Approximating Minimum-Power Degree and Connectivity Problems *

Guy Kortsarz[†] Vahab S. Mirrokni[‡] Zeev Nutov[§] Elena Tsanko[¶]

Abstract

Power optimization is a central issue in wireless network design. Given a graph with costs on the edges, the power of a node is the maximum cost of an edge incident to it, and the power of a graph is the sum of the powers of its nodes. Motivated by applications in wireless networks, we consider several fundamental undirected network design problems under the power minimization criteria. Given a graph $\mathcal{G} = (V, \mathcal{E})$ with edge costs $\{c(e) : e \in \mathcal{E}\}$ and degree requirements $\{r(v) : v \in V\}$, the Minimum-Power Edge-Multi-Cover (MPEMC) problem is to find a minimum-power subgraph G of $\mathcal{G}$ so that the degree of every node v in G is at least $r(v)$. We give an $O(\log n)$-approximation algorithms for MPEMC, improving the previous ratio $O(\log^4 n)$. This is used to derive an $O(\log n + \alpha)$-approximation algorithm for the undirected Minimum-Power k-Connected Subgraph (MPkCS) problem, where α is the best known ratio for the min-cost variant of the problem. Currently, $\alpha = O\left(\log k \cdot \log \frac{n}{n-k}\right)$ which is $O(\log k)$ unless $k = n - o(n)$, and is $O(\log^2 k) = O(\log^2 n)$ for $k = n - o(n)$. Our result shows that the min-power and the min-cost versions of the k-Connected Subgraph problem are equivalent with respect to approximation, unless the min-cost variant admits an $o(\log n)$-approximation, which seems to be out of reach at the moment.

*Preliminary version appeared in LATIN, pages 423-435, 2008.

[†]Rutgers University, Camden, guyk@camden.rutgers.edu.

Partially supported by NSF award number 0829959.

[‡]Microsoft Research, mirrokni@microsoft.com

[§]The Open University of Israel, Raanana nutov@openu.ac.il

[¶]IBM, Haifa tsanko@il.ibm.com

1 Introduction

1.1 Motivation and problems considered

Wireless networks are studied extensively due to their wide applications. The power consumption of a station determines its transmission range, and thus also the stations it can send messages to; the power typically increases at least quadratically in the transmission range. Assigning power levels to the stations (nodes) determines the resulting communication network. Conversely, given a communication network, the power required at v only depends on the farthest node that is reached directly by v. This is in contrast with wired networks, in which every pair of stations that need to communicate directly incurs a cost. An important network property is fault-tolerance, which is often measured by minimum degree or node-connectivity of the network. Such power minimization problems were vastly studied. See for example [1, 2, 7, 12, 13, 4, 3, 8, 14] for a small sample of papers in this area. The first problem we consider is finding a low power network with specified lower bounds on node degrees. The second problem is the Min-Power k-Connected Subgraph problem. We devise approximation algorithms for these problems, improving significantly the previously best known ratios.

Definition 1.1 *Let $G = (V, E)$ be a graph with edge-costs $\{c(e) : e \in E\}$. For $v \in V$, the* power *$p(v) = p_G(v)$ of v in G (w.r.t. c) is the maximum cost of an edge in G incident to v. The power of the graph is the sum of the powers of its nodes.*

Unless stated otherwise, graphs are assumed to be undirected and simple. Let $G = (V, E)$ be a graph. For $X \subseteq V$, $\Gamma_E(X) = \Gamma_G(X) = \{u \in V - X : v \in X, vu \in E\}$ is the set of neighbors of X in G, $\delta_E(X) = \delta_G(X)$ is the set of edges from X to $V - X$ in G, and $d_E(X) = |\delta_E(X)|$ is the degree of X in G. Let $\mathcal{G} = (V, \mathcal{E}; c)$ be a *network*, that is, $(V, \mathcal{E})$ is a graph and c is a cost function on $\mathcal{E}$. Let $n = |V|$. Sometimes, we write $\mathcal{G} = (V, \mathcal{E})$ and refer to $\mathcal{G}$ as a graph. Given a network $\mathcal{G} = (V, \mathcal{E}; c)$, we seek to find a low power *communication network*, that is, a low power subgraph $G = (V, E)$ of $\mathcal{G}$ that satisfies some property.

Definition 1.2 *Given a requirement function r on V, we say that a graph $G = (V, E)$ (or that E) is an* r-edge cover *if $d_G(v) \geq r(v)$ for every $v \in V$.*

Minimum-Power Edge-Multi-Cover (MPEMC):
Instance: A network $\mathcal{G} = (V, \mathcal{E}; c)$ and degree requirements $\{r(v) : v \in V\}$.
Objective: Find a min-power subgraph G of $\mathcal{G}$ so that G is an r-edge cover.

A graph is *k-connected* if it contains k internally-disjoint uv-paths for all $u, v \in V$.

Minimum-Power k-Connected Subgraph (MPkCS):
Instance: A network $\mathcal{G} = (V, \mathcal{E}; c)$, and an integer k.
Objective: Find a minimum-power k-connected spanning subgraph G of $\mathcal{G}$.

1.2 Related Work

Results on MPEMC: The Minimum-Cost Edge-Multi-Cover problem is essentially the fundamental b-Matching problem, which is solvable in polynomial time, c.f., [5]. The previously best known approximation ratio for the min-power variant MPEMC was $\min\{r_{\max}+1, O(\log^4 n)\}$ due to [7], where $r_{\max} = \max_{v \in V} r(v)$ denotes the maximum requirement.

Results on connectivity problems: Minimum-cost connectivity problems were studied extensively, see surveys in [10] and [11]. The currently best known approximation ratio for the Minimum-Cost k-Connected Subgraph (MCkCS) problem is $\alpha = O\left(\log k \cdot \log \frac{n}{n-k}\right)$ [15], which is $O(\log k)$ for all k but $k = n - o(n)$, and is $O(\log^2 k) = O(\log^2 n)$ for $k = n - o(n)$. For further results on other minimum-power connectivity problems, among them problems on directed graphs see [2, 7, 13, 12, 8, 14]. The following statement from [7], which first part was observed independently in [8], relates the power and cost variants.

Theorem 1.1 ([7, 8])

(i) *If there exists an α-approximation algorithm for* MCkCS *and a β-approximation algorithm for* MPEMC *then there exists a $(2\alpha + \beta)$-approximation algorithm for* MPkCS.

(ii) *If there exists a ρ-approximation algorithm for* MPkCS *then there exists a $(2\rho + 1)$-approximation for* MCkCS.

1.3 Our Results

The previous best approximation ratio for MPEMC was $\min\{r_{\max}+1, O(\log^4 n)\}$ [7]. We prove:

Theorem 1.2 *Undirected* MPEMC *admits an $O(\log n)$-approximation algorithm.*

The previously best known ratio for MPkCS was $O(\alpha + \log^4 n)$ [7], where α is the best ratio for MCkCS. From Theorems 1.2 and 1.1 we get:

Theorem 1.3 MPkCS *admits an* $O(\alpha + \log n)$*-approximation algorithm, where* α *is the best ratio for* MCkCS*. Thus unless* $k = n - o(n)$*,* MPkCS *admits an* $O(\log n)$*-approximation algorithm, and for* $k = n - o(n)$ *the approximation ratio is* $O(\log^2 n)$*.*

Theorem 1.3 implies that the min-cost and the min-power variants of the k-Connected Subgraph problem are equivalent with respect to approximation, unless the min-cost variant admits a better than $O(\log n)$-approximation; the latter seems to be out of reach at the moment.

2 Approximating MPEMC

Let opt denote the optimal solution value of a problem at hand. The most natural heuristic for approximating MPEMC is as follows. Guess opt (more precisely, using binary search, guess an almost tight lower bound τ on opt). Cover some fraction of the total requirements within budget opt, and iterate. Proposition 2.1 below shows that this strategy fails. Suppose that we are given an instance of MPEMC and a budget P and our goal is to solve the "budgeted coverage" version of MPEMC: find an edge set $I \subseteq \mathcal{E}$ so that $p(I) \leq P$ and the amount of requirement $\sum_{v \in V} \min\{d_I(v), r(v)\}$ covered by I is maximum. We show that this problem is at least as hard as the Densest k-Subgraph problem: given a graph $\mathcal{G} = (V, \mathcal{E})$ and an integer k, find a subgraph of $\mathcal{G}$ with k nodes that has the maximum number of edges. The best known approximation ratio for Densest k-Subgraph is roughly $n^{-1/3}$ [6], and in spite of numerous attempts to improve it, this ratio holds for over 12 years. We prove:

Proposition 2.1 *If there exists a* ρ*-approximation algorithm for the budgeted coverage version of* MPEMC *with unit costs, then there exists a* ρ*-approximation algorithm for* Densest k-Subgraph*.*

Proof: Given an instance $\mathcal{G} = (V, \mathcal{E}), k$ of Densest k-Subgraph, define an instance $(\mathcal{G}, r, P)$ of budgeted coverage version of MPEMC with unit costs as follows: $r(v) = k - 1$ for all $v \in V$ and $P = k$. Then the problem is to find a node subset $U \subseteq V$ with $|U| = k$ so that the number of edges in the subgraph induced by U in $\mathcal{G}$ is maximum. The later is the Densest k-Subgraph problem. $\square$

2.1 Reduction to bipartite graphs

We will show an $O(\log n)$-approximation algorithm for (undirected) *bipartite* MPEMC where $\mathcal{G} = (A + B, \mathcal{E})$ is a bipartite graph and $r(a) = 0$ for every $a \in A$ (so, only the nodes in B

may have positive requirements). The following statement shows that getting an $O(\log n)$-approximation algorithm for the bipartite MPEMC is sufficient.

Lemma 2.2 *If there exists a ρ-approximation algorithm for bipartite* MPEMC *then there exists a 2ρ-approximation algorithm for general* MPEMC.

Proof: Given an instance $(\mathcal{G} = (V, \mathcal{E}), c, r)$ of MPEMC, construct an instance $(\mathcal{G}' = (A + B, \mathcal{E}'), c', r')$ of bipartite MPEMC as follows. Let $A = \{a_v : v \in V\}$ and $B = \{b_v : v \in V\}$ (so each of A, B is a copy of V) and for every $uv \in \mathcal{E}$ add two edges: $a_u a_v$ and $a_v a_u$ each with cost $c(uv)$. Also, set $r'(b_v) = r(v)$ for every $b_v \in B$ and $r'(a_v) = 0$ for every $a_v \in A$. Given $F' \subseteq \mathcal{E}'$ let $F = \{uv \in \mathcal{E} : a_u b_v \in F' \text{ or } a_v b_u \in F'\}$ be the edge set in $\mathcal{E}$ that corresponds to F'. Now compute an r'-edge cover E' in $\mathcal{G}'$ using the ρ-approximation algorithm and output the edge set $E \subseteq \mathcal{E}$ that corresponds to E', namely $E = \{uv \in \mathcal{E} : a_u b_v \in E' \text{ or } a_v b_u \in E'\}$. It is easy to see that if F' is an r'-edge cover then F is an r-edge cover. Furthermore, if for every edge in F correspond two edges in F' ($|F'| = 2|F|$), then F is an r-edge cover if, and only if, F' is an r'-edge cover. The later implies that $\mathsf{opt}' \leq 2\mathsf{opt}$, where opt and opt' is the optimal solution value to $\mathcal{G}, c, r$ and $\mathcal{G}', c', r'$, respectively. Consequently, E is an r-edge cover, and $p_E(V) \leq p_{E'}(A + B) \leq \rho \cdot \mathsf{opt}' \leq 2\rho \cdot \mathsf{opt}$. □

2.2 An $O(\log n)$-approximation for bipartite MPEMC

We prove that bipartite MPEMC admits an $O(\log n)$-approximation algorithm. The *residual requirement* of B w.r.t. an edge set J is defined by $r_J(B) = \sum_{v \in V} \max\{r(b) - d_J(b), 0\}$.

Lemma 2.3 *For bipartite* MPEMC *there exists a polynomial time algorithm that given an integer τ either establishes that $\tau < \mathsf{opt}$ or returns an edge set $J \subseteq \mathcal{E}$ such that the following holds:*

$$p_J(V) \leq 4\tau \tag{1}$$

$$r_J(B) \leq 3r(B)/4 \tag{2}$$

Note that if $\tau < \mathsf{opt}$ then the algorithm may return an edge set J that satisfies (1) and (2); if the algorithm declares "$\tau < \mathsf{opt}$" then this is correct. An $O(\log n)$-approximation algorithm for the bipartite MPEMC easily follows from Lemma 2.3:

While $r(B) > 0$ do

- Find the least integer τ so that the algorithm in Lemma 2.3 returns an edge set J so that (1) and (2) holds (note that $\tau - 1 < \mathsf{opt}$).
- $E \leftarrow E + J$, $\mathcal{E} \leftarrow \mathcal{E} - J$, $r \leftarrow r_J$.

End While

The least integer τ as above can be found in polynomial time using binary search in the range $[0, \ldots, p(\mathcal{G})]$ as follows. Suppose that our current search range is $[\ell, \ldots, L]$. Assuming $L - \ell \geq 2$ (if $L - \ell \in \{0, 1\}$ there are at most 2 values to check), we check the value $\tau = \lfloor (\ell + L)/2 \rfloor$. We continue the search in the range $[\lfloor (\ell + L)/2 \rfloor + 1, \ldots, L]$ if the algorithm as in Lemma 2.3 establishes that $\tau < \mathsf{opt}$, and in the range $[\ell, \ldots, \lfloor (\ell + L)/2 \rfloor]$ otherwise. At the end the algorithm returns a solution for τ and establishes that $\tau - 1 \leq \mathsf{opt}$. Thus $\tau = O(\mathsf{opt})$. The number of iterations is $O(\log r(B))$, and at every iteration an edge set of power at most $O(\mathsf{opt})$ is added. Thus the algorithm can be implemented to run in polynomial time, and has approximation ratio $O(\log r(B)) = O(\log(n^2)) = O(\log n)$. Therefore all that remains is proving Lemma 2.3.

2.3 Proof of Lemma 2.3

Definition 2.1 *Let τ be an integer, let $R = r(B) = \sum_{b \in B} r(b)$. An edge $ab \in \mathcal{E}$, $b \in B$, is* dangerous *if $c(ab) \geq 2 \cdot \tau \cdot r(b)/R$. Let $\mathcal{I}$ be the set of non-dangerous edges in $\mathcal{E}$.*

Lemma 2.4 *$p_{\mathcal{I}}(B) \leq 2 \cdot \tau$.*

Proof: Note that $p_{\mathcal{I}}(b) \leq 2 \cdot \tau \cdot r(b)/R$ for every $b \in B$. Thus:

$$p_{\mathcal{I}}(B) = \sum_{b \in B} p_{\mathcal{I}}(b) \leq \sum_{b \in B} (2\tau \cdot r(b)/R) = \frac{2\tau}{R} \sum_{b \in B} r(b) = 2\tau \ .$$

□

Lemma 2.5 *Suppose that $\tau \geq p(E)$ for a feasible solution E. Then $J = E \cap \mathcal{I}$ covers at least $R/2$ of the total requirement, namely, $r_J(B) \leq R - R/2 = R/2$.*

Proof: Let $F = E - \mathcal{I}$ be the set of dangerous edges in E, and let $D = \{b \in B : d_F(b) > 0\}$. We claim that $r(D) \leq R/2$, implying that $J = E - F$ covers at least $R/2$ of the requirement. Our claim that $r(D) \leq R/2$ follows from the following sequence of inequalities:

$$\tau \geq p(E) \geq \sum_{b \in D} p_F(b) \geq \sum_{b \in D} (2\tau \cdot r(b)/R) = \frac{2\tau}{R} \sum_{b \in D} r(b) = \frac{2\tau}{R} r(D) \ .$$

□

Lemmas 2.4 and 2.5 imply that we may ignore the dangerous edges and still be able to cover within power τ a fraction of $1/2$ of the total requirement. If $\tau = O(\mathsf{opt})$, then once dangerous edges are ignored, the algorithm does not need to take the power incurred in B into account, as the total power of B w.r.t. all the non-dangerous edges is $2\tau = O(\mathsf{opt})$.

Therefore, the problem we want to solve is similar to the bipartite MPEMC, except that we want to minimize the power of A only. Formally:

Instance: A bipartite graph $\mathcal{G} = (A + B, \mathcal{I})$ with edge-costs $\{c(e) : e \in \mathcal{I}\}$, requirements $\{r(b) : b \in B\}$, and a budget $\tau = P$.
Objective: Find $J \subseteq \mathcal{I}$ with $p_J(A) \leq P$ and $\sum_{b \in B} \min\{d_J(b), r(b)\}$ maximum.

Note that in the above graph we assume that the dangerous edges were removed

In order to represent all possible power choices for a node $a \in A$, we built the following bipartite graph $\hat{G} = (\hat{A}, B, \hat{E})$. For every node $a \in A$ and every edge $e \in \delta_{\mathcal{G}}(a)$ add a node a_e into $\hat{A}$ and give it cost $c(e)$. Not all of $\delta_{\mathcal{G}}(a)$ is added into $\hat{E}$ but only edges ab so that $c(ab) \leq c(e)$. Intuitively, a choice of a_e implies a choice of power $c(e)$ for v. Hence it can reduce the demand only of nodes b so that $c(ab) \leq c(e)$. We assume $\tau \geq \mathsf{opt}$ (and may get a contradiction). Thus we discard any $a'_{e'}$ so that $c(e') > \tau$.

We treat the problem as a set-coverage problem. We apply the well known set-coverage algorithm on $\hat{G}$ (see [9]), except that we stop once the cost exceeds τ. We later show that the distinction between power and cost is not important here, as we run the algorithm on $\hat{G}$.

For a node a_e, let $E(a_e)$ be the edges of a_e in the original graph $\mathcal{G}$, namely, $E(a_e) = \{e' \in \delta_{\mathcal{G}}(a) \mid c(e') \leq c(e)\}$. We denote by $cover_J(a_e) = r_J(A) - r_{J \cup E(a_e)}(A)$. In the next algorithm we say that a_e is the best ratio node if $cover_J(a_e)/c(e)$ is the largest over all $a'_{e'}$ nodes.

Procedure GREEDY$(\hat{G}(\hat{V}, \hat{E}))$

1. $J \leftarrow \emptyset$; $S \leftarrow \emptyset$
2. **While** $c(S) \leq \tau$ **do**
   (a) Select the best ratio node a_e and add it to S
   (b) $c(S) \leftarrow c(S) + c(e)$
   (c) $J \leftarrow J \cup E(a_e)$
3. Return J

Let the final S be $S = \{a^1_{e_1}, \ldots, a^k_{e_k}\}$ (where the nodes were chosen in this order).

Claim 2.6 $p_J(A) \leq c(S) = \sum_{i=1}^{k} c(e_i)$

Proof: Fix a node a. Let e' be the maximum cost edge so that $E(a_{e'}) \cap J \neq \emptyset$. Note that by the definition of $\hat{G}$ and $E(a_e)$, for every $e'' \neq e'$ so that $E(a_{e''}) \cap J \neq \emptyset$, $E(a_{e''}) \subseteq E(a_{e'})$.

Thus at the end of the algorithm $\delta_{\mathcal{G}}(a) \cap J = E(a_{e'})$. This means that the contribution of a to $p_J(A)$ is $c(e')$. In contrast, all the copies of a contribute their cost to $c(S)$ (and in particular $c(e')$). The claim follows. $\square$

Lemma 2.7 *If $\tau \geq \mathsf{opt}$ the power added by the greedy algorithm is at most 2τ, while the demand covered by the algorithm is at least $r(B)/4$*

Proof: By Claim 2.6, in order to prove $p_J(A) \leq 2 \cdot \tau$, it is enough to show that $c(S) \leq 2 \cdot \tau$. Before $E(a^k_{e_k})$ is added into J, the cost of $c(S)$ was at most τ. Since $c(e_k) \leq \tau$, the first part of the claim follows.

We now bound from below the coverage of J. Note that even though the dangerous edges were removed, by Lemmas 2.4 and 2.5 there exists a set J' (in $\mathcal{G}$) that can cover $r(B)/2$ demand so that $p_{J'}(A) \leq \mathsf{opt}$. If the set J output by the algorithm, satisfies at least $r(B)/4$ of the demand covered by J', we are done. Else, at least $r(B)/4$ demand that can be satisfied by J' remains *uncovered at the end of the run of the algorithm.* Clearly, $p_{J'}(A) \leq \mathsf{opt} \leq \tau$, while J' will reduce the demand by $r(B)/4$. Going back to $\hat{G}$, J' corresponds to a collection $S' = \{a_e\}$ of nodes, whose sum of costs equals $p_{J'}(A)$ and the sum $\sum_{a_e \in S'} cover_J(a_e)$ is *at least* the coverage of J' namely, at least $r(B)/4$. By a simple averaging argument, *at the end of the run of the algorithm* there exists a node a_e so that $cover_J(a_e)/c(e) \geq r(B)/(4\tau)$. Let J_i be the edges in the partial solution before $E(a^i_{e_i})$ is added into J. Then clearly, this implies that for every i:

$$\frac{cover_{J_i}(a^i_{e_i})}{c(e_i)} \geq \frac{r(B)}{4p_{J'}(A)} \geq \frac{r(B)}{4\tau}. \tag{3}$$

We bound from below the coverage as follows:

$$\sum_{i=1}^{k} cover_{J_i}(a^i_{e_i}) = \sum_{i=1}^{k} c(e_i) \cdot \frac{cover_{J_i}(a^i_{e_i})}{c(e_i)} \geq \sum_{i=1}^{k} c(e_i) \frac{r(B)}{4\tau} \geq \frac{r(B)}{4}.$$

The last two inequalities follows from Inequality (3) and from the fact that $\sum_{i=1}^{k} c(e_k) \geq \tau$ by the algorithm. Note that if the set J does not satisfy the coverage lower bound $r(B)/4$ as stated in the lemma, by the above discussion we just proved that $\tau < OPT$. $\square$

Lemma 2.3 directly follows from Lemma 2.4 and Lemma 2.7.

Acknowledgment

We thank two anonymous referees whose comments helped in considerably simplifying the presentation. We especially thank the referee who sent the simplified algorithm for Lemma

2.7 and its proof.

References

[1] E. Althaus, G. Calinescu, I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment for symmetric connectivity in static ad-hoc wireless networks. *Wireless Networks*, 12(3):287–299, 2006.

[2] G. Calinescu, S. Kapoor, A. Olshevsky, and A. Zelikovsky. Network lifetime and power assignment in ad hoc wireless networks. In *ESA*, pages 114–126, 2003.

[3] G. Calinescu and P. J. Wan. Range assignment for biconnectivity and k-edge connectivity in wireless ad hoc networks. *Mobile Networks and Applications*, 11(2):121128, 2006.

[4] A. E. F. Clementi, P. Penna, and R. Silvestri. Hardness results for the power range assignmet problem in packet radio networks. In *APPROX*, pages 197–208, 1999.

[5] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Scrijver. *Combinatorial Optimization.* Wiley, 1998.

[6] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, pages 410–421, 2001.

[7] M. T. Hajiaghayi, G. Kortsarz, V. S. Mirrokni, and Z. Nutov. Power optimization for connectivity problems. *Math. Program.*, 110(1):195–208, 2007. Preliminary version in IPCO 2005.

[8] X. Jia, D. Kim, S. Makki, P.-J. Wan, and C.-W. Yi. Power assignment for k-connectivity in wireless ad hoc networks. *J. Comb. Optim.*, 9(2):213–222, 2005. Preliminary version in INFOCOM 2005.

[9] D. Johnson. Approximation algorithms for combinatorial problems. *J. Computing and System Sciences*, 9(3):256–278, 1974.

[10] S. Khuller. *Approximation algorithms for finding highly connected subgraphs,* Ch. 6 in Approximation Algorithms for NP-hard problems, Editor D. S. Hochbaum, pages 236-265. PWS, 1995.

[11] G. Kortsarz and Z. Nutov. *Approximating minimum-cost connectivity problems,* Ch. 58 in Approximation algorithms and Metaheuristics, Editor T. F. Gonzalez. Chapman & Hall/CRC, 2007.

[12] Y. Lando and Z. Nutov. On minimum power connectivity problems. In *ESA*, pages 87–98, 2007.

[13] Z. Nutov. Approximating minimum power covers of intersecting families and directed connectivity problems. In *APPROX*, pages 236–247, 2006.

[14] Z. Nutov. Approximating minimum power k-connectivity. to appear in Ad-Hoc-NOW, 2008.

[15] Z. Nutov. An almost $O(\log k)$-approximation for k-connected subgraphs. In *SODA*, pages 912–921, 2009.