A 2-Approximation Algorithm for Finding an Optimum 3-Vertex-Connected Spanning Subgraph

Vinchenzo Auletta   Yefim Dinitz[*][†]   Zeev Nutov[‡][§]

D. Parente

Abstract

The problem of finding a minimum weight k-vertex connected spanning subgraph in a graph $G = (V, E)$ is considered. For $k \geq 2$, this problem is known to be NP-hard. Combining properties of inclusion-minimal k-vertex connected graphs and of k-out-connected graphs (i.e., graphs which contain a vertex from which there exist k internally vertex-disjoint paths to every other vertex), we derive an auxiliary polynomial time algorithm for finding a $(\lceil \frac{k}{2} \rceil + 1)$-connected subgraph with a weight at most twice the optimum to the original problem. In particular, we obtain a 2-approximation algorithm for the case $k = 3$ of our problem. This improves the best previously known approximation ratio 3. The complexity of the algorithm is $O(|V|^3|E|) = O(|V|^5)$.

---

[*]Up to 1990, E. A. Dinic, Moscow.

[†]Dept. of Mathematics and Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel. *E-mail*: `dinitz@cs.bgu.ac.il`.

[‡]This work was done as a part of the author's D.Sc. thesis at the Dept. of Mathematics, Technion, Haifa, Israel.

[§]Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany. *E-mail*: `nutov@mpi-sb.mpg.de`.

1 Introduction

Connectivity is a fundamental property of graphs, which has important applications in network reliability analysis and network design problems. Recently, much effort has been devoted to problems of finding minimum cost subgraphs of a given weighted graph that satisfy given connectivity requirements (see [7] for a survey). A particular important class are the problems with uniform connectivity requirements, where the aim is to find a cheapest spanning subgraph which remains connected in presence of up to $k-1$ arbitrary edge or vertex failures (i.e., a minimum cost k-edge- or k-vertex-connected spanning subgraph, respectively). For the practical importance of the problem see, for example, Grötschel, Monma and Stoer [10]. In this paper we consider the vertex version[1] (henceforth we omit the prefix "vertex"), that is, the following problem:

Minimum weight k-connected subgraph problem: given an integer k and a k-connected graph with a nonnegative weight function on its edges, find its minimum weight k-connected spanning subgraph.

The case $k = 1$ is reduced to the problem of finding a minimum weight spanning tree. Beginning from $k = 2$, the minimum weight k-connected subgraph problem is known to be NP-hard. To see this, note that a 2-connected spanning subgraph of a graph G has $|V|$ edges if and only if G has a Hamiltonian cycle. A generalization to the case of any $k > 2$ is rather easy: let us add to such a G $k-2$ new vertices connected each to all vertices in the graph by edges of weight zero, arriving at an equivalent instance of a k-connected spanning subgraph problem.[2]

A few approximation algorithms are known for solving minimum weight k-connected subgraph problems (see [12] for a survey). An approximation algorithm is called *α-approximation*, or is said to achieve *approximation ratio* α, if it is a polynomial time algorithm that produces a solution of weight no more than α times the value of an optimal solution. For an arbitrary k, the best known approximation algorithm is due to Ravi and Williamson [18]; it achieves the approximation ratio $2H(k)$, where $H(k) = 1 + \frac{1}{2} + \ldots + \frac{1}{k}$ is

[1] For a survey on results concerning edge-connectivity see, for example, [12].

[2] Recently, Fernandes [6] showed that the minimum weight 2-*edge*-connected subgraph problem is MAX SNP-hard.

the kth Harmonic number. Note that, for the cases $k = 2, 3$, this algorithm achieves approximation ratios 3, $3\frac{2}{3}$, respectively.

For particular instances of the problem, there were obtained more efficient algorithms. For the case when edge weights satisfy the triangle inequality, a $(2 + \frac{2(k-1)}{n})$-approximation algorithm for an arbitrary k was suggested by Khuller and Raghavachari in [13]. Recently, Cheriyan and Thurimella [3] suggested a $(1 + \frac{1}{k})$-approximation algorithm for the problem of finding a minimum *size* k-connected spanning subgraph (i.e., a k-connected spanning subgraph with minimal number of edges), k arbitrary.

For a general instance of the minimum weight k-connected subgraph problem, approximation ratios better than in [18] were obtained for small values of k. Khuller and Raghavachari [13] developed a $(2 + \frac{1}{n})$-approximation algorithm for $k = 2$; it was improved to approximation ratio 2 in [17]. Penn and Shasha-Krupnik [17] introduced a 3-approximation algorithm for the case $k = 3$. A simpler and faster 3-approximation algorithm for $k = 3$ was developed in [16].

The main result of this paper is a 2-approximation algorithm for the minimum weight 3-connected subgraph problem. This improves the best previously known performance guarantee 3 [17, 16]. This is done by combining certain properties of minimally k-connected graphs, certain techniques from recent approximation algorithms [13, 17, 16], and some new ideas and techniques.

The complexity of the suggested algorithm is $O(n^5)$, where n is the number of vertices in the graph. Our algorithm can be applied for the case $k = 2$ as well; it has the same performance (approximation ratio 2 and complexity $O(n^5)$) as the algorithm in [17].

Based on this paper, the continuation paper [4] shows a 3-approximation algorithm for $k = 4, 5$, improving the previously best known approximation ratios $4\frac{1}{6}, 4\frac{17}{30}$, respectively. Recently, in [15], it was shown that the algorithms of these two papers can be combined with the algorithm of [18] to achieve a slightly better approximation guarantee than $2H(k)$ for all k.

This paper is organized as follows. In Section 2 we give notations and describe known results used in the paper. Section 3 studies k-out-connected graphs (i.e., graphs that have a vertex from which there exist k internally disjoint paths to any other vertex). In Section 4 we use properties of minimally k-connected graphs to derive a 2-approximation algorithm for the minimum weight 3-connected subgraph problem.

The preliminary versions of this paper are [1, 5].

2 Preliminaries and Notations

Let $G = (V, E)$ be an undirected simple graph (i.e., without loops and multiple edges) with vertex set V and edge set E. For a vertex v of a graph (resp., digraph) G we denote by $N_G(v)$ the set of neighbors of v in G, and by $d_G(v) = |N_G(v)|$ the *degree* (resp., *outdegree*) of v in G. In the case G is understood, we omit the subscript "G" in these notations.

A graph G with a nonnegative weight (cost) function w on its edges is referred to as a *weighted graph* and is denoted by (G, w), or simply by G if w is understood. For a weight function w and $E' \subseteq E$, we use the notation $w(E') = \sum\{w(e) : e \in E'\}$. For a subgraph $G' = (V', E')$ of a weighted graph (G, w), $w(G')$ is defined to be $w(E')$. A subgraph $G' = (V', E')$ is called *spanning* if $V' = V$; in this paper, we use only spanning subgraphs and, thus, sometimes omit the word "spanning". Similar notations are used for digraphs.

A subset $C \subseteq V$ is a (*vertex*) *cut* of a connected graph G if $G \setminus C$ is disconnected; if $|C| = k$ then such C is called a *k-cut*. A *side* of a cut C is the vertex set of a connected component of $G \setminus C$. A graph G is *k-connected* if it is a complete graph on $k + 1$ vertices or if it has at least $k + 2$ vertices and contains no l-cut with $l < k$. The *connectivity* of G, denoted by $\kappa(G)$, is defined to be the maximum k for which G is k-connected. In what follows we assume that $|V| \geq k + 2$; thus $\kappa(G)$ is the cardinality of a minimum cut of G.

A set of paths is said to be *internally disjoint* if no two of them have an internal vertex in common. Following [7], a graph (resp., digraph) such that there exist k internally disjoint paths from a certain vertex r to any its other vertex is said to be *k-out-connected from r*. The following statements are well known and can be easily deduced from Menger's Theorem: (i) in a graph which is k-out-connected from r, any l-cut with $l < k$, if such exists, must contain r; (ii) a graph G is k-connected if and only if it is k-out-connected from every vertex of G. The latter implies that, for any vertex r of a k-connected weighted graph, the weight of an optimal k-out-connected from r

spanning subgraph is less or equal to the weight of an optimal k-connected spanning subgraph.

A graph G is called *minimally k-connected* if $\kappa(G) = k$, but for any $e \in E$, $\kappa(G \setminus e) < k$. Observe that every k-connected graph contains a minimally k-connected spanning subgraph. Thus, among the subgraphs which are optimal solutions for the minimum weight k-connected subgraph problem, there always exists a minimally k-connected one.

Throughout the paper, let $\mathcal{G} = (\mathcal{G}, w)$ denote the input graph, n and m denote the number of its vertices and edges, respectively, and w^* denote the value of an optimal solution to our problem.

The *underlying graph* of a digraph D is the simple graph $U(D)$ obtained from D by replacing, for every $u, v \in V$, the set of arcs with endnodes u, v, if nonempty, by an edge (u, v). The *directed version* of a weighted graph (G, w) is the weighted digraph $D(G)$ obtained from G by replacing every undirected edge of G by the two antiparallel directed edges with the same ends and of the same weight. For simplicity of notations, we denote the weight function of $D(G)$ also by w.

Frank and Tardos [8] showed that for a directed graph, the problem of finding a minimum weight k-out-connected subdigraph from a given vertex r is solvable in polynomial time; a faster algorithm is due to Gabow [9]. This polynomial solvability was used as a basis for deriving approximation algorithms for several augmentation problems (see, for example, [13, 17, 16]). The main idea behind most of these algorithms is as follows. First, to add a new "external" vertex r and connect it by edges to certain k vertices of the input graph. Then, to find a minimum weight k-out-connected subdigraph from r in the directed version. It is shown in [13] that the underlying graph of thus obtained k-out-connected subdigraph, after deleting r, is $\lceil \frac{k}{2} \rceil$-connected and its weight is at most twice the weight of an optimal k-connected subgraph.[3] For $k = 2$, a slight modification of this technique gives a 2-connected subgraph [13, 17], while for $k = 3$, an additional set of edges is added to make thus obtained subgraph 3-connected [17, 16].

In our algorithm, we show a method to choose such r as a vertex of

[3]In the case of edge connectivity, the underlying graph of *any* k-edge-out-connected subgraph is k-edge-connected. This observation was used in [14] to derive a fast and simple 2-approximation algorithm for the minimum weight k-*edge*-connected subgraph problem, k arbitrary.

the *input* graph. This guarantees that the resulting subgraph is $(\lceil \frac{k}{2} \rceil + 1)$-connected. For the case $k = 3$ considered in this paper, $\lceil \frac{3}{2} \rceil + 1 = 3$, and our improvement produces a better approximation algorithm.

Roughly, our algorithm works as follows. Among all spanning subgraphs which are k-out-connected from a vertex of degree k,[4] the algorithm finds one of weight at most twice the value of an optimal solution to our problem. For $k = 3$, such a subgraph is 3-connected, and it is the output of the algorithm.

3 Properties of k-out-connected graphs

In this section we study k-out-connected graphs, $k \geq 2$. In particular, we show that if a graph is k-out-connected from a vertex of degree k, then it is $(\lceil \frac{k}{2} \rceil + 1)$-connected.

Our motivation to study k-out-connected graphs is that, in this paper, we choose to approximate a minimum weight k-connected spanning subgraph by a certain k-out-connected spanning subgraph. Observe, however, that an *arbitrary* k-out-connected graph is not necessarily even 2-connected. Indeed, let us take two complete graphs on at least k vertices each and connect an additional vertex r to some $t \geq k$ vertices in each of these two graphs. The resulting graph is k-out-connected from r, but not 2-connected (since $\{r\}$ is a 1-cut). Observe that the degree of r in this example is at least $2k$. One may ask whether lower degree of r guarantees higher connectivity. The following Lemma establishes a lower bound on the connectivity of a k-out-connected graph from r relatively to the degree of r (generalizing [13, Theorem 4.3]).

Lemma 3.1 *Let G be a k-out-connected graph from a vertex r, and let C be an l-cut of G with $l < k$. Then $r \in C$, and for any side S of C holds: $l \geq k - |S \cap N(r)| + 1$. Thus $\kappa(G) \geq k - \lfloor \frac{d(r)}{2} \rfloor + 1$.*[5]

Proof: The fact that r is in C was already established in Section 2.

[4]Here and further we mean the degree w.r.t. the subgraph.

[5]In fact, the bounds in Lemma 3.1 are tight in the following sense. For any $k \geq 2$ and $k \leq d \leq 2k$, there exists a graph which is k-out-connected from its vertex r of degree d and has connectivity exactly $k - \lfloor \frac{d}{2} \rfloor + 1$. Such a graph can be obtained by a generalization of the construction given above, as follows: we identify $k - \lfloor \frac{d}{2} \rfloor$ vertices of the two complete graphs and connect the additional vertex r to one common vertex (if d is odd) and to at least $\lfloor \frac{d}{2} \rfloor$ non common vertices of each one of the complete graphs.

Let now S be a side of C. If $k \leq |S \cap N(r)|$, then the statement is trivial, so assume $k > |S \cap N(r)|$. Let us choose a vertex $v \in S$ and consider a set of k internally disjoint paths between r and v. Since those paths begin with distinct edges, at most $|S \cap N(r)|$ of them may not contain a vertex from $C \setminus r$. This implies that every one of the other at least $k - |S \cap N(r)|$ paths must contain each at least one vertex from $C \setminus r$. These vertices are distinct, hence $l - 1 \geq k - |S \cap N(r)|$, as required. To see that $\kappa(G) \geq k - \lfloor \frac{d(r)}{2} \rfloor + 1$, observe that every cut of G has a side S for which $|S \cap N(r)| \leq \lfloor \frac{d(r)}{2} \rfloor$. $\square$

The highest connectivity that can be guaranteed by Lemma 3.1 for a k-out-connected graph from r corresponds to the lowest possible degree of r, which is k. For such graphs, Lemma 3.1 implies the following statement.

Corollary 3.2 *Let G be a k-out-connected graph from a vertex r of degree k, $k \geq 2$. Then G is $(\lceil \frac{k}{2} \rceil + 1)$-connected. In particular, if $k \in \{2, 3\}$, then G is k-connected.*

4 Minimally k-connected graphs and the minimum weight 3-subgraph problem

In this section we show how to find a subgraph which is k-out-connected from a vertex of degree k and has weight at most twice the value of an optimal k-connected subgraph. Combining this with Corollary 3.2, we arrive at a 2-approximation algorithm for the minimum weight 3-connected subgraph problem.

Our first aim is to establish that among optimal solutions to the minimum weight k-connected subgraph problem there always exists one which has a vertex of degree k (recall that its k-connectivity implies that it is k-out-connected from that vertex). This is straightforward by combining existence of an optimal solution graph which is minimally k-connected and the following theorem of Halin [11] (see also [2]).

Theorem 4.1 ([11]) *Any minimally k-connected graph has a vertex of degree k.*

Remark. Let us define a *minimally k-out-connected graph* as a k-out-connected graph G such that, for every its edge e, $G \backslash e$ is not k-out-connected. Then, combining Theorem 4.1 with Corollary 3.2, we obtain an interesting characterization of minimally 2 and 3-connected graphs: *For $k \in \{2, 3\}$, a graph is minimally k-connected if and only if it is minimally k-out-connected from a vertex of degree k.*

Let w^* denote the weight of an optimal k-connected subgraph. We now suggest an algorithm that finds a *subgraph which is k-out-connected from a vertex of degree k and has weight at most* $2w^*$ (using the approach of [14, 13], where it was shown how to find such a subgraph but without the degree constraint). We use the following simple observation:

Fact 4.2 *A graph G' is k-out-connected from a vertex r if and only if its directed version $D(G')$ is k-out-connected from r, or, which is equivalent, $D(G')$ without the edges entering r is k-out-connected from r.*

Before presenting our algorithm, let us consider the following auxiliary problem. Let $(\mathcal{D}, w)$ be a weighted digraph and r a vertex of $\mathcal{D}$. Among all k-out-connected from r subdigraphs of $\mathcal{D}$ such that r has outdegree k in them, if any, find one of the minimal weight. Using penalty methods, this problem can be easily reduced to the problem of finding an optimal k-out-connected subdigraph (and thus solved by a single run of algorithm [9]) as follows. Let $M = w(\mathcal{D}) + 1$, and let w_r be the weight function obtained from w by adding M to the weight of each arc incident to r. Let us consider a minimum weight k-out-connected subdigraph from r in $(\mathcal{D}, w_r)$, say, D_r; clearly, there are no arcs incoming r in it. Observe that, by the definition of M, for any two subgraphs D' and D'' of $\mathcal{D}$ holds: (i) if $d_{D'}(r) < d_{D''}(r)$ then $w_r(D') < w_r(D'')$ and (ii) if $d_{D'}(r) = d_{D''}(r)$ then $w_r(D') \leq w_r(D'')$ if and only if $w(D') \leq (D'')$. This implies that if the outdegree of r in D_r is k, then D_r is an optimal solution to the discussed problem; otherwise, this problem has no feasible solution.

Let us return to the original problem. Our algorithm solves the above auxiliary problem in the directed version $\mathcal{D} = D(\mathcal{G})$ of $\mathcal{G}$ for every vertex r; it outputs the cheapest one among the underlying graphs of the subdigraphs D_r constructed as solutions to these problems.

Out-Connected Subgraph Algorithm (OCSA)

Input: A weighted graph $(\mathcal{G}, w)$, $\mathcal{G} = (V, E)$, and an integer k.

Output: A subgraph $\tilde{G}$ of $\mathcal{G}$ and a vertex $\tilde{r}$, such that $\tilde{G}$ is k-out-connected from $\tilde{r}$ and $d_{\tilde{G}}(\tilde{r}) = k$, if exists.

Set $\tilde{G}, \tilde{r}$ undefined, $\tilde{w} = \infty$, $M = 2w(\mathcal{G}) + 1$;
For every vertex $r \in V$ *do*:
(1) Set $w_r(e) = w(e) + M$ if e is incident to r, and $w_r(e) = w(e)$ otherwise;
(2) Find a minimum weight k-out-connected from r subdigraph D_r of $D(\mathcal{G}, w_r)$, if such exists, by the algorithm [9];
(3) *If* the degree of r in $U(D_r)$ is k *and* $w(U(D_r)) < \tilde{w}$, *then* set $\tilde{G} = U(D_r)$, $\tilde{r} = r$, and $\tilde{w} = w(U(D_r))$;
end for
If $\tilde{w} < \infty$ *then* output $\tilde{G}$, $\tilde{r}$
else declare "$\mathcal{G}$ contains no subgraph which is k-out-connected from a vertex of degree k";

Lemma 4.3 *For any integer $k \geq 1$ and any weighted graph $\mathcal{G}$ that contains a spanning subgraph which is k-out-connected from a vertex of degree k, OCSA outputs such a subgraph of weight at most twice the minimal possible. The complexity of OCSA is $O(k^2 n^3 m)$.*

Proof: Let G' be a k-out-connected from a vertex of degree k (say, r') spanning subgraph of $\mathcal{G}$ with the minimal weight (say, w'). At some iteration, the algorithm chooses $r = r'$. Observe that the subdigraph $D(G')$ of $D(\mathcal{G})$ is (i) k-out-connected from r' (by Fact 4.2), and (ii) the outdegree of r' in it is exactly k. By the above discussion, the constructed subgraph $D_{r'}$ is an optimal one among the subgraphs of $D(\mathcal{G})$ with these two properties, hence $w(D_{r'}) \leq w(D(G'))$. Therefore, after this iteration $\tilde{G}$ and $\tilde{r}$ are defined, and

$$\tilde{w} \leq w(U(D_{r'})) \leq w(D_{r'}) \leq w(D(G')) = 2w(G') = 2w'.$$

Thus, OCSA outputs a pair $(\tilde{G}, \tilde{r})$, where $w(\tilde{G}) \leq 2w'$.

Observe that at any iteration of the algorithm in which the pair $(\tilde{G}, \tilde{r})$ is updated, the properties $d_{\tilde{G}}(\tilde{r}) = k$ and $\tilde{G}$ is k-out-connected from $\tilde{r}$ are maintained. Thus, the same is valid at the end of the algorithm for the output $(\tilde{G}, \tilde{r})$, as required.

We now show the time complexity. The dominating time is spent for finding subdigraphs D_r. The time complexity of the algorithm [9] is $O(k^2n^2m)$, and the number of its executions in OCSA is n. The complexity $O(k^2n^3m)$ follows. $\square$

Theorem 4.4 *For any integer $k \geq 2$ and any weighted k-connected graph $\mathcal{G}$, OCSA outputs a $(\lceil \frac{k}{2} \rceil + 1)$-connected spanning subgraph of $\mathcal{G}$ of weight at most $2w^*$, in time $O(k^2n^3m)$.*

Proof: Let G^* be any minimally k-connected optimal subgraph of $\mathcal{G}$; its weight is w^*. By Theorem 4.1, there exists a vertex r^* which has degree k in G^*; note that G^* is k-out-connected from r^*. Lemma 4.3 implies that the subgraph output by OCSA has weight at most $2w^*$ and that it is $(\lceil \frac{k}{2} \rceil + 1)$-connected, by Corollary 3.2. The time bound is implied by Lemma 4.3. $\square$

Since for $k = 2, 3$ holds $\lceil \frac{k}{2} \rceil + 1 = k$, the above discussion implies our main result, as follows.

Theorem 4.5 *For $k \in \{2, 3\}$, OCSA is a 2-approximation algorithm for the minimum weight k-connected subgraph problem, with complexity $O(mn^3) = O(n^5)$.*

Acknowledgment

The authors thank Joseph Cheriyan for suggesting the penalty method as a way to improve the time complexity of our algorithm and for his other useful comments.

References

[1] V. Auletta and D. Parente: Better algorithms for minimum-weight connectivity problems. *Proc. 14th Annual Symposium on Theoretical Aspects of Computer Science (STACS '97)*, R. Reischuk and M. Morvan Eds., Lecture Notes in Comp. Sci., Vol. 1200, Springer-Verlag, Berlin

(1997), 547–558. (See also Tech. Rep. Universita' di Salerno 6/96 (July 1996).)

[2] B. Bollobás: *Extremal graph theory, Chapter I.* Academic Press, London (1978).

[3] J. Cheriyan and R. Thurimella: Approximating Minimum-Size k-Connected Spanning Subgraphs via Matching. *Proc. 37th Annual IEEE Symp. on Foundations of Comp. Sci.*, Oct. 1996, 292–301.

[4] Ye. Dinitz and Z. Nutov: A 3-Approximation Algorithm for Finding Optimum 4,5-Vertex-Connected Spanning Subgraphs. *J. of Algorithms*, this issue, ??–??.

[5] Ye. Dinitz and Z. Nutov: Finding minimum weight k-vertex connected spanning subgraphs: approximation algorithms with factor 2 for $k = 3$ and factor 3 for $k = 4, 5$. *Proc. 3rd Italian Conf. on Algorithms and Complexity, CIAC'97* (Rome, March 1997), Lecture Notes in Comp. Sci., v. 1203, Springer-Verlag, 1997, 13–24.

[6] C. G. Fernandes: A better approximation ratio for the minimum k-edge-connected spanning subgraph problem, Proc. 8th Annual ACM-SIAM Symp. on Discrete Algorithms, Jan. 1997, 629–638.

[7] A. Frank: Connectivity augmentation problems in network design. *Mathematical Programming, the State of the Art.* J. R. Birge and K. G. Murty eds. (1994), 34–63.

[8] A. Frank and E. Tardos: An application of submodular flows. *Linear Algebra and its Application*, **114/115** (1989), 329–348.

[9] H. N. Gabow: A representation for crossing set families with application to submodular flow problems. *Proc. 4th Annual ACM-SIAM Symp. on Discrete Algorithms* (1993), 202–211.

[10] M. Grötschel, C. Monma and M. Stoer: *Design of survivable networks. Handbook in Operation Research and Management Science, Vol. 7, Network Models.* M. O. Ball, T. L. Magnanti, C. L. Monma and G. L. Nemhauser eds. (1993).

[11] R. Halin: A theorem on n-connected graphs. *J. Combinatorial Theory* **7** (1969), 150–154.

[12] S. Khuller: Approximation algorithms for finding highly connected subgraphs. In *Approximation algorithms for NP-hard problems*, D. S. Hochbaum ed., PWS Publishing Co., Boston, 1996, 236–265.

[13] S. Khuller and B. Raghavachari: Improved approximation algorithms for uniform connectivity problems. *J. of Algorithms* **21**, (1996), 434–450.

[14] S. Khuller and R. Thurimella: Approximation algorithms for graph augmentation. *Journal of Algorithms* **14** (1993), 214–225.

[15] Z. Nutov: Improved approximation algorithms for finding optimum k-vertex-connected spanning subgraphs. TR CORR 98-03, Dept. of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada, Feb. 1998.

[16] Z. Nutov and M. Penn: Faster approximation algorithms for weighted triconnectivity augmentation problems. *Operations Research Letters* **21**, (1997), 219–223.

[17] M. Penn and H. Shasha-Krupnik: Improved approximation algorithms for weighted 2 & 3 vertex connectivity augmentation problems. *J. of Algorithms* **22** (1997), 187–196.

[18] R. Ravi and D. P. Williamson: An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica* **18**, (1997), 21–43.