Relay placement for two-connectivity

Gruia Calinescu
Illinois Institute of Technology

Previously presented at IFIP Networking 2012

September 2013

Outline

1. Problem formulation and examples
2. Previous work and today's results
3. Approximation algorithm
4. Conclusions and open problems

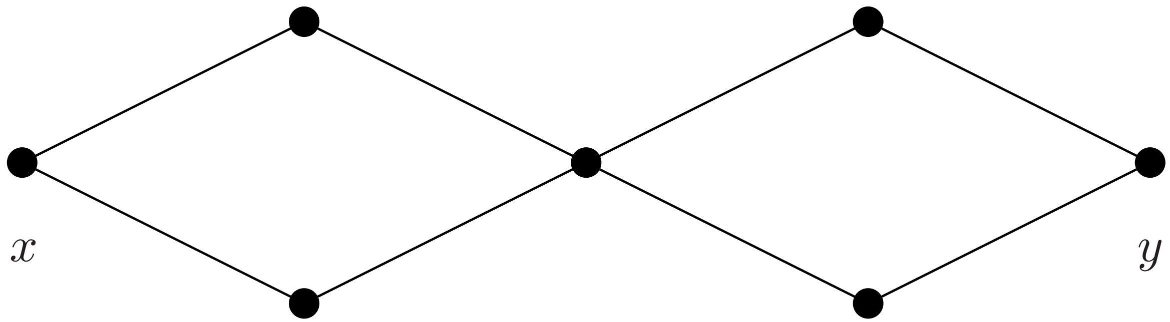
Relay Placement

Input: multiset S in a normed space (representing already deployed sensors). 2D or 3D, can place relay anywhere.

Output: minimum size multiset Q such that $U(\cup S)$ two connected, where $U(P)$ is the unit-disk graph induced by P. Q represents the relay nodes.

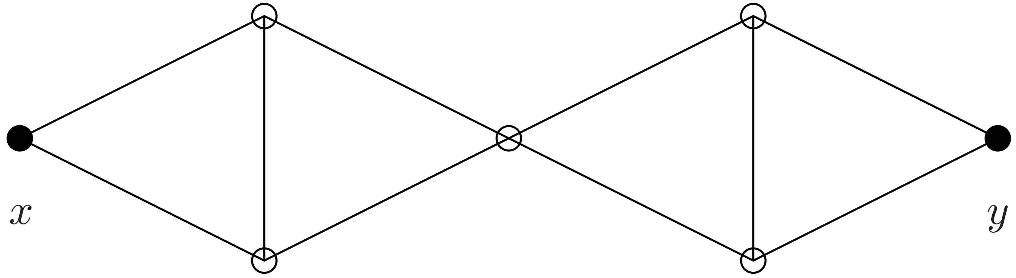
The unit-disk graph of a set of points has an edge between two points if their distance is at most 1 (normalize to 1 the transmission range of sensors).

First variant: two-edge-connectivity



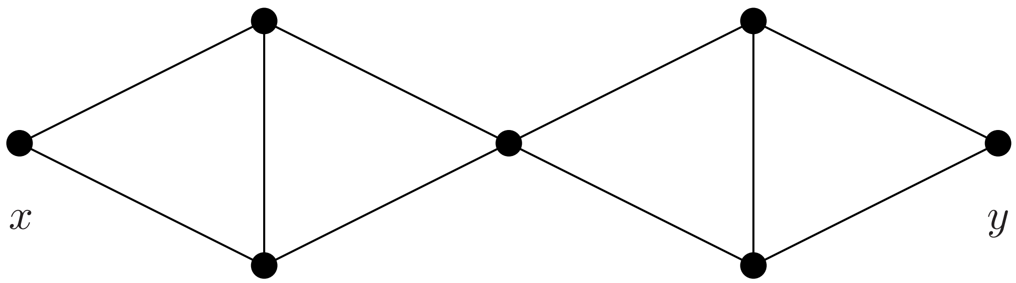
Required: have between any two nodes two edge-disjoint paths (as in the figure, for x, y). Idea: protection against edge failures.

First variant: two-edge-connectivity



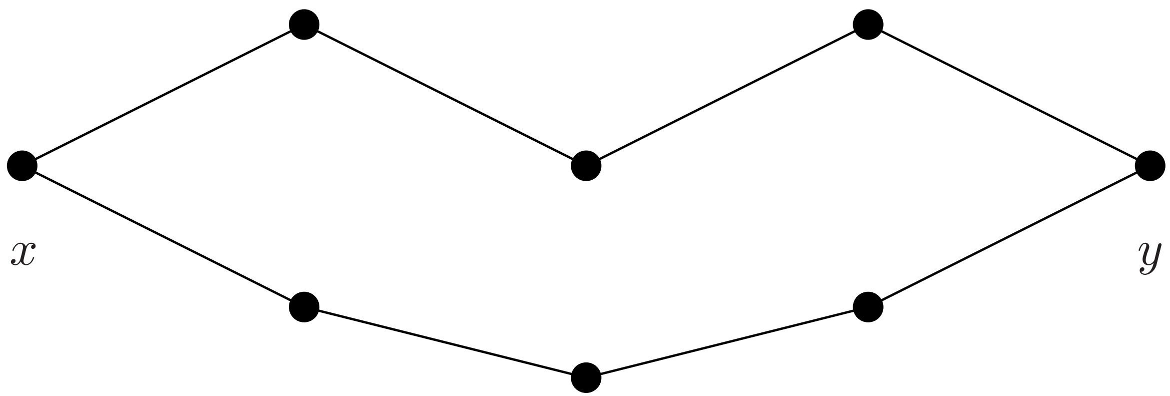
$U(S \cup Q)$ must allow between any two *sensors* two edge-disjoint paths (in the figure, for x, y are the sensors; empty circles are relays).

First variant: two-edge-connectivity



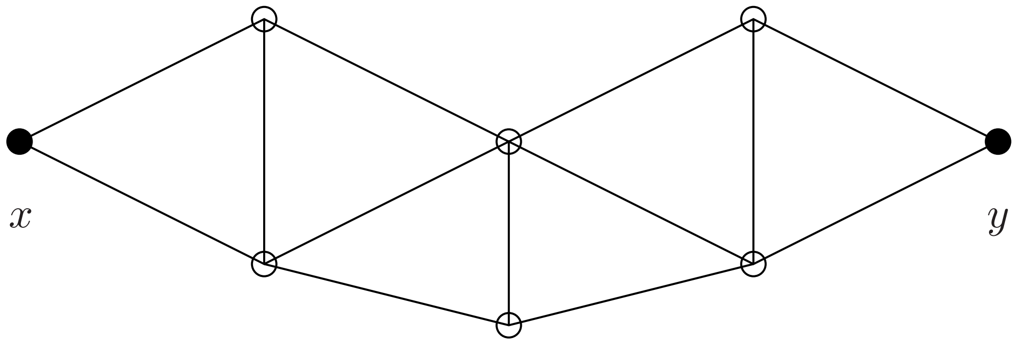
Same problem (requires proof): $U(S \cup Q)$ must allow between any two *nodes* two edge-disjoint paths (in the figure, for x, y are the sensors).

Second variant: two-vertex-connectivity



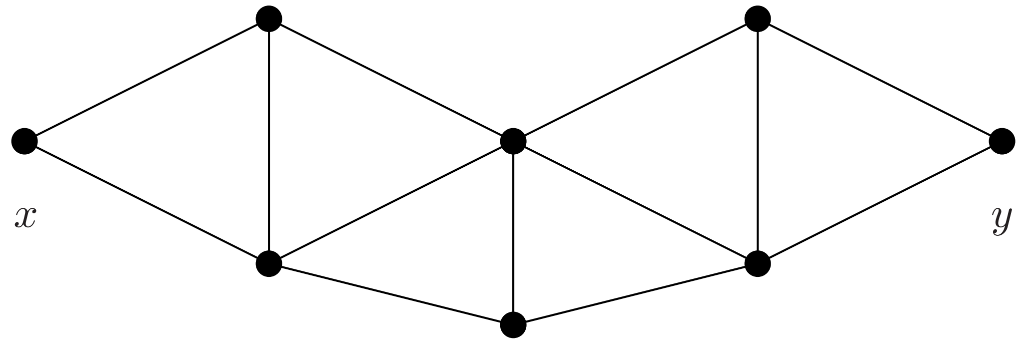
Requirement: have between any two nodes two internally node disjoint paths (as in the figure, for x, y). Idea: protection against node failures.

Second variant: two-vertex-connectivity

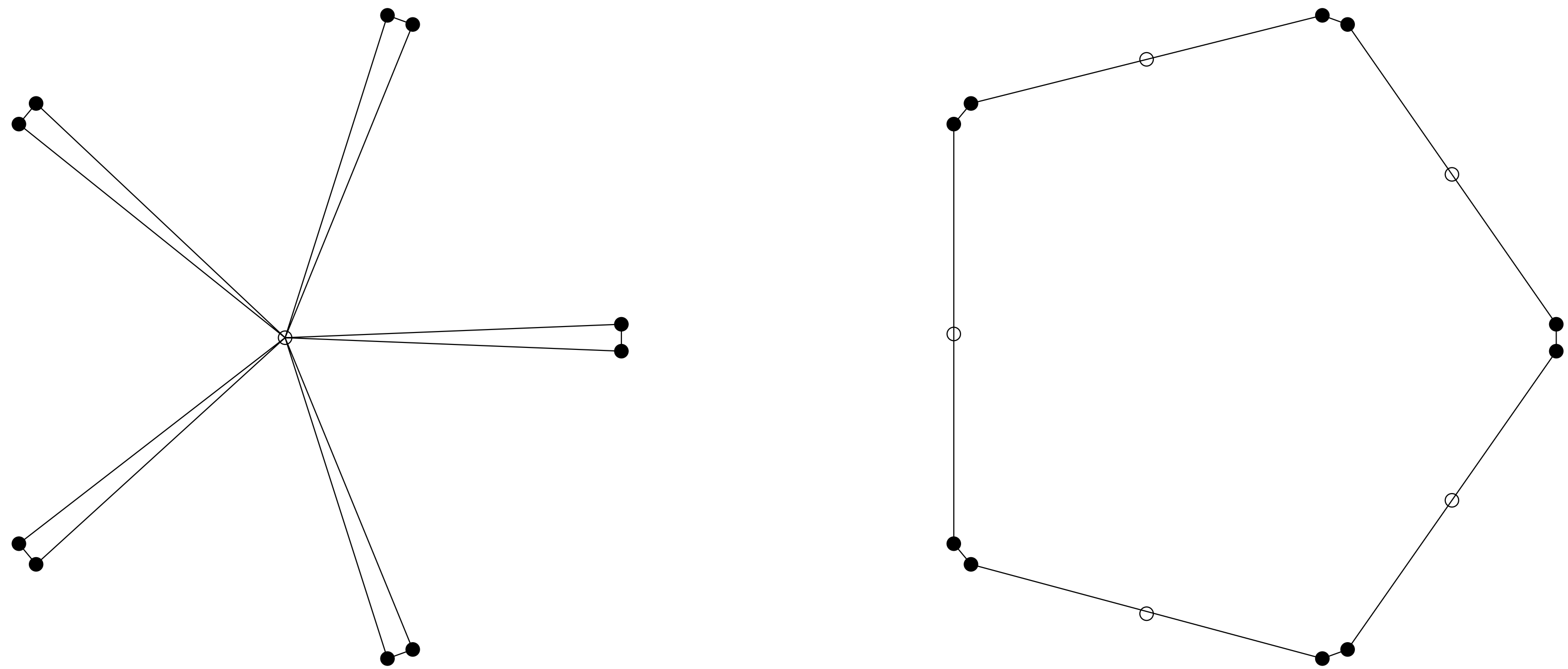


$U(S \cup Q)$ must allow between any two *sensors* two internally node disjoint paths (in the figure, for x, y are the sensors; empty circles are relays).

Second variant: two-vertex-connectivity



Same problem (requires proof): $U(S \cup Q)$ must allow between any two *nodes* two internally vertex disjoint paths (in the figure, for x, y are the sensors).



LEFT: OPT for two-edge-connectivity. The nodes of S are black disks, and the relay nodes are empty circles. RIGHT: an optimum bead solution.

Two-vertex-connectivity (*biconnectivity*): *opt* $= 2$.

Bead Solutions

Q bead-solution if $U(Q \cup S)$ contains a two-edge-connected graph (or biconnected, respectively) H where each node of Q has degree exactly two.

To construct a bead solution: get a two-edge-connected (or biconnected, respectively) graph H on S; let $w(x, y) = \max(0, \lceil ||x, y|| \rceil - 1)$, where $||u, v||$ is the distance from u to v. Place $w(x, y)$ relay nodes to connect x to y for every edge of H.

Previous results

NP-Hard for simple connectivity. $O(k^4)$ approximation for k-connectivity (polynomial-time algorithm with output with the number of relays at most $10k^4$ times the optimum).

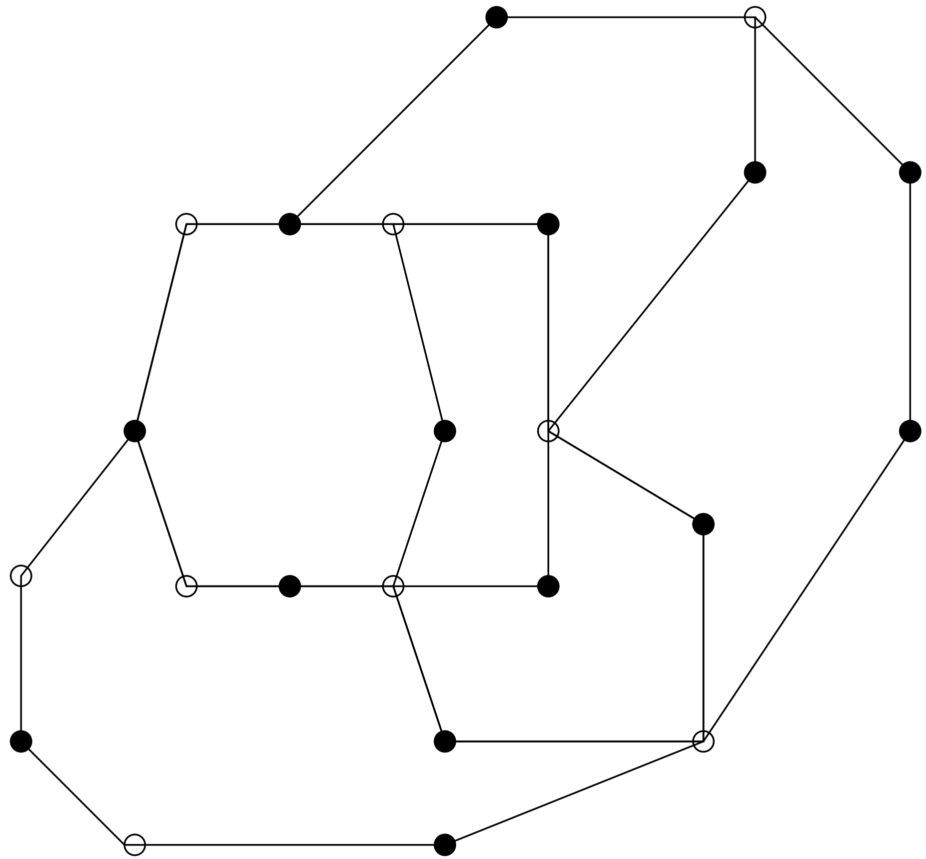
(INFOCOM 2006): algorithm for two-edge connectivity and algorithm for biconnectivity. Approximation ratio of at most $2d_{MST}$, where d_{MST} is the maximum degree of a minimum degree Minimum Spanning Tree in the normed space.

New results: Ratios of d_{MST} for biconnectivity and $2d_{MST} - 1$ for two-edge-connectivity respectively.

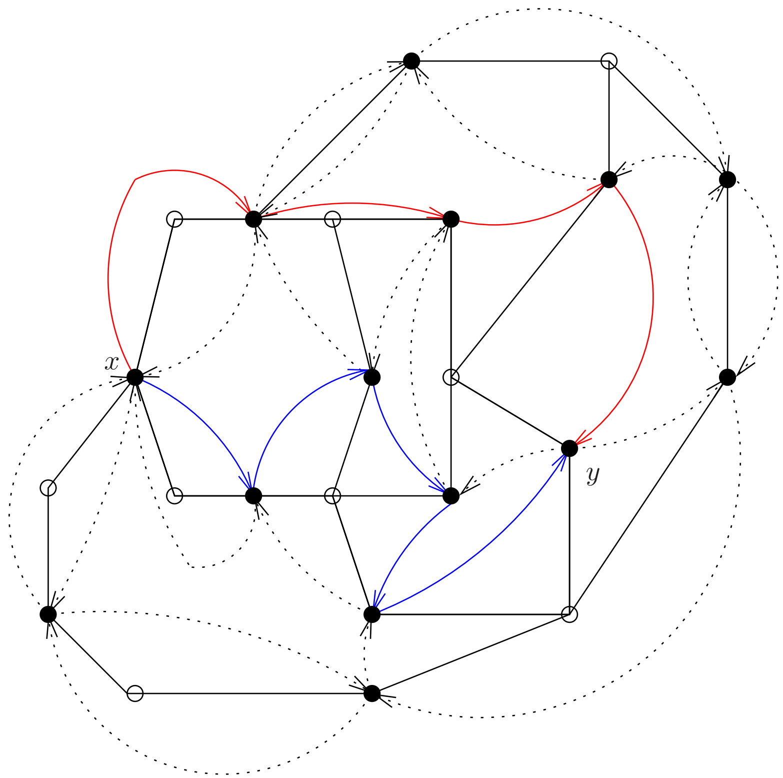
Bypass Steiner vertices (parsimony)

Let H be a biconnected planar undirected graph, and replace every edge by two anti-parallel directed arcs. Let S be a subset of $V(H)$. Then there exists a set of arc-disjoint paths P_i of H, all starting and ending at a vertex of S and without interior vertices from S, such that, if we replace each P_i by an arc e_i joining the start and end vertex of P_i, we obtain a biconnected digraph on S. Non-planar: "fractional outconnected" variant.

Bypass Steiner vertices (parsimony)



S: black disks; Steiner vertices: circles.



Planar graph: black solid edges. S: black disks; Steiner vertices: circles. Paths P_i: all the arcs. Two internally vertex disjoint $x-y$ paths: red, blue.

Related Work

In 2-D space, $d_{MST} = 5$. In 3-D space, $d_{MST} = 13$.

MINIMUM STEINER POINTS TREE: given S in the plane, find minimum Q such that $U(S \cup Q)$ is connected.

Lin and Xue 1999: NP-Hard. Ratio of 5 for bead-MST.

Mandoiu and Zelikovsky (2000): bead-MST, ratio of 4, and $d_{MST} - 1$ in normed spaces. We use this.

Cheng, Du, Wang, and Xu (2008): randomized 2.5

Arbitrary normed spaces, Nutov and Yaroshevitch (2009): $\lfloor (d_{MST} + 1)/2 \rfloor + 1 + \epsilon$-approximation.

Two-edge-connectivity

KKS algorithm: (recall) $w(x,y) = \max(0, \lceil ||x,y|| \rceil - 1)$. Note: w is not a metric. If $w(x,y) > 0$, allow parallel edges of weight w between x and y; otherwise allow only one edge of weight 0, plus parallel edges of weight 1, creating multigraph G.

Use **Algorithm KV** to compute in G a set of edges A, attempting to minimize $w(A)$ while (V,A) is two-edge-connected. Replace each edge of positive weight by new beads (that is, every such edge has its own distinct beads).

Algorithm KV is a 2-approximation for MINIMUM WEIGHT SPANNING TWO-EDGE-CONNECTED SUBGRAPH.

Inside Algorithm KV

Replace each edge xy by two directed arcs xy and yx, of the same weight, creating a digraph $\overrightarrow{G}$. (*bidirected* G). Pick in $V(G)$ an arbitrary root r. Use Weighted Matroid Intersection to compute two arc-disjoint r-rooted arborescences A and B such that $w(A) + w(B)$ is minimized.

Output an edge xy if either xy or yx are in $A \cup B$. It is known that this output is two-edge-connected.

We construct in $\overrightarrow{G}$ two arc-disjoint r-arborescences A and B satisfying $w(A) + w(B) \leq 9\mathit{opt}$.

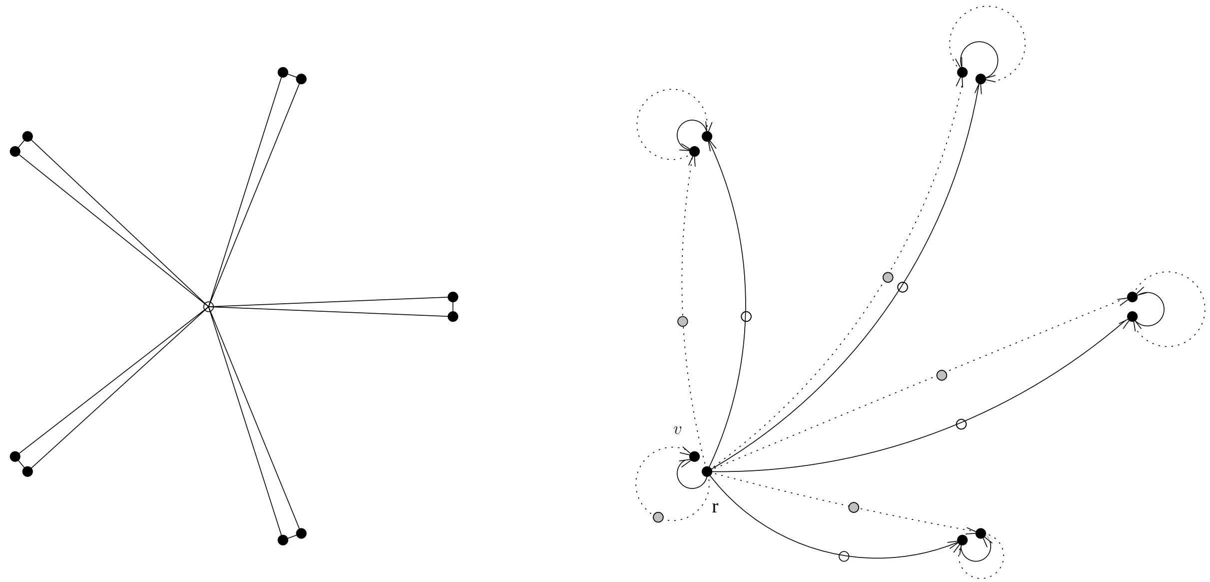
Algorithm KV analysis idea

We construct in $\overrightarrow{G}$ two arc-disjoint r-arborescences A and B satisfying $w(A) + w(B) \leq 9$*opt*.

Use the 4-approximation analysis for the Minimum Spanning Tree algorithm for MSPT, used twice for each arborescence. Instead of 4, $d_{MST} - 1$ in other normed spaces.

Another "+1" is unavoidable, since we may have to add two connectivity between to adjacent sensors.

Next slide shows analysis is tight, assuming redundant relays are not removed.

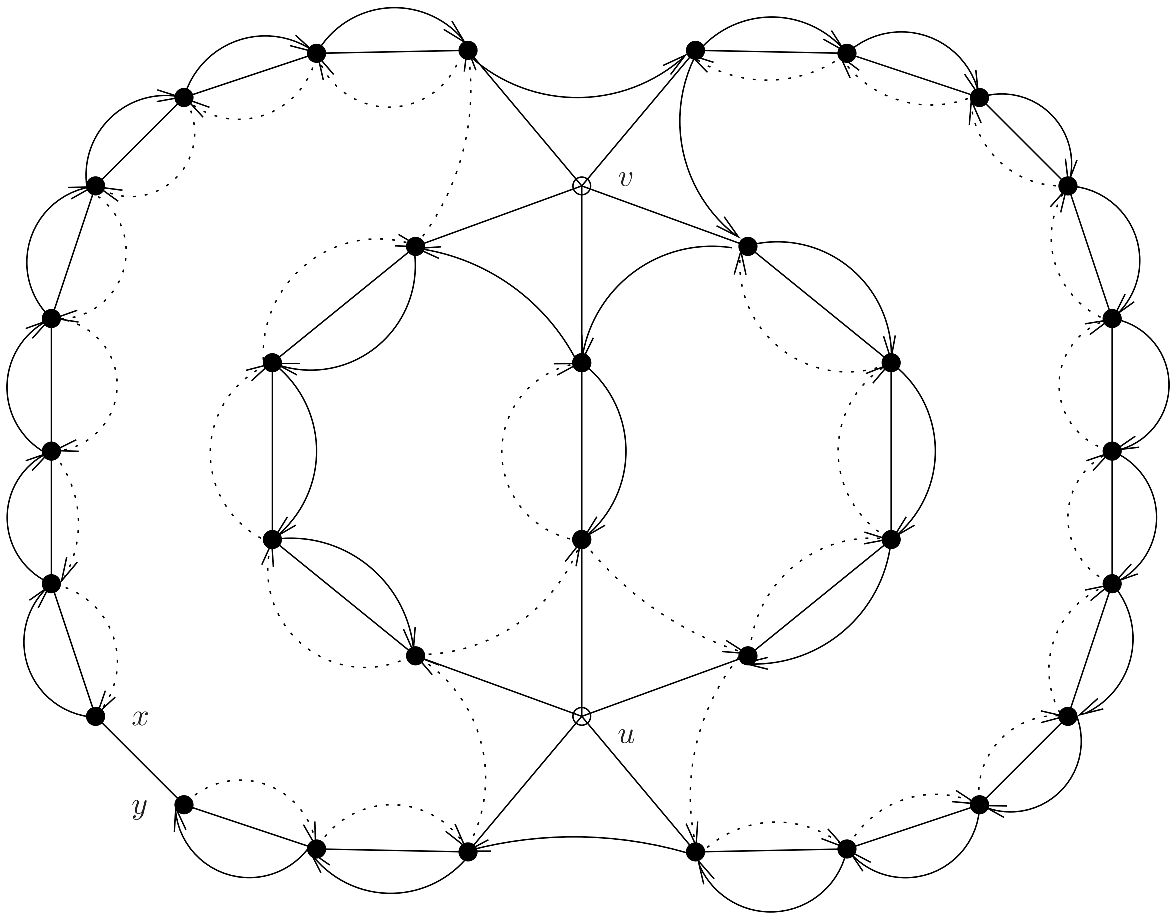


Left: the optimal previous solution. S is given by the black nodes. Right: possible output of **Algorithm KV**, with the two arborescences given by solid and dotted arcs, and beads empty or dotted. One arc of weight $w = 1$ from r to v is required by one of the two arborescences.

Biconnectivity algorithm

Use **Algorithm KR** to compute in G a set of edges A, attempting to minimize $w(A)$ while (V, A) is biconnected. Replace each edge of positive weight by new beads (that is, every such edge has its own distinct beads).

Algorithm KR is a 2-approximation for MINIMUM WEIGHT SPANNING BICONNECTED SUBGRAPH.



v
u
x
y

Biconnectivity analysis tight

The nodes of S are black disks. Optimum uses the relay nodes u and v. If we start **Algorithm KR** with x and y as in the figure, ten edges of weight one would be chosen by the algorithm (precisely, the arcs passing "around" each of u and v, each arc needing a bead node). The two arborescences from Whitty's Theorem are represented, except for arcs sx and sy, by dotted and solid arcs, respectively. Only nine beads by starting with x, y not U-adjacent. In a larger example, one or two beads saved still results in a ratio of five.

Conclusions

Using variants of previously proposed algorithms, we improved the approximation ratio of TWO-CONNECTED RELAY PLACEMENT for biconnectivity from $2d_{MST}$ to d_{MST}, and for two-edge-connectivity from $2d_{MST}$ to $2d_{MST} - 1$. Euclidean 2-D space: $d_{MST} = 5$, in 3-D: $d_{MST} = 13$.

Assuming that no post-processing removes redundant relay nodes, these ratios are tight.

Acknowledgement: this work was supported by NSF grant NeTS-0916743 (ARRA - NSF CNS-0916743).