\alpha > 1$).

5. Acknowledgments

The authors would like to thank Brighten Godfrey for helpful discussions concerning the relevance of galaxy cutsets to practical problems such as Denial of Service attacks on networks.

References

[1] M. Agrawal and N. Kayal and N. Saxena. "PRIMES is in P". *Ann. of Math.*, 160(**2**), pp. 781-793, 2004.
[2] V. Chvátal. "Star-cutsets and perfect graphs". *Journal of Combinatorial Theory B*, 39, pp. 189-199, 1985.
[3] G. Cornuéjols. "The Strong Perfect Graph Theorem". *Optima*, 70, pp. 2-6, 2003.
[4] S. Finbow and A. King and G. MacGillivray and R. Rizzi. "The firefighter problem for graphs of maximum degree three". *Discrete Mathematics*, 307(**16**), pp. 2094-2105, 2007.
[5] M. Garey and D. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
[6] G. Gunther. "Neighbour-connectivity in regular graphs". *Discrete Appl. Math.*, 11(**3**), pp. 233-243, 1985.
[7] B.L. Hartnell. "Firefighter! An application of domination". Presentation. 24th Manitoba Conference on Combinatorial Mathematics and Computing, 1995.
[8] D. Kempe and J. Kleinberg and É. Tardos. "Maximizing the spread of influence through a social network". In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137-146, 2003.
[9] S. Khuller. "Approximation algorithms for finding highly connected subgraphs". In D. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pp. 236-265, PWS Pub. Co., Boston, 1995.
[10] G. Kortsarz and R. Krauthgamer and J. Lee. "Hardness of approximation algorithm for vertex-connectivity network design problems". *SIAM Journal on Computing*, 33(**3**), pp. 704-720, 2004.
[11] G. Kortsarz and Z. Nutov. "Approximating minimum cost connectivity problems". In T. Gonzalez, editor, *Handbook on Approximation Algorithms and Metaheuristics*, Chapter 58, Chapman & Hall / CRC, 2007.
[12] N. Sonnerat and A. Vetta. "Network Connectivity and Malicious Attacks". Pre-print.