Approximation Algorithms for Minimum-Cost k-Vertex Connected Subgraphs

Joseph Cheriyan , Santosh Vempala , Adrian Vetta
Presented by Yilin Shen

June 11, 2010

Contents

Contents

Contents

Contents

Setpair Definition

Definition (Setpair $W = (W_t, W_h)$)

A setpair $W = (W_t, W_h)$ is an ordered pair of disjoint vertex sets; either W_t or W_h may be empty.

Setpair Definition

Definition (Setpair $W = (W_t, W_h)$)

A setpair $W = (W_t, W_h)$ is an ordered pair of disjoint vertex sets; either W_t or W_h may be empty.

Definition ($\delta(W) = \delta(W_t, W_h)$)

The set of edges with one end-vertex in W_t and the other in W_h.

Setpair Formulations and Relaxation

Setpair Formulation

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & \sum_{e \in \delta(W)} \geq f(W) \quad && \forall W \in S \\
& x_i \in \{0, 1\} && \forall e \in E
\end{aligned}
\tag{1}
$$

where S is all possible combinations of setpairs.

Setpair Formulations and Relaxation

Setpair Formulation

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(W)} \geq f(W) \quad && \forall W \in S \\ & x_i \in \{0,1\} && \forall e \in E \end{aligned} \tag{1}$$

where S is all possible combinations of setpairs.

k-VCSS

$$f(W) = \begin{cases} \max\{0, k - |V \setminus (W_h \cup W_t)|\}, & \text{if } W_t \neq \emptyset \text{ and } W_h \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

An $\log(k)$ Approximation Algorithm for Undirected Graphs

Theorem (Frank and Tardos)

Let $G = (V, E)$, r, and $c : E \to R_+$ be as above. There is a 2-approximation algorithm for the mincost k-outconnected problem. Moreover, the subgraph found by this algorithm has cost at most $2z(k)$, where $z(k)$ the optimal solution of LP.

An $\log(k)$ Approximation Algorithm for Undirected Graphs

Theorem (Frank and Tardos)

Let $G = (V, E)$, r, and $c : E \to R_+$ be as above. There is a 2-approximation algorithm for the mincost k-outconnected problem. Moreover, the subgraph found by this algorithm has cost at most $2z(k)$, where $z(k)$ the optimal solution of LP.

Definition (3-Critical 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 $\kappa(G - S) = \kappa(G) - |S|$, $S \in V, |S| \leq 3$, where $\kappa(G)$ denote the vertex connectivity.

An $\log(k)$ Approximation Algorithm for Undirected Graphs

Theorem (Frank and Tardos)

Let $G = (V, E)$, r, and $c : E \rightarrow R_+$ be as above. There is a 2-approximation algorithm for the mincost k-outconnected problem. Moreover, the subgraph found by this algorithm has cost at most $2z(k)$, where $z(k)$ the optimal solution of LP.

Definition (3-Critical 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 $\kappa(G - S) = \kappa(G) - |S|$, $S \in V, |S| \leq 3$, where $\kappa(G)$ denote the vertex connectivity.

Theorem (Mader)

A 3-critical graph with vertex connectivity k has less than $6k^2$ vertices.

$\log(k)$ Approximation Algorithm (Cont.)

Approximation Algorithm

1. $H_1 \leftarrow$ minimum spanning tree on G
2. Find three vertices r_1, r_2, r_3 by exhaustively checking for each vertex set such that $\kappa(H_i - S) > l - 3$, $l = \kappa(H_i)$
3. Apply Frank-Tardos algorithm with each root r_j to find a supergraph $H_{i,j}$ on H_i which is $(l+1)$-outconnected from r_j
4. H_{i+1} is the union of $H_{i,1} + H_{i,2} + H_{i,3}$

$\log(k)$ Approximation Algorithm (Cont.)

Lemma

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

log(k) Approximation Algorithm (Cont.)

Lemma

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

Lemma

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

log(k) Approximation Algorithm (Cont.)

Lemma

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

Lemma

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

Theorem

Let $G = (V, E)$ be a k-vertex connected graph with at least $6k^2$ vertices. Then the algorithm terminates with a k-VCSS that has cost at most $6 \log k z(k)$, where $z(k)$ is the optimal value of the LP relaxation. The algorithm runs in time $O(k^2 n^4 (n + k^{2.5}))$.

Structure of a Basic Solution (Extreme Point Optimum Solution)

- Crossing Setpairs
- Bisubmodular Functions, Crossing Bisupermodular Functions
- Skew Bisupermodular Functions

Cont.

Theorem

Let the requirement function f of (LP) be skew bisupermodular, and let x be a feasible solution to (LP) such that $x_e > 0$ for all edges $e \in E$. Suppose that the setpairs W and Y have $f(W) > 0$, $f(Y) > 0$, and moreover, W and Y overlap, and are tight (also, note that W is tight, it overlaps Y, and $f(W) > 0$). Then one of the following holds:

- *The setpairs $W \otimes Y$ and $W \oplus Y$ are tight, and* $\chi_W + \chi_Y = \chi_{W \otimes Y} + \chi_{W \oplus Y}$.
- *The setpairs $\bar{W} \otimes Y$ and $\bar{W} \oplus Y$ are tight, and* $\chi_W + \chi_Y = \chi_{\bar{W} \otimes Y} + \chi_{\bar{W} \oplus Y}$.

where χ_W denote the edge incidence vector of $\delta(W)$ and a setpair W is called tight if $x(\delta(W)) = f(W)$ given a feasible solution x to (LP).

Cont.

Theorem

Let the requirement function f of (LP) be skew bisupermodular, and let x be a basic solution to (LP) such that $0 < x_e < 1$ for all edges $e \in E$. Then there exists a non-overlapping family $\mathcal{L}$ of tight setpairs such that:

- *Every setpair $W \in L$ has $f(W) \geq 1$.*
- $|\mathcal{L}| = |E|$.
- *The vectors χ_W, $W \in L$, are linearly independent.*
- *x is the unique solution to $\{x(\delta(W)) = f(W), \forall W \in \mathcal{L}\}$.*

Cont.

Theorem

Let k and n be positive integers, and let $\epsilon < 1$ be a positive number such that k is at most $(1 - \epsilon)n$. There is a polynomial-time algorithm that, given an n-vertex (directed or undirected) graph, finds a solution to the k-vertex connectivity problem of cost at most $O(\sqrt{n/\epsilon})$ times the optimal cost.

Cont.

Theorem

Let k and n be positive integers, and let $\epsilon < 1$ be a positive number such that k is at most $(1-\epsilon)n$. There is a polynomial-time algorithm that, given an n-vertex (directed or undirected) graph, finds a solution to the k-vertex connectivity problem of cost at most $O(\sqrt{n/\epsilon})$ times the optimal cost.

Theorem

Let $\epsilon < 1$ be a positive number, and suppose that $k \leq (1-\epsilon)n$. Then any nonzero basic solution of (LP-VC) has an edge of weight $\Omega(\sqrt{\epsilon/n})$.

Cont.

Theorem

Let k and n be positive integers, and let $\epsilon < 1$ be a positive number such that k is at most $(1-\epsilon)n$. There is a polynomial-time algorithm that, given an n-vertex (directed or undirected) graph, finds a solution to the k-vertex connectivity problem of cost at most $O(\sqrt{n/\epsilon})$ times the optimal cost.

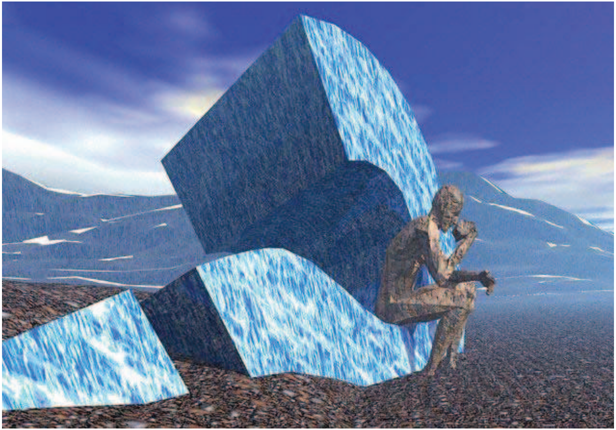
Theorem

Let $\epsilon < 1$ be a positive number, and suppose that $k \leq (1-\epsilon)n$. Then any nonzero basic solution of (LP-VC) has an edge of weight $\Omega(\sqrt{\epsilon/n})$.

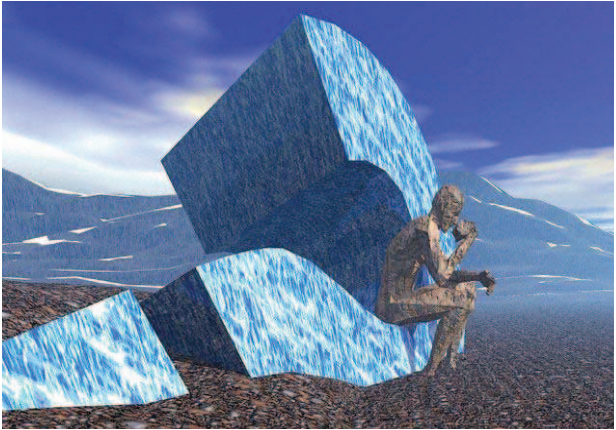
Theorem

Suppose that the requirement function f for the linear program (LP-VC) is crossing (or, skew) bisupermodular. Let x be a nonzero basic solution of (LP-VC), and let $\mathcal{L}$ be a non-crossing family of setpairs characterizing x. Then there exists an edge e with $x_e \geq 1/\Omega(\sqrt{|\mathcal{L}|})$.

Questions?



Questions?



Thank you !