The External Network Problem [*]

Jan van den Heuvel [†] and Matthew Johnson [†]

CDAM Research Report LSE-CDAM-2004-15

December 2004

Abstract

The connectivity of a communications network can often be enhanced if the nodes are able, at some expense, to form links using an external network. In this paper, we consider the problem of how to obtain a prescribed level of connectivity with a minimum number of nodes connecting to the external network.

Let $D = (V, A)$ be a digraph. A subset X of vertices in V may be chosen, the so-called external vertices. An internal path is a normal directed path in D; an external path is a pair of internal paths $p_1 = v_1 \cdots v_s$, $p_2 = w_1 \cdots w_t$ in D such that v_s and w_1 are external vertices (the idea is that v_1 can contact w_t along this path using an external link from v_s to w_1). Then D is externally-k-arc-strong if for each pair of vertices u and v in V, there are k arc-disjoint paths (which may be internal or external) from u to v.

We present polynomial algorithms that, given a digraph D and positive integer k, will find a set of external vertices X of minimum size subject to the requirement that D must be externally-k-arc-strong.

We also consider two generalisations of the problem : first we suppose that the number of arc-disjoint paths required from u to v may differ for each choice of u and v, and secondly we suppose that each vertex has a cost and that we are required to find a minimum cost set of external vertices. We show that these two problems are NP-hard.

Finally, we consider the analogue of this problem for vertex-connectivity in undirected graphs. A graph G with set of external vertices X is externally-k-connected if there are k vertex-disjoint paths joining each pair of vertices in G. We present polynomial algorithms for finding a minimum size set of external vertices subject to the requirement that G must be externally-k-connected for the cases $k \in \{1, 2, 3\}$.

1 Introduction

Communications networks have an obvious modelling in terms of graphs or digraphs, and often it is possible to express a network's reliability, or endurance, using some graph-theoretic property of the (di-)graph that represents it. In this note we consider the following situation : suppose that the nodes of a network N can access an external network. For example, N could be a fairly loose network whose nodes can also communicate using a second, more reliable, network. Another example is a wireless network where additional links can be formed using satellite connections. Suppose further that by using the external network the reliability of N can be improved, but that its use should be minimised as there is an associated cost (this could simply be a financial cost, or it could be a loss of security, or a restriction of node mobility).

[*] Research supported by EPSRC MathFIT grant no. GR/R83514/01.

[†] Centre for Discrete and Applicable Mathematics, Department of Mathematics, London School of Economics, Houghton Street, London WC2A 2AE, U.K.; email : {jan, matthew}@maths.lse.ac.uk.

The problem we have is this:

- if we require N to have a prescribed level of reliability, which nodes should use the external network?

This is the most general statement of the *External Network Problem.*

In this paper we are concerned with the problem when the reliability of the network N is given by the edge-connectivity of a graph or the arc-strong connectivity of a digraph, or the vertex-connectivity of a graph. All graphs may have multiple edges.

Let us give some definitions that will allow us to formally state the problem where we consider the arc-strong connectivity of a digraph $D = (V, A)$; the most interesting version for which we can provide general answers. An *internal path* in D is a sequence of vertices and arcs $v_1 e_1 v_2 \cdots e_{t-1} v_t$ where, for $1 \le i \le t-1$, e_i is the arc $v_i v_{i+1}$. The *set of external vertices* is a subset of V and is denoted X_V. An *external path* of D is a pair of internal paths $p_1 = v_1 \cdots v_s$, $p_2 = w_1 \cdots w_t$ where v_s and w_1 are external vertices. Thus, in this paper, when we use the term *path* without specifying whether it is internal or external, it can be either. The digraph D is *externally-k-arc-strong*, and X_V is *k-good*, if between each pair of vertices in V, there are k arc-disjoint paths. A digraph is *internally-k-arc-strong* if each pair of vertices is joined by k-arc-disjoint internal paths (this is just the usual definition of k-arc-strong).

<u>External Network Problem</u>

Input: A digraph $D = (V, A)$ and a positive integer k.
Output: A k-good set of external vertices $X_V \subseteq V$ of minimum size.

In Section 5 we shall present polynomial algorithms that will find a k-good set X_V of minimum size. In the next section we consider the special case of undirected graphs. In Sections 3 and 4 we introduce some preliminary results needed before we can present the algorithms.

First we discuss an alternative setting for this problem. Suppose D is a digraph which is not k-arc-strong. The (symmetric) *Source Location Problem* requires us to find a smallest possible set S (a subset of V that we call a set of sources) such that for each $v \in V \setminus S$ there are k arc-disjoint internal paths from S to v and k arc-disjoint paths from v to S. We show that a k-good set of external vertices is also a set of sources and vice versa. Let X_V be a k-good set of external vertices for D, and choose a vertex $v \in X_V$. For each $u \in V \setminus X_V$, there are k arc-disjoint paths from u to v. If any of the paths are external, then consider only the first internal path that joins u to an external vertex: this provides k arc-disjoint internal paths from u to X_V. To obtain k arc-disjoint internal paths from X_V to u, take the k arc-disjoint paths from v to u and, for those that are external, consider only the second internal path that joins an external vertex to u. Thus X_V is a set of sources. Now suppose that S is a set of sources. Identify the vertices of S to form a new digraph D'. Clearly D' is internally-k-arc-strong. Thus each pair of vertices are joined by k arc-disjoint internal paths, and, if we let the vertices of S be external vertices, these correspond to arc-disjoint paths in D. So S is a k-good set of external vertices. Ito *et al.* [4] described a polynomial algorithm for the Source Location Problem in the case where k is fixed; the algorithms we present (which would also provide solutions to the Source Location Problem) are polynomial even if k is not fixed.

If the principal cause of unreliability in a network is node failure, then network reliability can be modelled using vertex-connectivity. We cannot expect that, in general, the Source Location Problem (which has been extensively studied) will provide solutions to the External Network Problem : the Source Location Problem with vertex-connectivity requirements (see [3, 5]) is not equivalent to the natural formulation of the External Network Problem with vertex-connectivity requirements. This is discussed further in Section 8.

It is also possible to generalise the External Network Problem with arc-connectivity requirements for digraphs : let $c : V(D) \longrightarrow \mathbb{R}$ be a cost function and let $d : V(D) \times V(D) \longrightarrow \mathbb{R}$ be a demand function. Now a set of external vertices is *good* if for each pair of vertices $u, v \in V$ there are $d(u, v)$ arc-disjoint (internal or external) paths from u to v, and the problem is to find a minimum cost set of good external vertices. In Section 7, we show that this generalisation is NP-hard if either the cost or demand function is non-uniform.

2 Undirected Graphs

In this section $G = (V, E)$ is an undirected graph and k is a positive integer. The following definitions are the obvious analogues of those for digraphs : a set of external vertices X_V is a subset of V; an internal path of G is a sequence of incident vertices and edges and an external path is pair of internal paths where the final vertex of the first and the initial vertex of the second are external vertices; G is externally-k-edge-connected, and X_V is k-good, if each pair of vertices is joined by k edge-disjoint paths.

We shall describe how to find a minimum size k-good set of external vertices for G. This could be achieved by replacing each edge of G by a pair of oppositely oriented arcs and using the algorithms for digraphs presented in the final section. We shall see, however, that the problem for undirected graphs is far simpler than that for digraphs.

For $P \subseteq V$, the *degree* $d(P)$ is the number of edges joining P to $V \setminus P$. If $P = \{v\}$, we just write $d(v)$.

Definition 1 *A non-empty set $P \subseteq V$ is* k-critical *if $d(P) < k$, and, for each proper subset Q of P, $d(Q) \geq k$.*

It can easily be seen that X_V is a k-good set of external vertices if and only if it intersects each k-critical set. We say that X_V *covers* any k-critical set that it intersects. Thus our aim is to cover all the k-critical sets of G with as few vertices as possible. We need the following lemma, which is well-known and easily proved.

Lemma 1 *Let P and Q be subsets of V. Then $d(P) + d(Q) \geq d(P \setminus Q) + d(Q \setminus P)$.*

Proposition 2 *The k-critical sets of G are pairwise disjoint.*

Proof : Let P and Q be distinct k-critical sets of G. By the definition of k-critical, one cannot be a subset of the other. So if P and Q are not disjoint, then $P \setminus Q \neq \varnothing$ and $Q \setminus P \neq \varnothing$. Using the definition of critical again,

$$\begin{aligned} d(P \setminus Q) + d(Q \setminus P) &\geq 2\,k, \qquad \text{and} \\ d(P) + d(Q) &< 2\,k. \end{aligned}$$

This contradicts Lemma 1 which proves that the hypothesis that P and Q are not disjoint must be false. □

We immediately have the following result.

Theorem 3 *A minimum size k-good set of external vertices for a graph G contains as many vertices as there are k-critical sets in G.*

It is not necessary, however, to find the k-critical sets to find a minimum size k-good set of external vertices. To check that a set X_V is k-good, identify the vertices of X_V to form a graph G' : X_V is k-good if and only if G' is k-edge-connected which can be checked using, for example, a flow algorithm. To find a minimum k-good set, let $X_V = V$ and then repeatedly remove any vertex from X_V as long as the set obtained is k-good. In this way a minimal set is obtained, and every minimal set is a set of minimum size.

3 Critical Sets in Digraphs

In the sequel, $D = (V, A)$ is a digraph and k is a positive integer.

For $P \subseteq V$, $d^-(P)$ is the number of arcs joining $V \setminus P$ to P, and $d^+(P)$ is the number of arcs joining P to $V \setminus P$.

Definition 2 *A non-empty set $P \subseteq V$ is k-*critical *if $d^-(P) < k$ or $d^+(P) < k$, and, for each proper subset Q of P, $d^-(Q) \geq k$ and $d^+(Q) \geq k$.*

A k-critical set P is *k-in-critical* if $d^-(P) < k$ and *k-out-critical* if $d^+(P) < k$ (it is possible for a set to be both k-in-critical and k-out-critical).

Again, it can easily be seen that X_V is a k-good set of external vertices if and only if it intersects each k-critical set.

Definition 3 *Let $\mathcal{P} = P_1, \ldots, P_s$ be a collection of k-critical sets. The* relation graph *of these sets has vertex set $\mathcal{P}$ and contains an edge joining P_i and P_j, $i \neq j$, if $P_i \cap P_j \neq \varnothing$.*

We will say that k-critical sets are *neighbours* if they intersect (even when not explicitly referring to a relation graph).

We investigate the structure of relation graphs. The results of this section follow easily from the results of Ito *et al.* [4] on the Source Location Problem.

Lemma 4 *If P is a k-in-critical set in D and $Q \neq P$ is a k-out-critical set in D, then $P \cap Q = \varnothing$.*

Proof : By the definition of k-critical neither P nor Q is a subset of the other. If they are not disjoint, then the three sets of vertices $S_1 = P \setminus Q$, $S_2 = P \cap Q$ and $S_3 = Q \setminus P$ are all non-empty. Let $S_4 = V \setminus (P \cup Q)$ (this is possibly the empty set), and let $s_{i,j}$ be the number of arcs from S_i to S_j. As $d^-(S_1) \geq k$ and $d^+(S_3) \geq k$ (since they are proper subsets of k-critical sets),

$$s_{2,1} + s_{3,1} + s_{4,1} \geq k \quad (1)$$

$$s_{3,1} + s_{3,2} + s_{3,4} \geq k. \quad (2)$$

And as $P = S_1 \cup S_2$ is in-critical and $Q = S_2 \cup S_3$ is out-critical,

$$\begin{aligned} s_{3,1} + s_{3,2} + s_{4,1} &< k, & (3) \\ s_{2,1} + s_{3,1} + s_{3,4} &< k. & (4) \end{aligned}$$

Adding (1) to (2), and (3) to (4), we obtain a contradiction. □

Recall that an undirected graph G is *chordal* if it contains no induced cycles of length more than three (that is, if every cycle of length at least four has a *chord*, an edge joining two vertices that are not adjacent in the cycle).

Proposition 5 *For any digraph, the relation graph G of any collection of its k-critical sets is a chordal graph and if a collection of k-critical sets $\mathcal{Q}$ form a clique in G, then $\bigcap_{P \in \mathcal{Q}} P \neq \varnothing$.*

Proof: If G is not chordal, there is a cycle induced by some sets $P_1, \ldots, P_t$, $t \geq 4$. By Lemma 4, these sets are either all in-critical or all out-critical. Suppose that the latter holds (it will be clear that the former case can be similarly proved). We can assume that $P_i \cap P_j \neq \varnothing$ if $|i - j| = 1 \bmod t$ and $P_i \cap P_j = \varnothing$ otherwise. Then

$$\sum_{|i-j|=1 \bmod t} d^+(P_i \cap P_j) \;\leq\; \sum_{i=1}^{t} d^+(P_i) \;<\; t\,k.$$

The first sum is obtained by counting the number of arcs from $P_i \cap P_j$ to $V \setminus (P_i \cap P_j)$. Each such arc joins P_i to $V \setminus P_i$ or P_j to $V \setminus P_j$ (or both) so is also counted at least once when evaluating the second sum. The second inequality follows from the definition of k-critical. Thus, as there are t terms in the first sum, $d^+(P_i \cap P_j) < k$ for some i, j, $|i - j| = 1 \bmod t$, a contradiction as $P_i \cap P_j$ is a proper subset of a k-critical set. This proves that G is chordal.

If $\mathcal{Q}$ is a clique of size 1 or 2, then there is a vertex in every set in $\mathcal{Q}$. Let $\mathcal{Q} = P_1, \ldots, P_t$, $t \geq 3$, be the smallest clique such that $\bigcap_{i=1}^{t} P_i = \varnothing$. Again we can assume that the k-critical sets are all out-critical. Then

$$\sum_{i=1}^{t} d^+\left(\bigcap_{j \neq i} P_j\right) \;\leq\; \sum_{i=1}^{t} d^+(P_i) \;<\; t\,k.$$

Again, every arc counted when evaluating the first sum is counted at least once for the second sum (the second inequality is the same as before). Each of the t sets in the first sum is non-empty (as each intersection is of a collection of sets that forms a clique smaller than $\mathcal{Q}$, so they have a common vertex), and $d^+\left(\bigcap_{j \neq i} P_j\right) < k$ for some i, a contradiction as these sets are proper subsets of k-critical sets. Therefore there must be a vertex contained in every set in $\mathcal{Q}$, and the proposition is proved. □

Theorem 6 *A minimum size k-good set of external vertices for a digraph D is the same size as a maximum size family of disjoint k-critical sets.*

Proof: A k-good set of external vertices must be at least as big as a family of disjoint k-critical sets since it must intersect each set in the family. We prove the theorem by finding a k-good set of external vertices $X_V = \{x_1, \ldots, x_t\}$ and a family of disjoint k-critical sets $P_1, \ldots, P_t$ such that x_i covers P_i, $1 \leq i \leq t$.

We shall use a well-known property of chordal graphs (see, for example, [1]).

- For any chordal graph G, there exists $v \in V(G)$ such that v and its neighbours form a clique.

Therefore, using this property and Proposition 5,

- for any collection $\mathcal{P}$ of k-critical sets, there exists a k-critical set $P \in \mathcal{P}$ such that the intersection of P and all its neighbours in $\mathcal{P}$ is non-empty; we call P an *end-set*.

Let G_1 be the relation graph of all the k-critical sets of D. Let P_1 be an end-set of G_1, and let x_1 be a vertex that covers P_1 and all its neighbours. Let G_2 be the relation graph of all the$k$-critical sets not covered by x_1 and note that it contains no k-critical set that intersects P_1. Now suppose that we have found $x_1, \ldots, x_s$ and $P_1, \ldots, P_s$, $s < t$, and that the relation graph G_{s+1} of k-critical sets not yet covered contains no critical set that intersects P_i, $1 \leq i \leq s$. Let P_{s+1} be an end-set of G_{s+1} and let x_{s+1} be a vertex that covers P_{s+1} and all its neighbours. Note that G_{s+2}, the relation graph of uncovered k-critical sets, contains no set that intersects P_i, $1 \leq i \leq s+1$. If G_{s+2} is the null graph, then $s+1 = t$ and we are done. □

4 External Subsets

We will describe algorithms to find a minimum size k-good set in a digraph in the next section. The algorithms will use a generalisation of sets of external vertices: a *set of external subsets* is a disjoint collection of non-empty sets $\mathcal{X}_V = \{X_1, \ldots, X_t\}$ such that $X_i \subseteq V$ for $1 \leq i \leq t$.

Definition 4 *A set of external subsets is* k-good *if* $\bigcup_{i=1}^{t} X_i$ *is a k-good set of external vertices.*

The remaining definitions in this section assume that $\mathcal{X}_V$ is a k-good set of external subsets. An external subset $X \in \mathcal{X}_V$ is *redundant* if $\mathcal{X}_V \setminus \{X\}$ is also k-good. If $\mathcal{X}_V$ contains no redundant set, it is *minimally k-good*.

Definition 5 *For $u \in V$ and $X \in \mathcal{X}_V$, if $(\mathcal{X}_V \setminus \{X\}) \cup \{\{u\}\}$ is also a k-good set of external subsets, then u is an* alternative *to X. The* unrestricted set of alternatives *to X contains all such u and is denoted $A(X)$. The* restricted set of alternatives *to X is $A(X) \cap X$ and is denoted $B(X)$.*

In the algorithms, a common operation is to alter $\mathcal{X}_V$ by replacing one of the subsets X by its (restricted or unrestricted) set of alternatives X'. Notice that if $X' \neq \varnothing$, then $(\mathcal{X}_V \setminus \{X\}) \cup \{X'\}$ is also k-good.

Definition 6 *For $X \in \mathcal{X}_V$, a k-critical set is an* essential set *of X if it is covered by X but not by any other external subset.*

A vertex is an alternative to X if and only if it covers its essential sets. Thus if X is not redundant, $A(X)$ is equal to the intersection of the essential sets of X; if X is redundant, it has no essential sets and $A(X) = V$.

Definition 7 *If a set $X \subseteq V$ is a subset of an k-critical set P, then P is a* confining set *of X. If X is equal to the intersection of its confining sets, then it is* confined*.*

If a set $X \in \mathcal{X}_V$ is not redundant, then $A(X)$ is confined by the essential sets of X. If X is confined, then $B(X)$ is confined by the confining sets of X and the essential sets of X.

Definition 8 *$\mathcal{X}_V$ is* stable *if for each $X \in \mathcal{X}_V$, $X = A(X)$. It is* consistent *if for each $X \in \mathcal{X}_V$, $X = B(X)$.*

Notice that if $\mathcal{X}_V$ is stable, then it is also consistent and each $X \in \mathcal{X}_V$ is confined.

Proposition 7 *If X_V is a minimum size k-good set of external vertices and $\mathcal{X}_V$ is a stable set of external subsets, then $|X_V| \geq |\mathcal{X}_V|$.*

In the next section, we will see that from a stable set, it is possible to find a minimum size k-good set that contains one vertex from each external subset. This will prove that $|X_V| \leq |\mathcal{X}_V|$. Thus $|X_V| = |\mathcal{X}_V|$.

Proof: Let $\mathcal{X}_V = \{X_1, \ldots, X_t\}$. We will prove that $|X_V| \geq |\mathcal{X}_V|$ by finding a disjoint collection of k-critical sets $P_1, \ldots, P_t$.

As each external subset in $\mathcal{X}_V$ is its own set of alternatives, it is equal to the intersection of its essential sets, and each of these essential sets intersects only one source-set.

We use the structure of relation graphs of k-critical sets of D. Consider the relation graph G_1 of the essential sets of all of the sets $X_1, \ldots, X_t$. Let P_1 be an end-set of this graph, and, we might as well assume, that P_1 is an essential set for X_1. Thus $X_1 = A(X_1) \subseteq P_1$, and the essential sets of X_1 are P_1 and none, some or all of its neighbours. Recall from Proposition 5 that the intersection of P_1 with all its neighbours is non-empty. Each vertex in this intersection is certainly an alternative to X_1 — it covers all the essential sets — and thus a member of X_1. Therefore, as the essential sets each intersect only one of the source-sets, P_1 and *all* of its neighbours in G_1 are essential sets of X_1.

Now consider the relation graph G_2 of the essential sets except those of X_1. Let P_2 be an end-set and suppose it is an essential set for X_2. Note that P_2 does not intersect P_1 as no essential set that intersects P_1 is included in G_2. By the same argument as before, P_2 and all of its neighbours in G_2 are essential sets of X_2.

Then we look for an end-set in the relation graph of all essential sets except those of X_1 and X_2. From this and further repetitions we find $P_3, \ldots, P_t$. □

When we present the algorithms in the next section, it will be assumed that it is possible to check that a set of external vertices X_V is k-good in polynomial time. This can be done by contracting the vertices of X_V to obtain a digraph D' and checking that D' is k-arc-strong using, for example, a flow algorithm. Furthermore, we can check that $\mathcal{X}_V$ is a minimally k-good by checking that $\mathcal{X}_V \setminus \{X\}$ is not k-good for each $X \in \mathcal{X}_V$. To find the set of alternatives to some $X \in \mathcal{X}_V$, first check whether or not $\mathcal{X}_V \setminus \{X\}$ is k-good: if it is, then $A(X) = V$; otherwise $A(X)$ contains each vertex $u \in V$ such that $(\mathcal{X}_V \setminus \{X\}) \cup \{\{u\}\}$ is k-good.

5 Algorithms for Digraphs

We present two polynomial algorithms : StableSubsets finds a stable set of external subsets for a digraph D, and MinimumSet takes a stable set and finds a set of external vertices containing a single vertex from each external subset. By Proposition 7, this set of external vertices will have minimum size.

```
Algorithm StableSubsets

Input : A digraph $D = (V, A)$ where $V = \{v_1, \dots, v_n\}$.
Output : A $k$-good stable set of external subsets $\mathcal{X}_V$ for $D$.

let $\mathcal{X}_V = \{ \{v_1\}, \dots, \{v_n\} \}$;
while there exists a redundant set $R \in \mathcal{X}_V$ do
  let $\mathcal{X}_V = \mathcal{X}_V \setminus \{R\}$;
end /* while */
while there exists $Y \in \mathcal{X}_V$, $Y \neq A(Y)$ do                      /* L1 */
  let $\mathcal{X}_V = ( \mathcal{X}_V \setminus \{Y\} ) \cup \{A(Y)\}$;
  while there exists a redundant set $R \in \mathcal{X}_V$ then              /* L2 */
    let $\mathcal{X}_V = \mathcal{X}_V \setminus \{R\}$;
    while there exists $Z \in \mathcal{X}_V$, $Z \neq B(Z)$ do                  /* L3 */
      let $\mathcal{X}_V = ( \mathcal{X}_V \setminus \{Z\} ) \cup \{B(Z)\}$.
    end /* while */
  end /* while */
end /* while */

Output $\mathcal{X}_V$.
```

```
Algorithm MinimumSet

Input : A digraph $D = (V, A)$ with a $k$-good stable set of external subsets $\mathcal{X}_V$.
Output : $X_V$, a minimum size $k$-good set of external vertices for $D$.

while there exists $R' \in \mathcal{X}_V$, $|R'| \geq 2$ do                    /* L4 */
  choose $u \in R'$;
  let $\mathcal{X}_V = ( \mathcal{X}_V \setminus \{R'\} ) \cup \{\{u\}\}$;
  while there exists $Z \in \mathcal{X}_V$, $Z \neq B(Z)$ do                    /* L5 */
    let $\mathcal{X}_V = ( \mathcal{X}_V \setminus \{Z\} ) \cup \{B(Z)\}$.
  end /* while */
end /* while */
let $X_V = \bigcup_{X \in \mathcal{X}_V} X$.
Output $X_V$.
```

Using these algorithms, the time needed to find a minimum size k-good set of external vertices for a digraph $D = (V, A)$ is $O(n^4\, m \log(n^2/m))$ where $n = |V|$ and $m = |A|$. The bottleneck is the loop labelled L1 : it can be shown that this loop takes time $O(n^3\, S(n))$, where $S(n)$ is the complexity of an algorithm that decides whether a set of vertices is k-good.

As we remarked at the end of the previous section this can be done using a flow algorithm; in particular an algorithm of Hao and Orlin [2] means we can take $O(n\,m\,\log(n^2/m))$ for $S(n)$.

In the remainder of this section we prove the efficacy of the two algorithms.

To begin, StableSubsets considers the vertex set of V as a set of external subsets $\mathcal{X}_V$. Redundant sets are discarded until $\mathcal{X}_V$ is a minimal k-good set of external subsets.

The main part of the algorithm contains three nested **while** loops labelled L1, L2 and L3. We say that the algorithm *enters* a loop if the *loop condition* is satisfied. For example, the loop condition for L2 is that $\mathcal{X}_V$ contains a redundant set. Inside the loops $\mathcal{X}_V$ is altered by replacing external subsets by their sets of alternatives or by discarding external subsets. We shall show that after each alteration $\mathcal{X}_V$ is still k-good. Thus if the algorithm does not enter L1, then for each $X \in \mathcal{X}_V$, $X = A(X)$. Hence $\mathcal{X}_V$ is a k-good stable set of external subsets and we have the required output.

We shall prove the following stronger claims. (Recall that a set of external subsets $\mathcal{X}_V$ is consistent if $X = B(X)$ for each $X \in \mathcal{X}_V$.)

Claim 1 *(a) Each time the algorithm considers the loop condition for L1, $\mathcal{X}_V$ is a consistent k-good set of external subsets, and each $X \in \mathcal{X}_V$ is confined or a singleton, but not redundant.*
(b) Each time the algorithm considers the loop condition for L2, $\mathcal{X}_V$ is a consistent k-good set of external subsets, and each $X \in \mathcal{X}_V$ is confined or a singleton.

We prove the claim by induction. The first time the algorithm considers the loop condition for L1, $\mathcal{X}_V$ is a set of singletons and minimal. Thus each external subset is non-redundant and its own restricted set of alternatives, and hence $\mathcal{X}_V$ is consistent.

Suppose that $X = B(X)$ and X is non-redundant for each $X \in \mathcal{X}_V$ and that L1 is entered. A set Y is replaced by $Y' = A(Y)$ (and $Y \subsetneqq Y'$ since $Y = B(Y) \subseteq A(Y) \neq Y$). Clearly $B(Y') = Y'$, Y' is confined, and $\mathcal{X}_V$ is k-good. For each $X \in \mathcal{X}_V$, $X \neq Y'$, X remains a singleton or confined, and also the set of alternatives to X will, if anything, have grown when Y was replaced by $A(Y)$. Thus for all $X \in \mathcal{X}_V$, the claim that X is confined or a singleton and $X = B(X)$ still holds when the loop condition for L2 is considered. So if L2 is not entered, none of the external subsets is redundant; the algorithm returns to consider the condition for L1 and Claim 1 (a) holds.

Suppose that L2 is entered and a redundant set R is removed from $\mathcal{X}_V$. If L3 is not entered, then Claim 1 (b) holds.

So suppose that L3 is entered. We call the process encoded by L3 *reduction*: to *reduce* a k-good set of external subsets is to arbitrarily choose a external subset X and replace it by $B(X)$, and to repeat this until a k-good consistent set is obtained (note that after each alteration, the restricted sets of alternatives of each external subset must be recalculated). A k-good set of external subsets is *reducible* if this process is possible, that is, if each time we replace a external subset X by $B(X)$ we obtain a k-good set of external subsets (clearly the process must eventually terminate as the external subsets are continually getting smaller).

Proposition 8 *Let $\mathcal{X}_V$ be a consistent k-good set of external subsets such that each $X \in \mathcal{X}_V$ is confined or a singleton. Let $Z \in \mathcal{X}_V$ and let $Z' = \varnothing$ or $Z' = \{z\}$ for some $z \in Z$. If $(\mathcal{X}_V \setminus \{Z\}) \cup \{Z'\}$ is a k-good set of external subsets, then it is reducible. Moreover, each X in the set of external subsets obtained after the reduction is confined or a singleton.*

The proof is left to the end of the section.

Apply the proposition to the set $\mathcal{X}_V$ obtained just before R is discarded with $Z = R$ and $Z' = \varnothing$. Thus after the algorithm has finished looping through L3, Claim 1 (b) holds. Then another redundant set may be discarded and the algorithm may begin to loop through L3 again. When no more redundant sets can be found, the algorithm has finished its run through L1 and Claim 1 (a) holds. We have shown that StableSubsets will output a k-good stable set of external subsets.

MinimumSet contains two nested **while** loops, L4 and L5. If the algorithm does not enter L4, then each external subset contains one vertex and from this we obtain a minimum size k-good set of external vertices.

As before, we must show that as $\mathcal{X}_V$ is altered, it is always a k-good set of external subsets.

Claim 2 *Each time the algorithm considers the loop condition for L4, $\mathcal{X}_V$ is a k-good consistent set of external subsets, and each $X \in \mathcal{X}_V$ is confined or a singleton.*

We use induction to prove the claim. The first time the algorithm considers the loop condition for L4, $\mathcal{X}_V$ is stable and the claim holds (see the remark after Definition 8). Assume the claim holds as L4 is entered. After replacing an external subset R' by one of its alternatives, $\mathcal{X}_V$ is still k-good (by Definition 5). If the algorithm does not enter L5, then the claim is true. All that remains to be proved is that once the algorithm finishes going through L5, the resulting $\mathcal{X}_V$ satisfies the conditions in the claim. Notice that L5 is identical to L3 : so applying Proposition 8 with $Z = R'$ and $Z' = \{u\}$ will guarantee that this is the case. We have shown that MinimumSet will output a minimum size k-good set of external vertices.

We require one further result on chordal graphs before we prove Proposition 8.

Lemma 9 *Let u and v be non-adjacent vertices in a chordal graph G, and suppose that W_1, the set of vertices adjacent to both u and v, is non-empty. Let W_2 contain each vertex (other than u and v) that is adjacent to every vertex in W_1. Then u and v are in different components in $G - (W_1 \cup W_2)$.*

Proof : We show that if there is a u-v path in $H = G - (W_1 \cup W_2)$, then there is an induced cycle of length greater than 3 in G.

Let $up_1 \cdots p_r v$ be the shortest u-v path in H ($r \geq 2$ as $p_1 \notin W_1$). As $p_1 \notin W_2$, there exists $w \in W_1$ such that p_1 and w are not adjacent in G. If possible, choose the smallest $i \geq 2$ such that p_i is adjacent to w in G : then $\{u, p_1, \ldots, p_i, w\}$ induces a cycle in G. If no p_i is adjacent to w, then $\{u, p_1, \ldots, p_r, v, w\}$ induces a cycle. □

Proof of Proposition 8 : Let $\mathcal{X}_V$, Z and Z' be as in the proposition. During the reduction of $(\mathcal{X}_V \setminus \{Z\}) \cup \{Z'\}$ we repeatedly select an external subset X to replace with its restricted set of alternatives $B(X)$. A singleton will never be selected as it is equal to its restricted set of alternatives. If X is confined, then $B(X)$ is also confined, and if $B(X) \neq \varnothing$, then replacing X by $B(X)$ will give a new set of external subsets. Hence the proposition holds if $B(X) \neq \varnothing$ for all external subsets X encountered during reduction. Thus we will assume that we find a external subset with an empty restricted set of alternatives and show that this leads to a contradiction.

First some terminology: if a set X' is obtained from X by any number of replacements, then we say that X is an *ancestor* of X' and X' is a *descendant* of X (a set is its own ancestor and descendant).

Every external subset obtained during reduction is a subset of an external subset of $\mathcal{X}_V$. Thus if X and X^* are unrelated confined external subsets (neither is the ancestor of the other), then they are disjoint and there is a confining set of X that does not intersect all the confining sets of X^* (since if the relation graph of the confining sets of X and X^* were a clique, there would, by Proposition 5, be a vertex contained in all of them, hence in $X \cap X^*$). So there is a confining set of X that does not intersect X^*.

Now we assume that X is the first external subset obtained during reduction with $B(X) = \varnothing$, and find a contradiction. We can assume that $|X| \geq 2$ and X is confined. Thus $B(X)$ is equal to the intersection of the confining sets of X and the essential sets of X. As $B(X) = \varnothing$, by Proposition 5, two of these sets, say V_1 and V_2, are disjoint (and V_1 and V_2 must be essential sets of X, as X is a subset of each of its confining sets). Let G be the relation graph of all the in-critical sets of D. Apply Lemma 9 with $u = V_1$ and $v = V_2$ (W_1 is non-empty as it includes all the confining sets of X): we obtain a graph $H = G - (W_1 \cup W_2)$ such that V_1 and V_2 are in separate components of H, say H_1 and H_2.

We shall find a path in H from H_1 to H_2, a contradiction. We need the following result.

Assertion 10 *Suppose that an external subset Y is replaced by $B(Y)$ during reduction and that P is an essential but not confining set of Y. Then either*
- *$P \cap Z \neq \varnothing$, or*
- *there is an external subset T that was replaced by $B(T)$ earlier during reduction (i.e., before Y is replaced by $B(Y)$), and $P \cap T \neq \varnothing$ but $P \cap B(T) = \varnothing$.*

Proof: Let Y' be the ancestor of Y in $\mathcal{X}_V$. As P is not a confining set of Y, it is not a confining set of Y'. Thus P was covered by another external subset in $\mathcal{X}_V$ (else $Y' \neq B(Y')$, contradicting that $\mathcal{X}_V$ is consistent). If this was Z, then $P \cap Z \neq \varnothing$. Otherwise suppose an external subset W covered P. When Y is replaced by $B(Y)$, no descendant of W covers P. Thus at some point before Y was replaced, a descendant of W that does cover P was replaced by its restricted set of alternatives that does not cover P. Let this descendant be T. □

We use the assertion to find a sequence $X_1 P_1 X_2 \cdots X_r P_r$ where,
- $X_1 = X$ and $P_1 = V_1$;
- for $1 \leq j \leq r$, X_j is a confined external subset, P_j is an essential but not confining set of X_j;
- for $1 \leq j \leq r-1$, P_j is covered by X_{j+1} but not by $B(X_{j+1})$;
- $P_r \cap Z \neq \varnothing$;
- sets that are not consecutive in the sequence are disjoint.

The first two terms of the sequence are given. When the first $2\,j$ terms, $X_1 P_1 X_2 \cdots X_j P_j$, are known, apply Assertion 10 with $Y = X_j$ and $P = P_j$. If $P_j \cap Z \neq \varnothing$, then the sequence is found. Otherwise let $X_{j+1} = T$. Since X was the first external subset encountered with an empty restricted set of alternatives, $B(X_{j+1}) \neq \varnothing$ and is confined. As $B(X_{j+1})$ does not cover P_j, X_{j+1} must have an essential set that does not cover P_j : let this set be P_{j+1}. Note that P_{j+1} is not a superset of X_j and thus not a confining set of it. We must show that X_{j+1}

and P_{j+1} are each disjoint from the sets they are not consecutive o in the sequence. By the choice of P_{j+1} and X_{j+1}, $P_j \cap P_{j+1} = \varnothing$; and X_{j+1} is unrelated to X_j so they are disjoint. Let Q_j and Q_{j+1} be disjoint confining sets of X_j and X_{j+1}, respectively. Then $P_{j+1} \cap X_j = \varnothing$, else $\{Q_j, P_j, Q_{j+1}, P_j\}$ induces a 4-cycle in G. If $j > 1$, we must show that, for $1 \leq j' \leq j-1$, X_{j+1} and P_{j+1} do not intersect $X_{j'}$ or $P_{j'}$. Apply Lemma 9 with $u = P_j$ and $v = P_{j-1}$ (W_1 is not empty as it includes all the confining sets of X_j) : we obtain a graph $J = G - (W_1 \cup W_2)$ such that P_j and P_{j-1} are in separate components of J, say J_1 and J_2. As Q_{j+1} covers P_j and does not intersect X_j, it must be in J_1. For $1 \leq j' \leq j-1$, let $Q_{j'}$ be a confining set of $X_{j'}$ that does not intersect X_j. Note that $P_{j-1}Q_{j-1}P_{j-2} \cdots P_1Q_1$ is a path in G and must be in J_2 since none of these in-critical sets intersect X_j. Thus, for $1 \leq j' \leq j-1$, P_{j+1} and Q_{j+1} (and so X_{j+1}) are disjoint from $P_{j'}$ and $Q_{j'}$.

Every time in the construction above we find a set X_j that is replaced by its restricted set of alternatives $B(X_j)$ during reduction earlier than X_{j-1} was replaced by $B(X_{j-1})$. Therefore after a finite number of steps the sequence must end with a suitable P_r.

Once the sequence is found, let Q'_j, $2 \leq j \leq r$, be a confining set of X_j that does not intersect $X = X_1$. Then $P_1Q'_2P_2 \cdots Q'_rP_r$ must be a path in H_1. Thus we have found an in-critical set P_r in H_1 that intersects Z.

Use the same argument to find an in-critical set W in H_2 that intersects Z (find a path from V_2). If $|Z| \geq 2$, then there is a confining set U of Z that does not intersect every confining set of X. Thus $P_{r-1}UW$ is a path in H from H_1 to H_2. If $|Z| = 1$, then $P_{r-1} \cap W \supseteq Z$. Hence $P_{r-1}W$ is an edge in H. This final contradiction completes the proof of Proposition 8. □

6 Running Time of the Algorithms

The algorithms StableSubsets and MinimumSet contain three consecutive steps that must be taken to obtain a minimum size transversal.

1. Find a minimal set of external subsets.
2. Find a stable set of external subsets (L1).
3. Find a minimum size set of k-good external vertices (L4).

We will show that Step 2 is the bottleneck and determine the overall running time.

Let $S(n)$ be the complexity of an algorithm that checks whether or not a set is k-good.

The complexity of Step 1 is $O(n\,S(n))$: for each of the n external subsets $\{v_i\} \in \mathcal{T}$, we check whether $\mathcal{T} \setminus \{\{v_i\}\}$ is a set of external subsets.

Note that each **while** loop in one of the algorithms has two consecutive steps : checking the loop condition and, possibly, performing the loop content. To find the complexity of a loop, it is necessary to find the complexity of each step and how many times the algorithm may cycle through the loop. It takes time $O(n\,S(n))$ to check the loop condition of L3 : for each $Z \in \mathcal{T}$ and each vertex $v \in Z$, we must see if v is an alternative to Z, that is, if $(\mathcal{T} \setminus \{Z\}) \cup \{\{v\}\}$ is a set of external subsets. The content of L3 takes constant time, and the algorithm may pass through L3 at most n times within each run through L2 (as during each pass through L3 at least one vertex is removed from the sets of external subsets). Thus L3 has complexity $O(n^2\,S(n))$. It takes time $O(n\,S(n))$ to check the loop condition of L2 : we check whether $\mathcal{T} \setminus \{X\}$ is a set of external subsets for each $X \in \mathcal{T}$. The content of L2 is the loop L3 and, within each run through L1, the number of times the algorithm may consider the loop conditions for L2 and L3 is, in total, n (since both loops have an operation that removes vertices). Thus L2 has complexity $O(n^2\, S(n))$. The loop condition of L1 is checked in time $O(n^2\, S(n))$: for each vertex v and each external subset $Y \in \mathcal{T}$, we must see if v is an alternative to Y, that is, if $(\mathcal{T} \setminus \{Y\}) \cup \{\{v\}\}$ is a set of external subsets. The content of L1 is loop L2, and the algorithm may enter L1 at most n times (as the number of external subsets is reduced each time). Thus the complexity of L1 is $O(n^3\, S(n))$.

The loop condition for L4 has complexity $O(n)$, while the content takes constant time; the loop condition for L5 has complexity $O(n\, S(n))$, with the content again taking constant time. Loops L4 and L5 together are checked at most n times. So the third step has complexity $O(n^2\, S(n))$.

Thus the running time of the algorithms is $O(n^3\, S(n))$.

7 Non-uniform Cost and Demand Functions

In this section, we consider a generalised External Network Problem for arc-connectivity in digraphs. Let $c : V \longrightarrow \mathbb{R}$ be a *cost function* and let $d : V \times V \longrightarrow \mathbb{N}$ be a *demand function.* A set of external vertices is *good* if for each pair of vertices $u, v \in V$ there are $d(u, v)$ arc-disjoint (internal or external) paths from u to v. The cost of a set of vertices is the sum of its members' costs. The generalised External Network Problem is to find a minimum cost set of good external vertices.

Theorem 11 *The problem of finding a minimum cost good set of vertices for a digraph is NP-hard if either the cost function or the demand function is non-uniform*

It is helpful to consider the analogous generalisation of the Source Location Problem. Let $c' : V \longrightarrow \mathbb{R}$ be a cost function and let $d^+ : V \longrightarrow \mathbb{N}$ and $d^- : V \longrightarrow \mathbb{N}$ be demand functions. The generalised Source Location Problem is to find a minimum cost set of sources such that, for each $v \in V$, there are $d^+(v)$ arc-disjoint paths from v to the set of sources and $d^-(v)$ arc-disjoint paths from the set of sources to v. In [4, Corollary 1], it is shown that the generalised Source Location Problem for arc-connectivity in digraphs is NP-hard if the cost function or either of the demand functions is non-uniform. We can use this result directly to prove one case of Theorem 11.

Proof of Theorem 11 : The Source Location Problem with uniform demands k is equivalent to the External Network Problem with uniform demand k (we argued this in Section 1; the possibility that costs are not uniform does not make any difference). Thus a solution to the Source Location Problem with non-uniform costs can be obtained from a solution to the External Network Problem with non-uniform costs. As the former is NP-hard [4, Corollary 1] so is the latter.

Now suppose that the cost function is uniform. Let $H = (U, E)$ be a hypergraph where the hyperedges are not mutually disjoint. Let $D = (V, A)$ be a digraph with $V = U \cup E \cup \{x\}$ and $A = \{\, (u, e) \mid u \in e \in E \,\} \cup \{\, (e, x) \mid e \in E \,\}$. Let d be the demand function for D where $d(e_i, e_j) = 1$ for all $e_i, e_j \in E$, $i \neq j$, and all other demands are 0. Note that the union of a transversal of H and $\{x\}$ is a good set for D. We will show that such a set has minimum size. Thus we will have shown that the NP-hard problem of finding the size of a minimum transversal of a hypergraph can be reduced to the problem of finding a minimum size good set of external vertices for a digraph with a non-uniform demand function (except, possibly, if the hyperedges are disjoint, but, in this case, finding a minimum transversal is trivial).

Let X_V be a minimum size good set for D. The demand function requires only that there are paths in both directions between each pair in E. As there are no internal paths joining these vertices, X_V is a good set if and only if, for each $e \in E$, there is an internal path from X_V to e and an internal path from e to X_V (possibly $e \in X_V$ and the paths both have length zero). For each vertex $v \in X_V$, if there is an internal path from v to e, then v is an *in-cover* for e; if there is an internal path from e to v, then v is an *out-cover* for e. So X_V is good if and only if it contains an in-cover and an out-cover for each $e \in E$. In this proof, we will claim twice that X_V can be modified by removing and adding different vertices and that it remains good. The veracity of these claims can be established by checking that whenever we remove a vertex v from X_V, all the vertices of E for which v is an in-cover or an out-cover remain covered by some other vertex in X_V.

For each $e \in E$, the only arc from e joins it to x and there are no arcs leaving x. Thus either e or x must be in X_V to provide an out-cover for e. If x is not in X_V, then $E \subseteq X_V$. In this case, it is possible to find a good set of equal size that does contain x: there are at least two hyperedges e_1 and e_2 with non-empty intersection containing, say, u, so $X_V \setminus \{e_1, e_2\} \cup \{u, x\}$ is also good.

We now assume that $x \in X_V$. If $e \in X_V$, then we can choose a vertex $u \in e$ and $X_V \setminus \{e\} \cup \{u\}$ is also good so we can further assume $X_V \cap E = \varnothing$. Thus X_V must contain at least one vertex $u \in e$ for each $e \in E$ (to provide an in-cover for e). That is, X_V must contain a transversal of H. So the union of a transversal of H and $\{x\}$ is a minimum size good set. □

8 Vertex-connectivity Requirements

In this section, we consider only undirected graphs. The corresponding problems for directed graphs have not been studied.

As in Section 2, for a graph $G = (V, E)$, the set of external vertices is a subset of V denoted X_V. A set of external vertices is *k-vertex-good* and G is *externally-k-connected* if each pair of vertices is joined by k vertex-disjoint (internal or external) paths.

Notice that this definition implies that a k-vertex-good set of external vertices for a graph is a set of vertices such that each pair of vertices in the graph remains connected (using external paths if necessary) when fewer than k vertices are removed from the graph. Consider an alternative definition: a set of external vertices X_V is *k-vertex-good* if each pair of vertices remains connected (possibly only by external paths) when fewer than k vertices *not in X_V* are removed from the graph. That is, we suppose that a vertex chosen as an external vertex not only gains the ability to communicate with other external vertices but also becomes indestructible. When the Source Location Problem with vertex-connectivity requirements was studied (see, for example, [3]), definitions analogous to this alternative definition were used and the problem of finding smallest possible sets of sources was shown to be NP-hard if the required external connectivity is more than 2. We believe our original definition is more natural and we conjecture that there exist polynomial algorithms to find a k-vertex-good set of external vertices for any k. In this section we show that the algorithms exist for $k \leq 3$.

Theorem 12 *There exist polynomial algorithms to find minimum size k-vertex-good sets for $k \leq 3$.*

Proof : We shall not give explicit algorithms. In each case, we will show that if a graph is decomposed in a certain way, then it can be seen that a set is k-vertex-good if and only if it intersects particular components of this decomposition. Polynomial algorithms for finding these decompositions are well-known (see, for example, [1] if necessary). Bear in mind that if a graph is k-connected, then the empty set is a k-vertex-good set.

Case 1 : $k = 1$.
If G is not connected, then clearly a set is 1-vertex-good if and only if it contains a vertex from each connected component, and a minimum size set contains precisely one vertex from each component.

Case 2 : $k = 2$.
Recall that a *block* of a graph G is either a maximally 2-connected subgraph or a bridge and that a vertex that belongs to more than one block is a cutvertex. Let $B(G)$ be the graph that has the blocks and cutvertices of G as its vertices with edges joining each cutvertex to the blocks that contain it. Recall that $B(G)$ is a forest. Suppose that G is not 2-connected (and contains more than 2 vertices) and so contains more than one block. A block β is an *end-block* if it has degree 1 in $B(G)$; that is, it contains only one cutvertex. This cutvertex is contained in every internal path joining vertices in β to other parts of the graph. So if a set is 2-vertex-good it is necessary for it to contain one vertex from each end-block and also, obviously, two vertices from each block that has degree 0 in $B(G)$. It is easy to check that this is also sufficient.

Case 3 : $k = 3$.
We require a characterisation of 2-connected graphs due to Tutte [6]. We need some further definitions. A *bond* is a pair of vertices joined by 3 or more parallel edges. A *cleavage unit* is a graph that is either 3-connected, a cycle or a bond. If edges are added to a graph G, the *augmented graph* obtained is called G^a, and the edges of G are called *real*, the additional edges are *virtual.*

For any 2-connected graph G, there is an augmented graph G^a such that a collection of cleavage units can be obtained where

- each real edge is in exactly one cleavage unit, and
- each virtual edge is in exactly two cleavage units.

Furthermore, if G^a is obtained using as few virtual edges as possible, then the graph that has the cleavage units as vertices and whose edge set is the set of virtual edges — each virtual edge joins the pair of cleavage units that contain it — is a tree. It is called the *cleavage unit tree* of G and is denoted $T(G)$. Tutte [6] showed that for any 2-connected graph G, $T(G)$ is unique and no edge of $T(G)$ joins two cycles or two bonds. Any pair of vertices joined by a virtual edge is *a hinge.* Vertices in a hinge belong to more than one cleavage unit and other vertices belong to only one. If cleavage units are separated in $T(G)$ by the removal of a virtual edge, then they are separated in G by the removal of the corresponding hinge. If more than two cleavages units contain the same hinge, then exactly one of them is a bond, and in $T(G)$ the bond is adjacent to all the other units that share the hinge. Thus in a cleavage unit that is not a bond there are no parallel virtual edges.

We shall need a technical result, Lemma 13, which requires some further definitions. Let p, q, r and s be (not necessarily distinct) vertices in a 2-connected graph G. We call a set of paths with specified ends vertex-disjoint if they intersect only, if at all, at their ends. A set of 3 vertex-disjoint paths in G^a that join each of p, q and r to s is called a *pqrs*-pathset. Let Π be a *pqrs*-pathset. When we say that a vertex or edge is in Π, we mean that it is in one of the three paths. Each virtual edge $e = uv$ in Π belongs to 2 cleavage units. We will call one of these *used* and the other *unused.* If exactly one of these two units contains real edges of Π, then it is the used unit. Otherwise we can choose which is used and which unused. Let $S(\Pi)$ be a set of cleavage units that includes each cleavage unit that contains a real edge of Π and the used cleavage unit of each virtual edge. Let $S^*(\Pi)$ be the smallest set of cleavage units that contains $S(\Pi)$ and that induces a connected component of $T(G)$. (To see that $S(\Pi)$ might not be connected in $T(G)$ even though Π is connected in G, suppose that $e = uv \in U$ and $e' = vw \in U'$ are adjacent edges in a path in Π, and that $\{v, x\}$ is a hinge shared by U and U', $x \notin \{u, w\}$. Then U and U' will not be adjacent in $T(G)$ if there is a bond containing $\{v, x\}$.)

If $S^*(\Pi)$ does not contain any unused cleavage unit of a virtual edge in Π, then Π is *realisable.* Note that if Π is realisable, then for each virtual edge e in Π, in $T(G)$, e joins its unused cleavage unit to $S^*(\Pi)$, and so no two virtual edges in Π have the same unused cleavage unit (as this would allow us to find a cycle in $T(G)$).

Lemma 13 *Let Π be a realisable pqrs-pathset for a 2-connected graph G.*
(1) Then there is a pqrs-pathset Π^* for G that contains no virtual edges.
(2) Moreover, if Π does not contain a virtual edge e^* that has one end incident with $S^*(\Pi)$ in $T(G)$, then Π^* contains no edges from cleavage units separated from $S^*(\Pi)$ in $T(G)$ by e^*.

Proof : To prove (1), we will first show that we can replace a virtual edge of Π to obtain another realisable *pqrs*-pathset. Then we will show that if we continue to replace virtual edges in this way, we will eventually obtain a *pqrs*-pathset containing no virtual edges.

Suppose that a path $\pi \in \Pi$ contains a virtual edge $e = uv$. Let U be the unused cleavage unit of e. Find a path ρ from u to v in U other than e; by definition a cleavage unit is 2-connected or a bond so this path can be found (possibly it is simply an edge parallel to e). Substitute ρ for e in π to obtain a new pathset Π'. Note that as e separates U from $S^*(\Pi)$ in $T(G)$, no cleavage unit in $S^*(\Pi)$ contains a vertex of $U \setminus \{u, v\}$. Therefore the paths of Π' are also vertex-disjoint and Π' is a *pqrs*-pathset. Also, as each unused cleavage unit U' of a virtual edge $e' \neq e$ in ρ is adjacent to U in $T(G)$, Π' is realisable.

Note that each time we obtain a new realisable *pqrs*-pathset Π' in this way, the set $S^*(\Pi')$ contains one more cleavage unit than $S^*(\Pi)$. As, for each virtual edge in the pathset, there is a cleavage unit not in $S^*(\Pi')$, we must eventually obtain a pathset containing no virtual edges.

To prove (2) note that each new edge we add to the pathsets belongs only to cleavage units not in $S^*(\Pi)$ so we will never add e^*. Thus no cleavage unit separated from $S^*(\Pi)$ by e^*

can ever become an unused cleavage unit and so we will never add edges from these cleavage units. This completes the proof of Lemma 13. □

Now we show how to find a 3-vertex-good set for a graph. We divide the problem into two cases according to whether or not the graph is 2-connected.

Case 3a: $k = 3$ and G is 2-connected.
A cleavage unit of G that has degree 1 in $T(G)$ and is not a bond is called an *end-unit* of G. The vertices of an end-unit not in its unique hinge can be disconnected from the rest of the graph by removing the two vertices of the hinge. Thus a 3-vertex-good set for G must contain a vertex from each end-unit. In fact, we claim that a set X_V is a minimum size 3-vertex-good set for G if and only if it contains each vertex with fewer than 3 neighbours in G and one vertex that is not in the hinge from every end-unit of G. We have already said enough to establish that it is necessary that X_V contain such vertices. To show that it is sufficient, we prove that there are 3 vertex-disjoint paths (possibly external) between each pair of distinct vertices x and y.

Suppose that there is a cleavage unit U that contains both x and y. If U is 3-connected or a bond, then there is a set Π of 3 vertex-disjoint paths in U that join x and y. Let U be the used cleavage unit of each virtual edge in Π. Then Π is a realisable $xxxy$-pathset ($S^*(\Pi) = U$) and, by Lemma 13, there are 3 internal vertex-disjoint paths that join x and y.

Suppose U is a cycle. If neither x nor y is in a hinge, then they must have degree 2 in G and therefore are both external vertices. If x and y belong to the same hinge in U, then there is another cleavage unit containing x and y and it is either 3-connected or a bond; the case dealt with above.

In all remaining cases we can assume there is a cleavage unit U that contains x but not y. Let e be the virtual edge in $T(G)$ that is incident with U and whose removal separates U from every cleavage unit that contains y, while $\{u, v\}$ is the hinge incident with e. Let U' be a cleavage unit that contains y. Let e' be the virtual edge in $T(G)$ that is incident with U' and whose removal separates it from U, while $\{p, q\}$ is the hinge incident with e'. Finally, let C_x, C_y and C_π be the 3 components of $T(G) \setminus \{e, e'\}$ where $U \in C_x$ and $U' \in C_y$. Since G is 2-connected, there exist two vertex-disjoint paths with distinct ends, say π_1 and π_2, that join $\{u, v\}$ to $\{p, q\}$. Note that π_1 and π_2 contain only edges from cleavage units in C_π.

To find 3 vertex-disjoint paths that join x and y we will find 3 vertex-disjoint paths in C_x that join x to each of u, v and an external vertex. Using the same method we can find 3 vertex-disjoint paths in C_y that join y to each of p, q and an external vertex. Pasting all these paths together we obtain 3 vertex-disjoint paths, one of which is external, that join x and y in G.

In fact, we will find a $uvwx$-pathset Π where $w \in U$ and either w is an external vertex, or it is in a hinge of U and incident with a virtual edge $e'' = \{w, z\}$, $e'' \neq e$. In the latter case, let C' be the component of $C_x \setminus e''$ that does not contain U. Let U'' be an end-unit in C'. Then there is a path in G from the external vertex in U'' to w that contains only real edges from cleavage units in C'. Thus it is sufficient to find vertex-disjoint paths in U from x to each of u, v and w, with the proviso that $w = u$ only if $x = u$. If we let U be the used cleavage unit of every virtual edge in Π, then it is realisable. If we can also show that Π does not contain e or, if w is not an external vertex, e'', then, applying Lemma 13, we obtain 3 paths in C_x that join x to u, v and an external vertex.

There are three cases according to what type of cleavage unit U is. In each case we find Π and show that it does not contain e and, when necessary, e''

Case 3ai: U is 3-connected.
If U is an end-unit, then let w be the external vertex that it contains. If U is not an end-unit, then let $\{w, z\}$ be a hinge in U incident with a virtual edge $e'' \neq e$, $w \notin \{u, v\}$ (such a hinge exists if U is not an end-unit). Let Π contain 3 vertex-disjoint paths in U from x to each of u, v and w. If $x \notin \{u, v\}$, then $e \notin \Pi$ and if $e'' \in \Pi$, then clearly it is the last edge on the path from x to w; simply switch the labels w and z (clearly $z \notin \{u, v\}$ in this case) and use the same path with e'' omitted. If $x = u$, then we require only 2 paths which can be found in $U \setminus e$. Again switch labels if e'' is in the path that joins x to w.

Case 3aii: U is a bond.
We can assume $x = u$. If U contains 2 or more virtual edges, then let $w = x$ and let e'' be a virtual edge joining u and v distinct from e. Let Π contain two paths of zero length and a path from u to v that is a single edge other than e and e''.

If U contains only one virtual edge, then the unique adjacent cleavage unit is a cycle (if it were 3-connected, e would be redundant). Thus $x = u$ has only 2 neighbours in G and it is an external vertex. Let $w = x$ and, again, let Π contain two paths of zero length and an edge from u to v that is not e .

Case 3aiii: U is a cycle.
If x is not a hinge vertex, then x is an external vertex so let $x = w$. Let Π contain vertex-disjoint paths in U that join x to each of u, v and w (the latter has zero length) and $e \notin \Pi$ since $x \notin \{u, v\}$.

If x is in a hinge, then let U'' be one of the other cleavage units that contains that hinge. As U'' is a bond or 3-connected, let $U = U''$ and we have a case already described whether or not $y \in U''$.

Case 3b: $k = 3$ and G is not 2-connected.
Let $B(G)$ be the graph of blocks and cutvertices defined in Case 2 above and for each 2-connected block β, let $T(\beta)$ be the cleavage unit tree of β. To obtain a 3-vertex-good set of external vertices we include vertices to satisfy the following conditions considered in order (so later conditions may already be satisfied when they are considered). We must include

- each vertex with fewer than 3 neighbours in G,
- for each block β, one non-hinge vertex from each end-unit of β unless a non-hinge vertex of the unit is a cutvertex of G,
- one vertex from each 2-connected block that contains only 2 cutvertices,
- two vertices (that are not cutvertices) from each end-block of G that is 2-connected, and
- if G is not connected, three vertices from each connected component C of G, or, if we have $|V(C)| < 3$, every vertex in C.

The necessity of each condition is easily seen. Thus to show that a set X_V is a minimum size k-vertex-good set of external vertices if it contains these vertices, we find 3 vertex-disjoint paths between each pair of distinct vertices x and y.

First suppose that x and y belong to the same block β of G. We know, from the previous case, that if there are not 3 vertex-disjoint paths between x and y, then we can find a set of vertex-disjoint paths that contains 2 internal paths from x to y and paths from x to w and y to z, where w and z are non-hinge vertices in distinct end-units U and U' of β. We can consider U and U' to be given, but w and z can be chosen. If U contains an external vertex, then let that be w. Otherwise let w be the vertex that is a cutvertex of G, and therefore in another block of G. Clearly there is a path from w to an external vertex in an end-block of G that does not contain any edges of β. Thus, however w is chosen, we find a path from x to an external vertex. Choosing z in the same way, we obtain an external path between x and y.

Now suppose that x and y do not belong to the same block. Let β be a block that contains x but not y. When we refer to a cutvertex of β we shall mean a vertex of β that is a cutvertex of G. Notice that, by the choice of X_V,

- either x is an external vertex, or
- we can find 3 distinct vertices u, v and w in β such that each is either a cutvertex (possibly x) or an external vertex.

In the latter case, if x and y are in the same connected component of G, then choose one of u, v and w to be the cutvertex that separates x and y. If there are vertex-disjoint paths in β from x to each of u, v and w, then one of these can, possibly, be extended to an internal path to the block that contains y and the others can be extended, perhaps trivially, to paths that end in an external vertex, since if, say, u is a cutvertex, we can find a path from u to an external vertex in an end-block. As the same analysis holds for y, we can find 3 vertex-disjoint paths between x and y.

We must show that if x is not an external vertex, then we can find paths to u, v and w. Let U be a cleavage unit of β that contains x and is not a bond. Choose 3 vertices p, q and r in U such that

- either $p = u$ or p is in the hinge that separates x and u, and
- either $q = v$ or q is in the hinge that separates x and v, and
- either $r = w$ or r is in the hinge that separates x and w.

We can alter the choice of u, v or w so that it is not necessary to choose p, q and r all belonging to the same hinge. Thus we may assume that p, q and r are distinct, except that we might choose up to two of them to be x. If p, q or r is in a hinge, then let the virtual edge incident with that hinge be e_1, e_2 or e_3, respectively. If we can find a reliable $pqrx$-pathset Π that does not contain e_1, e_2 or e_3, then by Lemma 13, we can find paths in G from x to each of p, q and r and these can be extended to paths to u, v and w.

To construct the pathset, we consider two cases.

Case 3bi : U is 3-connected.
Clearly we can find vertex-disjoint paths from x to each of p, q and r. If a path contains, say, $e_1 = pz$, then we can change our choice of p to z to obtain the required pathset.

Case 3bii : U is a cycle.
As x is not an external vertex it must be a cutvertex or in a hinge. If it is a cutvertex, then

we let, say, $p = x$, and the other two paths can be found in U avoiding unwanted virtual edges as in the previous case.

If x is in a hinge, then let e be the virtual edge incident with the hinge. If there are two external or cutvertices in the cleavage units of β separated from U by e in $T(\beta)$, then let them be u and v. Thus we can let $p = q = x$ and the pathset is easily found. It is not possible for there to be no external or cutvertices in these cleavage units since amongst them there must be an end-unit which must contain either an external vertex or a cutvertex. Suppose that there is exactly one external or cutvertex z in these units. Note that x belongs to another cleavage unit U' that is not a bond (otherwise x is an external or a cutvertex) and such that x does not belong to the hinge that separates U' from z. Choosing $U = U'$ we have a case already described.

This completes the proof of Theorem 12. □

References

[1] R. Diestel, *Graph Theory*. Springer-Verlag, New York, 1997.

[2] J. Hao and J. B. Orlin, *A faster algorithm for finding the minimum cut in a graph*. J. of Algorithms, **17** (1994), 424–446.

[3] H. Ito, M. Ito, Y. Itatsu, H. Uehara, and M. Yokohama, *Location problems based on node-connectivity and edge-connectivity between nodes and node-subsets*. Lecture Notes in Computer Science, **1969** (2000), 338–349.

[4] H. Ito, K. Makino, K. Arata, S. Honami, Y. Itatsu, and S. Fujishige, *Source location problem with flow requirements in directed networks*. Optimization Methods and Software, **18** (2003), 427–435.

[5] H. Nagamochi, T. Ishii, and H. Ito, *Minimum cost source location problem with vertex-connectivity requirements in digraphs*. Info Process Letters, **80** (2001), 287–294.

[6] W. T. Tutte, *Connectivity in Graphs*. University of Toronto, Toronto, 1966.