Available online at www.sciencedirect.com
SCIENCE DIRECT®
Operations Research Letters 32 (2004) 212–216
Operations Research Letters
www.elsevier.com/locate/dsw

Finding 2-edge connected spanning subgraphs

Woonghee Tim Huh[1]

Industrial Engineering and Operations Research Department, Columbia University, New York, NY 10027, USA

Received 5 April 1999; received in revised form 17 April 2001; accepted 21 August 2003

Abstract

This paper studies the NP-hard problem of finding a minimum size 2-edge connected spanning subgraph (2-ECSS). An algorithm is given that on an r-edge connected input graph $G=(V,E)$ finds a 2-ECSS of size at most $|V|+(|E|-|V|)/(r-1)$. For r-regular, r-edge connected input graphs for $r=3$, 4, 5 and 6, this gives approximation guarantees of $\frac{5}{4}$, $\frac{4}{3}$, $\frac{11}{8}$ and $\frac{7}{5}$, respectively.

Keywords: Graph; 2-edge connectivity; Approximation algorithm; Linear programming

1. Introduction

The *minimum size* 2-*edge connected spanning subgraph* (2-*ECSS*) *problem* is the following: Given a 2-edge connected graph G, find a spanning subgraph H of G such that H is 2-edge connected and H has the least number of edges. We denote the number of edges in a minimum size 2-ECSS by $\varepsilon(G)$. We can immediately note that $\varepsilon(G) \geqslant |V(G)|$, and the equality holds only if G has a Hamiltonian cycle. This problem is NP-hard because the NP-hard Hamiltonian cycle problem reduces to it [2].

An α-*approximation algorithm* for a combinatorial optimization problem is an algorithm which runs in polynomial time, and delivers a feasible solution to the problem whose cost is within a multiplicative factor α of the cost of the optimal solution. The number α is called the *approximation guarantee* of the algorithm.

Khuller and Vishkin [4] gave a $\frac{3}{2}$-approximation algorithm for the minimum size 2-ECSS problem using the depth-first search algorithm and a notion called “tree carving”. Garg et al. [3] claimed to have a $\frac{5}{4}$-approximation algorithm for the minimum size 2-ECSS problem, but no proof was given. Cheriyan et al. [2] devised an algorithm for the minimum size 2-ECSS problem with the approximation guarantee of $\frac{17}{12}$. They used an ear decomposition to construct a feasible 2-ECSS.

In this paper, we give an algorithm which, on an r-edge connected input graph, finds a 2-ECSS of size at most $|V| + (|E| - |V|)/(r - 1)$, where $r \geqslant 2$. For r-regular, r-edge connected input graphs, the LP

[1] Research done while the author was at the Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Canada, and partially supported by the Natural Sciences and Engineering Research Council of Canada, and Doorae Community Movement.

E-mail address: huh@ieor.columbia.edu (W.T. Huh).

0167-6377/$ - see front matter
doi:10.1016/j.orl.2003.08.006

relaxation of the integer program has the optimal value of $|V|$, and thus for $r = 3, 4, 5$ and 6, this gives approximation guarantees of $\frac{5}{4}, \frac{4}{3}, \frac{11}{8}$ and $\frac{7}{5}$, respectively.

Since the submission of this paper, Vempala and Vetta [6] presented a $\frac{4}{3}$-algorithm for general graphs by first finding a minimum subgraph of degree at least 2. Krysta and Kumar [5] recently improved the approximation guarantee to $\frac{4}{3} - \varepsilon$, and for 3-regular graphs, to $\frac{21}{16} + \varepsilon$.

2. Definitions and notation

An *ear decomposition* of the graph $G = (V, E)$ is a partition of $E(G)$ into $P_0 + P_1 + \cdots + P_k$ such that P_0 is a trivial path with one vertex, and for $1 \leqslant i \leqslant k$, P_i is a nontrivial path or cycle where the subgraph formed by $\{P_0, P_1, \ldots, P_{i-1}\}$ contains both end vertices of P_i, but none of the internal vertices of P_i. Each P_i is called an *ear*, and if P_i has length l, it is an l-ear. We say k is the number of ears.

It is easy to observe that $k = |E| - |V| + 1$. If the ear decomposition $P_0 + P_1 + \cdots + P_k$ has t 1-ears, then we can discard all 1-ears, and obtain a valid ear decomposition for a spanning subgraph H of G. This subgraph H is 2-edge connected by Proposition 1 below. Thus we note that $|E| - t$ is an upper bound for ε.

Proposition 1 (Whitney [7]). *A graph is 2-edge connected if and only if it has an ear decomposition.*

For $S \subset V$, we let $\delta(S) = \{vw \in E | v \in S, w \notin S\}$ and $\gamma(S) = \{vw \in E | v, w \in S\}$. We denote by $\Gamma(v) = \{vw \in E | w \in V\}$ the set of neighbors of vertex v.

3. Description of the algorithm

The algorithm 2-ECSS is based on finding an ear decomposition. The discussion in the previous section motivates us to maximize the number of 1-ears. This is equivalent to minimizing the number of ears whose length is greater than 1. Our greedy algorithm attempts to achieve this by constructing long ears one by one.

The first ear P_1 is some cycle. At an intermediate step, when there is a subgraph $H := P_1 + \cdots + P_{i-1}$ which is not spanning, we find an edge w_0w_1 such that w_0 is in H but w_1 is not. Then we iteratively build a path P by using only vertices $w_1, w_2, \ldots$ not in H. When the iteration cannot proceed at some vertex z, we call this vertex the *critical vertex* of the iteration. At this point, we have either that $\Gamma(z)$ intersects with H, or that $\Gamma(z)$ is contained in $V(P) - w_0$. In the first case, we add this path $P_i := P + zv$ where $v \in V(H) \cap \Gamma(z)$ to H. In the second case, we shrink a cycle formed by a subset of $V(P) - w_0$. We terminate when we find a spanning subgraph.

Algorithm 1 states the details of algorithm 2-ECSS. The input graph is an r-edge connected multigraph $G = (V, E)$ of order n with m edges, where $r \geqslant 2$. The r-edge connectivity condition is needed in Theorem 3 because we require that the input G of every recursive call to 2-ECSS must have a minimum degree r. The other piece of initial input is the empty graph $H = (\emptyset, \emptyset)$. In each recursive step, H is a 2-edge connected, not necessarily spanning, subgraph of G. When there

Algorithm 1 (2-ECSS): Approximation Algorithm for the Minimum Size 2-ECSS Problem

Input: r-edge connected multigraph $G = (V, E)$ with no loop, and a 2-edge connected submultigraph H.
Output: 2-ECSS S of G such that H is a subgraph of S.

if H is a spanning subgraph of G **then**
  Return H. {Case A: H is a 2-ECSS.}
else if H is empty **then**
  Let P be any cycle in G. {Case B: Found the first nontrivial cycle.}
  Call algorithm 2-ECSS (G, P) to obtain S.
  Return S.
else

We have ($H \subsetneq V$).
Choose $w_0 \in H$ such that w_0 is adjacent to a vertex outside H.
$j := 0$.
Let P be the trivial path from w_0 to w_0.
while w_j has a neighbor w_{j+1} outside H and P **do**
  Add w_{j+1} to P.
  Increment j.
end while
if w_j has a neighbor w_{j+1} in H **then**
  Add w_{j+1} to P. {Case C1: Found an ear.}
  Call algorithm 2-ECSS ($G, H \cup P$) to obtain S.
  Return S.
else
  We have $\Gamma\{w_j\} \subset \{w_1, \ldots, w_j\}$. {Case C2: Found a cycle.}
  Let $h \geqslant 1$ be the smallest index such that w_h is a neighbor of w_j.
  Let C be the cycle $w_h, w_{h+1}, \ldots, w_j, w_h$.
  Shrink the cycle C into vertex v_C and get G_C.
  Call algorithm 2-ECSS (G_C, H) to obtain S.
  Return $S \cup C$.
end if
end if

is no ambiguity, we use H to refer to both H and $V(H)$.

4. Analysis: r-edge connected graphs

Theorem 3 holds trivially if $r = 2$. Continue the analysis by assuming $r \geqslant 3$.

Algorithm 2-ECSS terminates when it finds a spanning subgraph S. The algorithm also finds an ear decomposition of S such that each ear length is at least 2. Note that the size of $E(S)$ is $(n-1)$ plus the number of ears in the construction. Thus our goal is to give an upper bound on the number of ears using the sum of degrees.

To achieve this goal, we introduce a potential function Φ such that $\Phi(v)$ is an upper bound on the number of neighbors of v in $V(G) \setminus V(H)$. Initially set $\Phi(v) = deg(v)$. We will show that if H is neither empty nor spanning, then each iteration reduces $\sum_v \Phi(v)$ by $2(r-1+\sigma)$, where σ is the decrease in the size of $V(G) \setminus V(H)$.

Case A: We make no change in Φ.

Case B: For each $v \in P$ in the first nontrivial cycle, we decrease $\Phi(v)$ by 2. This case occurs only once to construct the first cycle P_1. Note that the length of cycle P is σ, and $\sum_v \Phi(v)$ decreases by 2σ.

Case C1: We decrease Φ by 1 for each end vertex of the ear P. (In case where both end vertices of P are the same, we decrease Φ of the end vertex by 2.) We decrease by 2 for each internal vertex of P. This accounts for edges in the newly added path P. Furthermore, for *each* edge $f = yz \in \delta(z) \setminus E(P)$ where z is the critical vertex, we subtract 1 from $\Phi(y)$ and from $\Phi(z)$. This accounts for edges which disappear from $G \setminus V(H)$. There are at least $r-2$ edges in $\delta(z) \setminus E(P)$ for r-edge connected graphs. Thus if l represents the length of P, then $\sum_v \Phi(v)$ decreases by at least $2l + 2(r-2)$. We note that σ is $l-1$.

Case C2: First, for each vertex $v \in C$ we decrease $\Phi(v)$ by 2. This accounts for edges in the cycle C. In addition, for *each* edge $f = yz \in \delta(z) \setminus E(C)$ where z is the critical vertex, we subtract 1 from $\Phi(y)$ and also from $\Phi(z)$. (Once again there are at least $r-2$ such edges.) This accounts for some chords in C. Then let $\Phi(v_C) = \sum_{w \in V(C)} \Phi(w)$. If we let l be the length of C, then $\sum_v \Phi(v)$ decreases by at least $2l + 2(r-2)$.

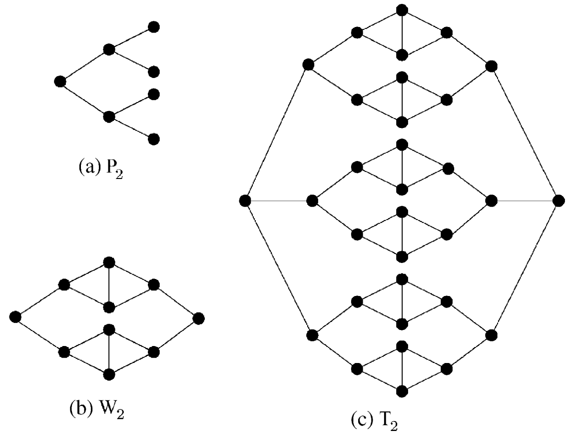

Fig. 1. Example showing the necessity of the r-edge connectivity condition.

We see that σ is $l-1$. Now we have the following proposition.

Proposition 2. *Let σ_i be the decrease in the size of $V(G)\setminus V(H)$ at iteration $i=1,2,\ldots$. Case B reduces $\sum_v \Phi(v)$ by $2\sigma_i = 2\sigma_1$. Each iteration of Cases* C1 *and* C2 *reduces $\sum_v \Phi(v)$* by $2(r-1+\sigma_i)$. *Also, $\Phi(v)$ is an upper bound on the number of neighbors of v in $V(G)\setminus V(H)$.*

Theorem 3. *Let G be an r-edge connected multigraph with n vertices and m edges, where $r \geqslant 2$. Then algorithm* `2-ECSS` *outputs a 2-ECSS S of G satisfying*

$$|E(S)| \leqslant n + \frac{m-n}{r-1}.$$

Proof. Suppose S is obtained after t iterations. Then S has $n-1+t$ edges. Since initially $\sum_{v\in V}\Phi(v) = 2m$ and Φ should stay nonnegative, Proposition 2 implies

$$2m \geqslant 2\sigma_1 + \sum_{i=2}^{t}(2(r-1+\sigma_i))$$

$$= 2(t-1)(r-1) + 2n.$$

By dividing this expression by $2(r-1)$ and rearranging it, we obtain the desired result. □

Corollary 4. *Let G be an r-edge connected r-regular multigraph with n vertices and m edges, where $r \geqslant 2$. Then we have*

$$|E(S)| \leqslant \left(1 + \frac{r-2}{2r-2}\right) n.$$

Therefore for $r = 3$, 4, 5 and 6, we have the approximation guarantees of $\frac{5}{4}, \frac{4}{3}, \frac{11}{8}$ and $\frac{7}{5}$, respectively.

For a special case where G is 3-edge connected and 3-regular, we can obtain the same result by a simpler algorithm using Menger's Theorem [1].

We remark that the r-edge connectivity condition cannot be discarded for the analysis of algorithm `2-ECSS`. Consider the following example. We denote by P_h the perfect binary tree where all the leaf vertices are at level h. Note that P_h has $2^{h+1}-1$ vertices and $2^{h+1}-2$ edges. We define the Web Graph W_h by taking two copies of P_h, identifying each pair of corresponding leaves, and adding an edge between a pair of identified "leaf" vertices that share the common parent in P_h. Furthermore, we define the Tri-Web T_h by taking three copies of W_h and introducing two degree 3 vertices, each of which is adjacent to three corresponding vertices of degree 2 in W_h. See Fig. 1 for illustration when $h=2$. We observe T_h has $9\cdot 2^h - 4$ vertices and $\frac{27}{2}\cdot 2^h - 6$ edges.

Note that if we run algorithm `2-ECSS` on T_h, which is 3-regular but not 3-edge connected, the algorithm may find and shrink all the 4-cycles at the start. In that case, the output S has $12\cdot 2^h - 6$ edges. Thus the ratio $\frac{|E(S)|}{|V(T_h)|}$ asymptotically approaches $\frac{4}{3} > \frac{5}{4}$.

Acknowledgements

The author would like to thank Joseph Cheriyan for inspiration and very useful discussions.

References

[1] B. Bollobás, Graph Theory: An Introductory Course, Springer, Berlin, 1979.

[2] J. Cheriyan, A. Sebő, Z. Szigeti, An improved approximation algorithm for minimum size 2-edge connected spanning subgraphs, in: R.E. Bixby, E.A. Boyd, R.Z. Ríos-Mercado (Eds.), Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science, Vol. 1412, Springer, Berlin, 1998, pp. 126–136.

[3] N. Garg, V.S. Santosh, A. Singla, Improved approximation algorithms for biconnected subgraphs via better lower

bounding techniques, in: Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, Austin, Texas, 25–27 January 1993, pp. 103–111.

[4] S. Khuller, U. Vishkin, Biconnectivity approximations and graph carvings, J. ACM 41 (2) (1994) 214–235.

[5] P. Krysta, V.S.A. Kumar, Approximation algorithms for minimum size 2-connectivity problems, in: Proceedings of the 18th International Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, 15–17 February 2001.

[6] S. Vempala, A. Vetta, Factor $\frac{4}{3}$ approximations for minimum 2-connected subgraphs, in: Proceedings of the Third Workshop on Approximation, Saarbrucken, Germany, 5–8 September 2000.

[7] H. Whitney, Nonseparable and planar graphs, Trans. Amer. Math. Soc. 34 (1934) 339–362.