AN APPROXIMATION ALGORITHM FOR THE MINIMUM-COST K-VERTEX CONNECTED SUBGRAPH

JULY 12, 2001, REVISED: JUNE 10, 2002

JOSEPH CHERIYAN, SANTOSH VEMPALA, AND ADRIAN VETTA

ABSTRACT. We present an approximation algorithm for the problem of finding a minimum-cost k-vertex connected spanning subgraph, assuming that the number of vertices is at least $6k^2$. The approximation guarantee is 6 times the kth harmonic number (which is $O(\log k)$), and also this is an upper bound on the integrality ratio for a standard linear programming relaxation.

1. INTRODUCTION

Let $G = (V, E)$ be an undirected graph, let each edge $e \in E$ have a nonnegative cost c_e, and let k be a positive integer. The *mincost k-VCSS* problem is to find a spanning subgraph H of minimum cost such that H is k-vertex connected. (A graph is called *k-vertex connected* if it has at least $k+1$ vertices, and the removal of any $k-1$ vertices leaves a connected graph.) The problem is NP-hard for $k \geq 2$, and for $k = 1$ it is the minimum spanning tree problem. Our paper addresses the "special case" of the problem where the graph has order $|V| \geq 6k^2$; this too is NP-hard for $k \geq 2$. (So for a fixed k, our method handles all graphs except a finite set of "small" graphs, and our method fails on each of the "small" graphs.) Our approximation guarantee is 6 times the kth harmonic number, which is $O(\log k)$. Also, this is an upper bound on the integrality ratio for a standard linear programming relaxation. Several previous papers have attacked the mincost k-VCSS problem (without restrictions on $|V|$), with the goal of improving on the approximation guarantee (see the references). An approximation guarantee of more than $k/2$ has been presented in [11]; also, an upper bound of $O(k)$ on the integrality ratio was known [4, 5]. Better results were not known for our "special case," but we mention that our results may not improve on previous results for small k ($k = 2, 3, 4, \ldots$). (An $O(\log k)$ approximation guarantee was claimed earlier in [15], but subsequently an error has been found and that claim has been withdrawn; see the erratum of [15].) For more discussion on related problems and results, see the introduction of [3].

Mathematics Subject Classification (2000): Primary: 68W25, 05C40. Secondary: 68R10, 90C27.

Key words and phrases. approximation algorithm, ℓ-critically k-connected graph, linear programming relaxation, network design, k-outconnected graph, vertex connectivity.

Joseph Cheriyan (jcheriyan@math.uwaterloo.ca), Dept. of Comb. and Opt., Univ. Waterloo, Waterloo, ON, Canada, N2L 3G1. Supported in part by NSERC research grant OGP0138432.

Santosh Vempala and Adrian Vetta ({vempala, avetta}@math.mit.edu), Department of Mathematics, M.I.T., Cambridge, MA 02139, USA. Supported in part by NSF Career Award CCR-9875024.

Our algorithm is based on two results: (1) a polynomial-time algorithm of Frank and Tardos [5] for finding a minimum-cost k-outconnected subdigraph of a digraph (directed graph), and (2) an upper bound on the order of 3-critical graphs by Mader [12]. The Frank-Tardos algorithm has been applied earlier to the mincost k-VCSS problem by several authors, starting with Khuller and Raghavachari [10]; see also [1, 2, 11]. The scaling trick used in Lemma 3.2 below has been used earlier by [8, 9].

2. Notation and preliminary results

Throughout, we assume that the input graph $G = (V, E)$ is k-vertex connected. Let n denote $|V|$.

2.1. A linear programming relaxation. Let H^* denote a k-VCSS of minimum cost, and let $z^* = c(H^*) = \sum_{e \in E(H^*)} c_e$ denote its cost. The following LP (linear program) $P(k)$ gives a lower bound $z(k)$ on z^* (Frank discusses this LP in [4]). There is a variable x_e, $0 \le x_e \le 1$, for each edge e in G. The intention is that the edge incidence vector of every k-VCSS H (possibly, $H = H^*$) forms a feasible solution for $P(k)$. A setpair $W = (W_t, W_h)$ is an ordered pair of disjoint vertex sets, so $W_t \subseteq V$, $W_h \subseteq V$, and $W_t \cap W_h = \emptyset$. An edge uv of G is said to *cover* W if $u \in W_t, v \in W_h$ or $v \in W_t, u \in W_h$. Let $\delta(W)$ denote the set of all edges in G that cover W. If W_t contains at least one vertex, say p, and W_h contains at least one vertex, say q, then note that H has at least $k - (n - |W_t \cup W_h|)$ edges in $\delta(W)$, because on removing the vertices in $V - (W_t \cup W_h)$ from H, the resulting graph has at least this number of openly disjoint paths between p and q and each of these paths contributes one (or more) distinct edges to $\delta(W)$. (Two paths are called *openly disjoint* if every vertex that belongs to both paths is an end vertex of both paths.) Let $\mathcal{S}$ denote the set of all setpairs (W_t, W_h) such that $W_t \neq \emptyset$ and $W_h \neq \emptyset$. It is convenient to keep a parameter ℓ, where ℓ is a positive integer, and write the LP relaxation $P(\ell)$ for the mincost ℓ-VCSS problem.

$$
\begin{aligned}
P(\ell): \qquad z(\ell) \;=\; \text{minimize} \quad & \sum_{e \in E} c_e\, x_e \\
\text{subject to} \quad & \sum_{e \in \delta(W)} x_e \;\ge\; \ell \;-\; (n - |W_t \cup W_h|), \qquad \forall\, W \in \mathcal{S} \\
& 0 \;\le\; x_e \;\le\; 1, \qquad \forall\, e \in E.
\end{aligned}
$$

Lemma 2.1. *Let $z^*(\ell)$ be the minimum cost of an ℓ-VCSS. Then $z^*(\ell) \ge z(\ell)$.*

2.2. k-Outconnected subgraphs. A graph is said to be *k-outconnected* from a so-called *root* vertex r if there exist k openly disjoint paths from r to v, for each vertex v, $v \neq r$. The *mincost k-OC* problem is as follows: given an undirected graph $G = (V, E)$, a root vertex $r \in V$, and nonnegative costs on the edges, find a minimum-cost subgraph H of G such that H is k-outconnected from r. This problem is NP-hard for $k \ge 2$.

Theorem 2.2 (Frank and Tardos (1989), Khuller and Raghavachari (1996))**.** *Let $G=(V,E)$, r, and $c: E \to \mathbb{R}_+$ be as above. There is a 2-approximation algorithm for the mincost k-OC problem. Moreover, the subgraph found by this algorithm has cost at most $2z(k)$.*

Proof. In the directed version $\widehat{G}$ of G, each edge e of G is replaced by two oppositely oriented arcs, and each of these two arcs has cost c_e. Here is an LP relaxation (in fact, an LP formulation) $\widehat{P}$ of the directed mincost k-OC problem on $\widehat{G}$ (with any vertex r as the root): There is a variable x_a for each arc a in $\widehat{G}$; let $\mathcal{R}$ denote the set of all setpairs $W=(W_t, W_h)$ such that the root r is in W_t and $W_h \neq \emptyset$; and for $W \in \mathcal{R}$ let $\hat{\delta}(W)$ denote the set of arcs (u,v) in $\widehat{G}$ with $u \in W_t,\ v \in W_h$.

$$\begin{aligned} \widehat{P}: \quad & \text{minimize} \sum_{a \in E(\widehat{G})} c_a x_a \\ & \text{subject to} \sum_{a \in \hat{\delta}(W)} x_a \geq k - (n - |W_t \cup W_h|), \qquad \forall W \in \mathcal{R} \\ & 0 \leq x_a \leq 1, \qquad \forall a \in E(\widehat{G}). \end{aligned}$$

This LP $\widehat{P}$ has an integer optimal solution (see [4, Theorems 2.1, 2.2]). The Frank-Tardos algorithm solves the directed mincost k-OC problem on $\widehat{G}$ by finding a minimum-cost subdigraph $\widehat{H}$ that is k-outconnected from r, and the cost $c(\widehat{H})$ equals the optimal value of $\widehat{P}$. (The arc incidence vector of $\widehat{H}$ forms an optimal solution of $\widehat{P}$.) Finally, we claim that the optimal value of $\widehat{P}$ is at most $2z(k)$, hence, the undirected version of $\widehat{H}$ satisfies the theorem (it is a subgraph of G that is k-outconnected from r, and it has cost at most $2z(k)$).

To see that the optimal value of $\widehat{P}$ is at most $2z(k)$, observe that the LP relaxation of the directed mincost k-VCSS problem on $\widehat{G}$ has optimal value at most $2z(k)$ (because a feasible solution $\boldsymbol{x}$ of $P(k)$ (the k-VCSS LP on G) gives a feasible solution of the directed k-VCSS LP on $\widehat{G}$, by assigning the value x_e to each of the two arcs corresponding to each edge e). Moreover, an optimal solution of the directed k-VCSS LP on $\widehat{G}$ is also a feasible solution of $\widehat{P}$. Our claim follows. □

Remark: Our algorithm may apply this result to find a solution to the mincost ℓ-OC problem that has cost at most $2z(\ell)$, where $1 \leq \ell \leq k$.

2.3. 3-Critical graphs. For a graph G, let $\kappa(G)$ denote the *vertex connectivity*, i.e., the minimum number of vertices whose removal results in a disconnected graph or the trivial graph (namely, K_1). An *ℓ-separator* of a connected graph is a set of ℓ vertices whose removal results in a disconnected graph.

A graph $G=(V,E)$ is called *3-critical* if the vertex connectivity decreases by $|S|$ on removing the vertices in any set S of at most three vertices, that is, if $\kappa(G-S) = \kappa(G)-|S|, \quad \forall S \subseteq V, |S| \leq 3$. If G is *not* 3-critical, then note that there exists a set S of three vertices such that no $\kappa(G)$-separator contains all the vertices in S. Mader gives an upper bound on the order of 3-critical graphs, [12]. (The proof is written in German, and the result is discussed (without proof) in two survey papers written in English [13, 14].)

Theorem 2.3 (Mader (1977)). *A 3-critical graph with vertex connectivity k has less than $6k^2$ vertices.*

3. The algorithm and its analysis

The algorithm starts with $i := 1$, and a minimum-cost spanning tree H_1. Each iteration $i = 1, 2, \ldots,$ augments H_i to H_{i+1} by adding edges from $E(G) - E(H_i)$ such that the vertex connectivity increases by at least one, and the "augmenting cost" $c(H_{i+1}) - c(H_i)$ is approximately minimum. A detailed description of an iteration follows. Let $\ell = \kappa(H_i)$. If $\ell = k$, then we stop and output H_i as the desired k-VCSS. Now, suppose $\ell < k$. For each edge in H_i, we change the cost to zero (the other edges keep the original costs). By Mader's theorem (and the fact that n is at least $6k^2$) there exist three vertices such that no ℓ-separator of H_i contains all three vertices. We find three such vertices r_1, r_2, r_3 by exhaustively checking for each vertex set S of cardinality three whether $\kappa(H_i - S) > \ell - 3$. For each of these three vertices, we apply the Frank-Tardos algorithm with root r_j ($j = 1$, 2, or 3) and the modified edge costs to find a supergraph $H_{i,j}$ of H_i that is $(\ell + 1)$-outconnected from r_j. We take (the edge set of) H_{i+1} to be the union of (the edge sets of) $H_{i,1}$, $H_{i,2}$, and $H_{i,3}$.

Lemma 3.1. *At every iteration $i = 1, 2, \ldots,$ we have $\kappa(H_{i+1}) \geq \kappa(H_i) + 1$.*

Proof. Let $\ell = \kappa(H_i)$. Note that $\ell < k$. Suppose that $\kappa(H_{i+1}) = \ell$. Then H_{i+1} has an ℓ-separator C, $C \subset V$. Now, H_i is not 3-critical by Mader's theorem, since $n \geq 6k^2 > 6\ell^2$. Hence, there exist three vertices in H_i such that for each ℓ-separator of H_i, at least one of these three vertices is absent from the ℓ-separator. The algorithm finds three such vertices r_1, r_2, r_3. W.l.o.g. r_1 is absent from C. The graph $H_{i,1}$, which is a subgraph of H_{i+1}, is $(\ell + 1)$-outconnected from r_1. Hence, H_{i+1} has $(\ell + 1)$ openly disjoint paths between r_1 and v, for every other vertex v, and one of these paths survives in $H_{i+1} - C$. We have a contradiction, since $H_{i+1} - C$ is connected. The lemma follows. □

Lemma 3.2. *At every iteration $i = 1, 2, \ldots,$ we have $c(H_{i+1}) - c(H_i) \leq \dfrac{6z(k)}{k - \ell}$, where $\ell = \kappa(H_i)$.*

Proof. Note that $\ell < k$. We will prove that for each of the three supergraphs $H_{i,j}$ ($j = 1$, 2, or 3) of H_i, the augmenting cost $c(H_{i,j}) - c(H_i)$ is at most $2z(k)/(k - \ell)$. Then the lemma follows immediately.

Let $\boldsymbol{x} : E \to \mathbb{R}_+$ be an optimal solution to the linear program $P(k)$; note that the cost of $\boldsymbol{x}$ (with respect to the original edge costs c) is $z(k)$.

Recall that (during the construction of $H_{i,j}$, $j = 1, 2, 3$) the edge costs are modified such that an edge already in H_i has zero cost, while the other edges have the original costs. Let $\boldsymbol{x}' : E \to \mathbb{R}_+$ be given by

$$x'_e = \begin{cases} 1, & \text{if } e \text{ is in } H_i \\ \frac{x_e}{k-\ell}, & \text{otherwise.} \end{cases}$$

Clearly, $\boldsymbol{x}'$ has modified cost at most $z(k)/(k-\ell)$. We claim that $\boldsymbol{x}'$ is a feasible solution to the LP $P(\ell+1)$. Then, by Theorem 2.2, the Frank-Tardos algorithm finds an $(\ell+1)$-outconnected supergraph of H_i with augmenting cost at most $2z(k)/(k-\ell)$.

To see the claim, consider any setpair $W \in \mathcal{S}$ and its constraint in the LP $P(\ell+1)$,

$$\sum_{e \in \delta(W)} x'_e \geq (\ell+1) - q,$$

where $q = n - |W_t \cup W_h|$. First, suppose that H_i has no edges in $\delta(W)$. Then since H_i is ℓ-vertex connected, we have $q \geq \ell$. If $q \geq \ell+1$, then, obviously, $\boldsymbol{x}'$ satisfies this constraint. Otherwise, if $q = \ell$, then $\boldsymbol{x}'$ satisfies this constraint because (i) $\boldsymbol{x}$ satisfies the constraint of W in the LP $P(k)$, namely, $\sum_{e \in \delta(W)} x_e \geq k - \ell$, and (ii) each edge $e \in \delta(W)$ has $x'_e = x_e/(k-\ell)$. Now, suppose that H_i has $p \geq 1$ edges in $\delta(W)$. If $p < (\ell+1) - q$, then delete $\leq p$ vertices from W_t and W_h to obtain a new setpair $\hat{W}$ such that $\hat{W}_t \neq \emptyset \neq \hat{W}_h$ and H_i has no edges in $\delta(\hat{W})$, and then apply the previous reasoning to $\hat{W}$ to infer that $\boldsymbol{x}'$ satisfies the constraint of $\hat{W}$, and hence also of W. If $p \geq (\ell+1) - q$, then $\sum_{e \in \delta(W)} x'_e \geq |E(H_i) \cap \delta(W)| = p \geq (\ell+1) - q$. Thus the claim holds. □

Theorem 3.3. *Suppose that the input graph $G = (V, E)$ is k-vertex connected and has order $|V| \geq 6k^2$. Then the algorithm terminates with a k-VCSS that has cost at most $6(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k})z(k)$, where $z(k)$ is the optimal value of the LP relaxation. The algorithm runs in time $O(k^2 n^4 (n + k^{2.5}))$.*

Proof. The vertex connectivity of H_i increases by at least one in every iteration, starting from one, so the algorithm terminates with a k-VCSS in at most $k-1$ iterations. The cost of the k-VCSS is

$$\leq c(H_1) + \sum_{i=1}^{k-1} (c(H_{i+1}) - c(H_i)) \leq \frac{2z(k)}{k} + \sum_{\ell=1}^{k-1} \frac{6z(k)}{k-\ell} \leq 6(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k})z(k).$$

(Note that the minimum spanning tree H_1 is an optimal solution to the mincost 1-OC problem (with any vertex as the root), and an optimal solution $\boldsymbol{x}$ of the LP $P(k)$ gives a feasible solution $\frac{1}{k}\boldsymbol{x}$ of the LP $P(1)$, hence by the proof of Theorem 2.2, $c(H_1) \leq 2z(1) \leq \frac{2z(k)}{k}$.)

To see the running time, note that each iteration i $(1 \leq i < k)$ runs the Frank-Tardos algorithm at most three times, and tests $\kappa(H_i - S)$ for at most n^3 sets of vertices S of cardinality three. Gabow's algorithm [7] tests the vertex connectivity κ in time $O((n+\kappa^{2.5})\cdot\kappa n)$, and there is a version of the Frank-Tardos algorithm, due to Gabow, that runs in time $O(k^2n^2|E|)$, [6, Theorem 4.5]. □

Acknowledgments. The first author thanks Zeev Nutov for several discussions.

References

[1] V.Auletta, Y.Dinitz, Z.Nutov and D.Parente, A 2-approximation algorithm for finding an optimum 3-vertex-connected spanning subgraph, *Journal of Algorithms*, **32**, 1999, pp. 21–30.

[2] J.Cheriyan, T.Jordán and Z.Nutov, On rooted node-connectivity problems, *Algorithmica*, **30**, 2001, pp. 353–375.

[3] J.Cheriyan, S.Vempala and A.Vetta, Approximation algorithms for minimum-cost k-vertex connected subgraphs, *Proc. 34th Ann. ACM STOC*, pp. 306–312, May 2002.

[4] A.Frank, Connectivity augmentation problems in network design, in *Mathematical Programming: State of the Art 1994*, (Eds. J. R. Birge and K. G. Murty), pp. 34–63, The University of Michigan, Ann Arbor, MI, 1994.

[5] A.Frank and E.Tardos, An application of submodular flows, *Linear Algebra and its Applications*, **114/115**, 1989, pp. 329-348.

[6] H.N.Gabow, A representation for crossing set families with applications to submodular flow problems, *Proc. 4th Ann. ACM-SIAM Symposium on Discrete Algorithms*, 1993, pp. 202-211.

[7] H.N.Gabow, Using expander graphs to find vertex connectivity, *Proc. 41st IEEE FOCS*, Nov. 2000.

[8] M.Goemans, A.Goldberg, S.Plotkin, D.Shmoys, E.Tardos and D.Williamson, Improved approximation algorithms for network design problems, *Proc. 5th Ann. ACM-SIAM Symposium on Discrete Algorithms*, 1994, pp. 223-232.

[9] K.Jain, I.Mandoiu, V.V.Vazirani and D.P.Williamson, A primal-dual schema based approximation algorithm for the element connectivity problem, *Proc. 10th Ann. ACM-SIAM Symposium on Discrete Algorithms*, 1999, pp. 484-489.

[10] S.Khuller and B.Raghavachari, Improved approximation algorithms for uniform connectivity problems, *Journal of Algorithms*, **21**, 1996, pp. 434–450.

[11] G.Kortsarz and Z.Nutov, Approximating node connectivity problems via set covers, *Approximation Algorithms for Combinatorial Optimization*, Third international workshop, APPROX 2000, (K.Jansen, S.Khuller (eds.)), Springer, LNCS **1913**, 2000, pp. 194-205.

[12] W.Mader, Endlichkeitsätze für k-kritische Graphen, (German), *Math. Ann.*, **229**, 1977, pp. 143–153.

[13] W.Mader, Connectivity and edge-connectivity in finite graphs, *Surveys in Combinatorics (Proc. Seventh British Combinatorial Conf., Cambridge, 1979)*, pp. 66–95, London Math. Soc. Lecture Note Ser., 38, Cambridge Univ. Press, Cambridge-New York, 1979.

[14] W.Mader, On k-critically n-connected graphs. *Progress in Graph Theory* (Waterloo, Ont., 1982), pp. 389–398, Academic Press, Toronto, ON, 1984.

[15] R.Ravi and D.P.Williamson, An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica*, **18**, 1997, pp. 21-43. (Preliminary version in *Proc. 6th Ann. ACM-SIAM Symposium on Discrete Algorithms*, 1995, pp. 332-341.)

Erratum, September 2001, submitted to *Algorithmica*; (available on the web, at the URL http://www.almaden.ibm.com/cs/people/dpw).