A Unified Approach for Approximating 2-Edge-Connected Spanning Subgraph and 2-Vertex-Connected Spanning Subgraph

Ali Çivril*

June 4, 2024

Abstract

We provide algorithms for the minimum 2-edge-connected spanning subgraph problem and the minimum 2-vertex-connected spanning subgraph problem with approximation ratio $\frac{4}{3}$.

1 Introduction

We consider two fundamental connectivity problems, namely the minimum 2-edge-connected spanning subgraph problem (2-ECSS) and the minimum 2-vertex-connected spanning subgraph problem (2-VCSS).

In 2-ECSS, given an undirected simple graph $G = (V, E)$, one finds a 2-edge-connected spanning subgraph (2-ECSS) of G with minimum number of edges. The problem remains NP-hard and MAX SNP-hard even for subcubic graphs [7]. The first result improving the approximation factor 2 came from Khuller and Vishkin [15], which is a $\frac{3}{2}$-approximation algorithm. Cheriyan, Sebö and Szigeti [2] improved the factor to $\frac{17}{12}$. Vempala and Vetta [19], and Jothi, Raghavachari and Varadarajan [14] claimed to have $\frac{4}{3}$ and $\frac{5}{4}$ approximations, respectively. Krysta and Kumar [16] went on to give a $(\frac{4}{3} - \epsilon)$-approximation for some small $\epsilon > 0$ assuming the result of Vempala and Vetta [19]. A relatively recent paper by Sebö and Vygen [18] provides a $\frac{4}{3}$-approximation algorithm by using ear decompositions, and mentions that the aforementioned claimed approximation ratio $\frac{5}{4}$ has not appeared with a complete proof in a fully refereed publication. The result of Vempala and Vetta [19], which was initially incomplete, recently re-appeared in [13]. Given this, it is not clear if the ratio $(\frac{4}{3} - \epsilon)$ by Krysta and Kumar [16] still holds. Very recently, Garg, Grandoni, and Ameli [9] improved upon the ratio $\frac{4}{3}$ by a constant $\frac{1}{130} > \epsilon > \frac{1}{140}$, which stands as the best current result for approximating the problem.

In 2-VCSS, given an undirected simple graph $G = (V, E)$, one finds a 2-vertex-connected (or simply 2-connected) spanning subgraph (2-VCSS) of G with minimum number of edges. This problem is NP-hard via a reduction from the Hamiltonian cycle problem. Furthermore, by the result of Czumaj and Lingas [8], it does not admit a PTAS unless P = NP. The first result improving the factor 2 for 2-VCSS came from Khuller and Vishkin [15], which is a $\frac{5}{3}$-approximation algorithm. Garg, Vempala, and Singla [10] improved the approximation ratio to $\frac{3}{2}$. Cheriyan and Thurimella [3] also provides the same approximation ratio in a more general context, where they consider k-connectivity. Vempala and Vetta [19] claimed the ratio $\frac{4}{3}$, which is shown to be not valid by Heeger and Vygen [12]. Jothi, Raghavachari and Varadarajan [14] claimed the ratio $\frac{5}{4}$.

*Istanbul Atlas University, Computer Engineering Department, Kagithane, 34408 Istanbul, Turkey, e-mail: ali.civril@atlas.edu.tr

However, this claim has been later withdrawn (see [11]). Gubbala and Raghavachari [11] claimed to have a $\frac{9}{7}$-approximation algorithm. The only complete (and exceedingly long) proof of this claim is in Gubbala's thesis [17], which has not appeared anywhere else. To the best of our knowledge, the algorithm by Heeger and Vygen [12] with factor $\frac{10}{7}$ stands as the first refereed improvement over the factor $\frac{3}{2}$ after a long hiatus. This paper also contains a somewhat more detailed discussion about the aforementioned claimed results, implying that theirs is the first improvement. Very recently, Bosch-Calvo, Grandoni, and Ameli provided an algorithm with approximation ratio $\frac{4}{3}$ [1], the current best factor.

Although the two problems we consider are closely related to each other, as noted above, the work on these problems have generally appeared in distinct publications. The reason is that the solution being 2-edge-connected or 2-connected enforces different approaches if the algorithmic ideas are tightly coupled with the structure of the solution. We give a unified strategy to solve these problems, where we use an algorithmic idea almost oblivious to whether the solution should be 2-edge-connected or 2-connected. The starting point is [6] in which we gave an algorithm for 2-ECSS. Unfortunately, its analysis therein is not clear, and contains several inconsistencies, even with the corrigendum published after the paper [5]. The purpose of this paper is twofold: We provide a new algorithm for 2-ECSS, which starts from the ideas employed in [6], but with a clear analysis, thereby obtaining the ratio $\frac{4}{3}$. Secondly, we show that the same algorithmic idea can be applied to 2-VCSS to attain the same approximation ratio $\frac{4}{3}$, matching the current best approximation ratio. Both the algorithms and their analyses are considerably simpler compared to previous approaches. In essence, we do not use anything but a recursive local search over an inclusion-wise minimal starting solution.

2 Preliminaries

For both of the problems, we will use the lower bound derived from the dual of the natural LP relaxation for 2-ECSS. Here, $\delta(S)$ denotes the set of edges with one end in the cut S and the other not in S.

$$\begin{aligned} \text{minimize} \quad & \sum_{e\in E} x_e & & \text{(EC)} \\ \text{subject to} \quad & \sum_{e\in\delta(S)} x_e \geq 2, & \forall\, \emptyset \subset S \subset V, & \\ & 1 \geq x_e \geq 0, & \forall\, e \in E. & \end{aligned}$$

The following is the dual of (EC).

$$\begin{aligned} \text{maximize} \quad & \sum_{\emptyset\subset S\subset V} 2y_S - \sum_{e\in E} z_e & & \text{(EC-D)} \\ \text{subject to} \quad & \sum_{S:e\in\delta(S)} y_S \leq 1 + z_e, & \forall e \in E, & \\ & y_S \geq 0, & \forall\, \emptyset \subset S \subset V, & \\ & z_e \geq 0, & \forall\, e \in E. & \end{aligned}$$

We now give the LP relaxation for 2-VCSS from [4], which will also be of use in one of our arguments. Set $n := |V|$, $m := |E|$. A setpair is an ordered pair of sets $W = (W_t, W_h)$ such that

$W_t \subseteq V$, $W_h \subseteq V$ and $W_t \cap W_h = \emptyset$. We say that an edge $(u, v) \in E$ *covers* 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 E that cover W, and $\mathcal{S}$ denote the set of all setpairs (W_t, W_h) such that W_t and W_h are non-empty. The following is an LP relaxation for 2-VCSS.

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} x_e & & \text{(VC)} \\ \text{subject to} \quad & \sum_{e \in \delta(W)} x_e \geq 2, & W \in \mathcal{S}, |W_t \cup W_h| = n, & \\ & \sum_{e \in \delta(W)} x_e \geq 1, & W \in \mathcal{S}, |W_t \cup W_h| = n-1, & \\ & 1 \geq x_e \geq 0, & \forall\, e \in E. & \end{aligned}$$

We assume that the input graph G is 2-connected. Otherwise, the algorithm of the next section for 2-ECSS can be executed on blocks (maximal 2-connected subgraphs) of G separately. This is without loss of generality, since in that case the value of an optimal solution for 2-ECSS is the sum of those of blocks, and we can argue the approximation ratio only within a block. Given a vertex $v \in V$ and a feasible solution F, the degree of v on F is denoted by $deg_F(v)$. The vertex v is called a *degree-d vertex* on F if $deg_F(v) = d$, and a *high-degree vertex* on F if $deg_F(v) \geq 3$. For a path $P = v_1 v_2 \ldots v_{k-1} v_k$, v_1 and v_k are the *end vertices* of P, and all the other vertices are the *internal vertices* of P. A path whose internal vertices are all degree-2 vertices on F is called a *plain path* on F. A maximal plain path is called a *segment*. The length of a segment is the number of edges on the segment. If the length of a segment is ℓ, it is called an *ℓ-segment*. A 1-segment is also called a *trivial segment*. If a 2-ECSS remains feasible upon removal of an edge of a trivial segment, the edge is called *redundant*. For $\ell > 1$, an ℓ-segment is called a *non-trivial segment*. A non-trivial ℓ-segment with $\ell \leq 3$ is called a *short segment*, otherwise a *long segment*. Given a 2-VCSS F, if the removal of a segment from F violates feasibility, it is called a *weak segment* on F, otherwise a *strong segment* on F. If the end vertices of a segment are identical, it is called a *closed segment*.

3 The Algorithm for 2-ECSS and 2-VCSS

The algorithm starts by computing an inclusion-wise minimal 2-VCSS on G (Recall our assumption from the previous section). This can be computed by taking all the edges in E, and then deleting an element of this set one by one as long as the feasibility is not violated. Let F be such a solution. The algorithm then recursively modifies the running solution F via *improvement processes* to eliminate specific sets of edges from $E \setminus F$. More specifically, given an edge $e \in E \setminus F$, if there exists a set H of at least two edges in F such that $F \cup \{e\} \setminus H$ is feasible, then the operation switching from F to $F \cup \{e\} \setminus H$ is called an *improvement operation of size 1*. Given edges $e, f \in E \setminus F$, if there exists a set H of at least three edges in F such that $F \cup \{e, f\} \setminus H$ is feasible, then the operation switching from F to $F \cup \{e, f\} \setminus H$ is called an *improvement operation of size 2*. An operation is called an *improvement operation* if it is either of size 1 or size 2, and the included edge set is called a *critical edge set*. Given a strong short segment S on F and an internal vertex u of S, let $N(u) \subseteq E \setminus F$ denote the set of edges incident to u, which are not in F. In each iteration of a loop, the algorithm checks for a selected S and u if there is an improvement operation that includes an edge set from $N(u)$ to F. It switches to the cheaper 2-VCSS implied by the improvement operation if there is. Note that the possible sets of edges that might lead to an improvement operation can be examined in polynomial-time, as their sizes are constant. For explicitness, we list all the 5 types of improvement operations and the corresponding critical edge sets in Figure 1-Figure 5.

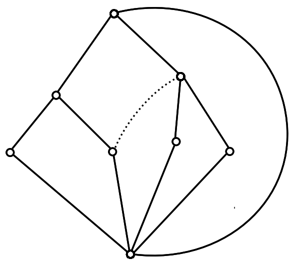


Figure 1: An example of an improvement operation of size 1

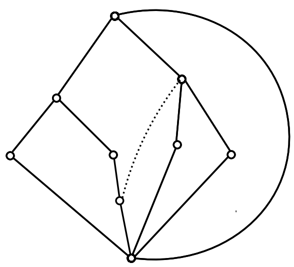


Figure 2: An example of an improvement operation of size 1

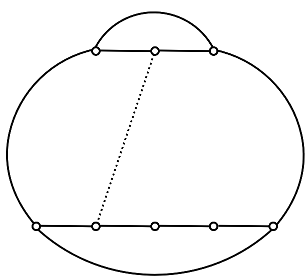


Figure 3: An example of an improvement operation of size 1

If there is no improvement operation that can be performed on u, fixing a critical edge set I in $N(u)$ to include, the algorithm attempts improvement operations via the strong short segments on $F \cup I$ that do not exist on F. In particular, it calls the procedure described above for S and

Algorithm 1: 2-ECSS-2-VCSS($G(V, E)$)

// Initialization
Let F be an inclusion-wise minimal 2-VCSS of G

// Improvement processes
while *there is a strong short segment S on F and an internal vertex u of S on which an improvement process has not been called* **do**
| Improvement-Process(F, S, u)

// Clean-up for 2-ECSS
if *computing a 2-ECSS* **then**
| Delete redundant edges from F

return F

Algorithm 2: IMPROVEMENT-PROCESS(F, S, u)

if *there is an improvement operation that can be performed on* u **then**
  Apply the improvement operation on u
  return
for *each set of critical edges* I *on* u **do**
  Let $\mathcal{S}$ be the set of strong short segments on $F \cup I$ that do not exist on F
  for *each strong short segment* T *in* $\mathcal{S}$ *and each internal vertex* v *of* T **do**
    if *no improvement process has been called on* (T, v) **then**
      IMPROVEMENT-PROCESS($F \cup I, T, v$)
      if *there is an improvement operation performed in* IMPROVEMENT-PROCESS($F \cup I, T, v$) **then**
        Perform deletion operation on $F \cup I$ in the order F, I
        return
if *there is no improvement operation performed in any of the calls above* **then**
  Restore F to the original set considered before the function call

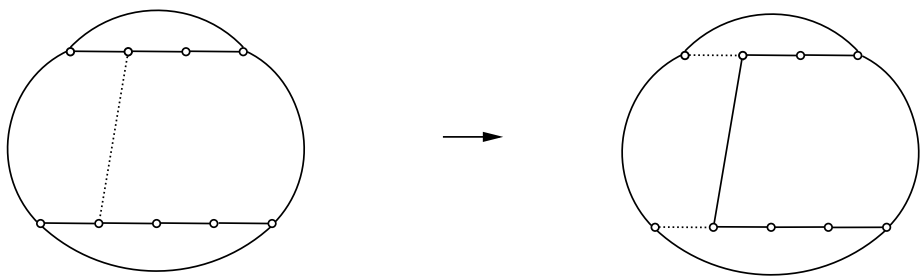


Figure 4: An example of an improvement operation of size 1

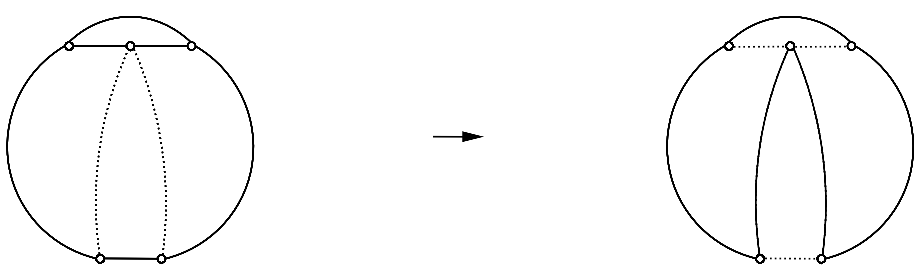


Figure 5: An example of an improvement operation of size 2

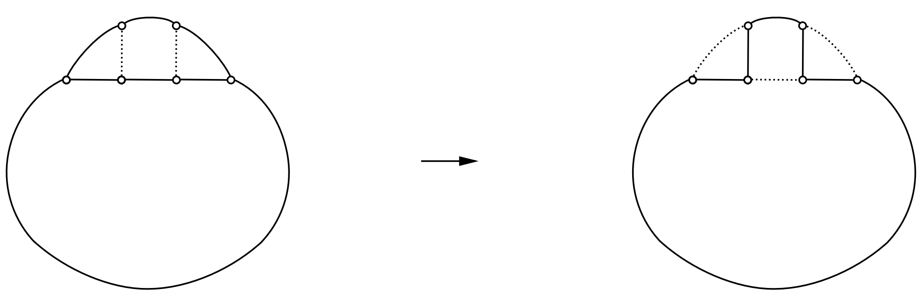


Figure 6: An example of an improvement process of recursion depth 2

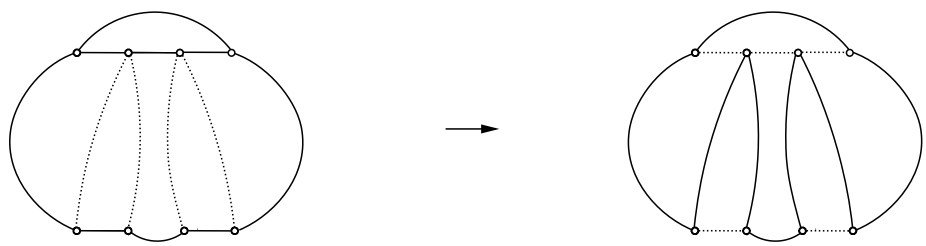


Figure 7: An example of an improvement process of recursion depth 2

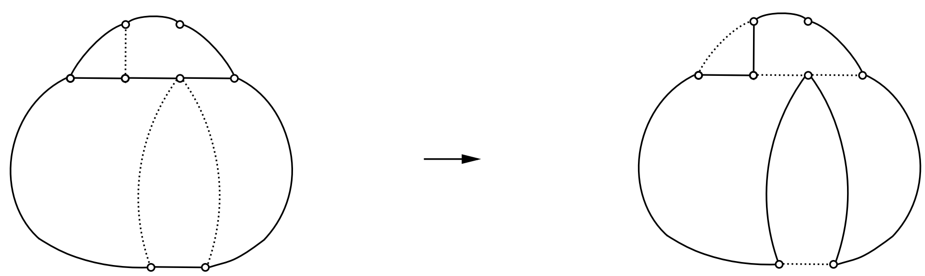


Figure 8: An example of an improvement process of recursion depth 2

u *recursively* on the internal vertices of these segments provided that no improvement process has been previously called on a given segment and an internal vertex. These calls are performed for all H on u and for each internal vertex $u \in S$. If there is an improvement operation in one of these calls, the called function returns and the caller performs a specific *deletion operation* as follows. It attempts to delete the edges from $F \cup I$ in the order F, I, where the order within the sets F and I are immaterial. Specifically, it deletes an edge as long as the residual graph remains a 2-VCSS. This enforces to keep the edges in I in the solution. Examples of these operations are given in Figure 6-Figure 8, where the depth of the recursion tree is 2. After the reverse-delete operation, the current function call returns. If no improvement operation is performed in any of the calls, the solution F is restored to the original one before the function call on u. The iterations continue until there is no strong short segment S and u on which we can perform an improvement process.

This completes the definition of the algorithm for 2-VCSS. For 2-ECSS, one final step is performed, which excludes all the redundant trivial edges, i.e., the trivial segments whose removal does not violate 2-edge-connectivity. The discussion of exact running times and efficient algorithms are not the focus of this paper. We only argue about polynomial-time solvability.

Proposition 1. *Algorithm 1 runs in polynomial-time.*

Proof. It suffices to see that the main loop of Improvement-Process terminates in polynomial number of operations. As noted, there are polynomially many sets I, since $|I|$ is constant. Starting from an internal vertex u of a strong short segment S, consider the recursion tree in which each node represents a recursive function call. By definition, each node of this tree is associated to an internal vertex of a strong short segment. A vertex can be a side vertex of constant number of strong segments. Since an improvement process is called at most once on a strong short segment-internal vertex pair, this implies that the number of nodes in the tree is polynomially bounded. So the algorithm terminates in polynomial number of operations. □

4 Proof of the Approximation Factor 4/3 for 2-ECSS and 2-VCSS

Lemma 2. *If F is a 2-ECSS returned by Algorithm 1, then a short segment S on F is not a closed segment.*

Proof. Recall that the input graph is assumed to be 2-connected. This implies that if S is a closed segment with an internal vertex u and the end vertex v, then there exists $w \in V$ with $(u, w) \in E \backslash F$. In this case however, the algorithm includes such an edge into the solution instead of (u, v), which is a contradiction. □

Let F be a solution returned by Algorithm 1, and $opt(G)$ denote the value of an optimal 2-ECSS on G.

Lemma 3. *There exists G_1, a 2-VCSS $F_1 \subseteq E(G_1)$, and a 2-ECSS $F_2 \subseteq E(G_1)$ such that the following hold:*

1. *F_1 is minimal with respect to inclusion.*
2. *F_2 is obtained from F_1 by deleting redundant edges.*
3. *For any internal vertex s of a strong short segment S on F_1, there is no edge $e \in E(G_1) \setminus F_1$ incident to s.*
4. *F is a 2-VCSS and $\frac{|F_1|}{opt(G_1)} \leq \frac{4}{3} \Rightarrow \frac{|F|}{opt(G)} \leq \frac{4}{3}$. F is a 2-ECSS and $\frac{|F_2|}{opt(G_1)} \leq \frac{4}{3} \Rightarrow \frac{|F|}{opt(G)} \leq \frac{4}{3}$.*

Proof. We reduce G to G_1 and F to F_1 by performing a series of operations. Let S be a strong short segment on F, and s be an internal vertex of S. Let O be an optimal 2-ECSS on G. Clearly, O contains two edges incident to s, say e_1 and e_2. Assume it contains a third edge e_3 incident to s. Let the other end vertices of these edges be w_1, w_2, and w_3, respectively. By the improvement operations performed by the algorithm, none of these vertices is an internal vertex of a strong short segment. If O contains all the edges incident to w_i that are in F, we call w_i a *special vertex*, for $i = 1, 2, 3$.

Claim 4. *There is at most one special vertex in the set $\{w_1, w_2, w_3\}$.*

Proof. Assume without loss of generality that w_1 and w_2 are special vertices. Then by the structure of a 2-ECSS, we can discard e_1 or e_2 from O without violating feasibility, which contradicts its optimality. □

Claim 5. *There exists an optimal 2-ECSS O' on G such that O' contains 2 edges incident to s.*

Proof. By Claim 4, there are at least two vertices in the set $\{w_1, w_2, w_3\}$ that are not special. Let two of them be without loss of generality w_2 and w_3. By the structure of a 2-ECSS, one of these vertices, say w_2, satisfies the following. There is a neighbor w_2' of w_2 such that $f = (w_2, w_2') \in F \backslash O$, and $O' = O \cup \{f\} \setminus \{e_2\}$ is another optimal solution. In this case the degree of s on O' is 2, which completes the proof. □

Let $O'(S)$ be the set of edges in this solution incident to the internal vertices of S, and let $F' = F \cup O'(S) \setminus P$ be a minimal 2-VCSS on G, where $P \subseteq F \setminus O'(S)$. Examples of this operation are illustrated in Figure 9 and Figure 10. Let $E(S)$ denote the set of edges incident to the internal vertices of S on $E(G)$, which excludes the edges in $O'(S)$. Note that this set contains P. Delete the edges in $E(S)$ from G to obtain G'. Perform these operations, including the switch to an optimal solution implied by Claim 5, recursively on the new strong short segments that appear on F', which

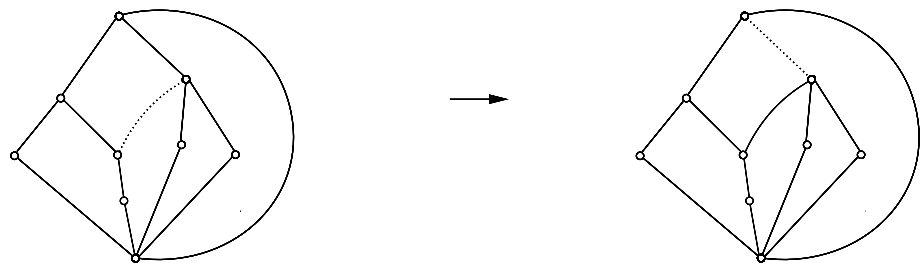


Figure 9: A transition from F to F_1 with $|P| = 1$

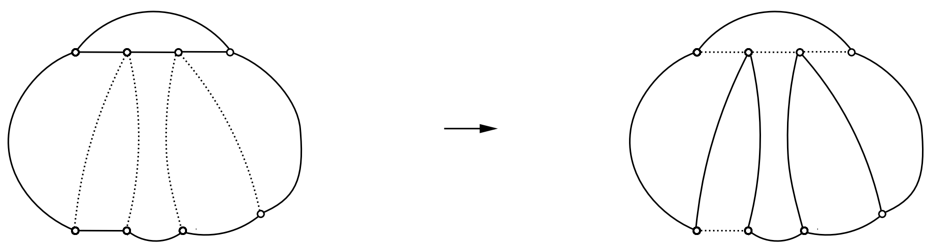


Figure 10: A transition from F to F_1 with $|P| = 4$

we call *emerging segments*. Note that since none of the aforementioned vertices w_1, w_2, and w_3 can be an internal vertex of a strong short segment as noted, the switch from O to O' cannot be reversed. After the recursion starting from S terminates, continue performing the described operations on the strong short segments on the residual solution and the graph. Let the results be F_1 and G_1. Given these, the first claim of the lemma follows from Claim 5. The third claim of the lemma holds, since there is no edge in $E(G_1) \setminus F_1$ incident to the internal vertices of a strong short segment on F_1 by construction. Let F_2 be a 2-ECSS obtained by deleting redundant edges from F_1, thus satisfying the second claim of the lemma. We now show that the fourth claim holds.

Claim 6. *If F is a 2-VCSS, then $|F_1| \geq |F|$. If F is a 2-ECSS, then $|F_2| \geq |F|$.*

Proof. Let S be a strong short segment on which we start the recursive operations above or an emerging segment. If F is a 2-VCSS, it suffices to see that $|P| \leq |O'(S)|$ in the construction described above. Note that by the algorithm and the construction of F_1, there is no improvement process performed on S that has improved the cost of the solution. Given this, $|P| > |O'(S)|$ derives a contradiction. In particular by all the listed improvement operations, we cannot have the configurations on the left hand sides of Figure 1-Figure 8. Note next that if e is an edge deleted from F_1 to obtain F_2, then we also have that e belongs to the set of redundant edges deleted to obtain a 2-ECSS F by the algorithm in the last step. Given this, the first assertion that $|F_1| \geq |F|$ when F is a 2-VCSS implies $|F_2| \geq |F|$ when F is a 2-ECSS. This completes the proof. □

We next note that $opt(G_1) \leq opt(G)$. This follows from our construction ensuring that there is an optimal solution O such that for any strong short segment S on F_1, $E(S)$ does not contain any edge from O. Combining this with Claim 6, we obtain $\frac{|F|}{opt(G)} \leq \frac{|F_1|}{opt(G_1)}$, which establishes the fourth claim of the lemma, and completes the proof. □

Let G_1, F_1, and F_2 be as implied by Lemma 3.

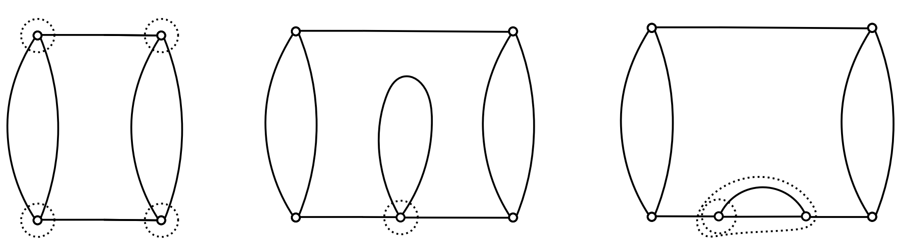
Lemma 7. *If F is a 2-VCSS, then $\frac{|F_1|}{opt(G_1)} \leq \frac{4}{3}$. If F is a 2-ECSS, then $\frac{|F_2|}{opt(G_1)} \leq \frac{4}{3}$.*

Proof. We construct a feasible dual solution in (EC-D) with total value at least $\frac{3}{4}|F_1|$ if F is a 2-VCSS, and $\frac{3}{4}|F_2|$ if F is a 2-ECSS. Given a strong short segment S on F_1 and an internal vertex s of S, assign $y_{\{s\}} = 3/4$. If the segment is a 3-segment, let $z_e = 1/2$ for the middle edge e of the segment to maintain feasibility. Let $y_{\{w\}} = 1/2$ for the internal vertices w of strong long segments and weak segments on F_1. Note that the overall assignment is feasible by the third claim of Lemma 3.

We distinguish a dual value we assign and its contribution in the objective function of (EC-D), which is twice the dual value. The latter is called the *dual contribution*, and the associated dual is said to *contribute* a certain value. We use a cost sharing argument, so that the cost of each segment is countered with a unique set of dual contributions with ratio at least $\frac{3}{4}$, which establishes the main result. To this aim, we impose that the strong short segments are paid by the dual contributions defined on their internal vertices and edges. This is with ratio at least $\frac{3}{4}$, since for a 2-segment the total dual contribution is $\frac{3}{2}$, and for a 3-segment the total dual contribution is $\frac{3}{2} + \frac{3}{2} - \frac{1}{2} = \frac{5}{2}$. The cost of the strong long segments are paid by the dual contributions defined on their internal vertices, which is with ratio at least $\frac{3}{4}$, since the cost of a long segment is at least 4. There remains the cost of weak segments. The cost of a weak segment is countered with the dual contributions of its internal vertices and a new set of contributions y we will later define. Thus for a weak ℓ-segment, the total contribution is $\ell - 1 + y$. We impose that $\ell - 1$ of this optimally pays for the $\ell - 1$ edges of the segment, and show $y \geq 3/4$. Let k be defined as follows: It is the number of weak segments on F_1, if F is a 2-VCSS. It is the number of weak segments on F_1, excluding the trivial segments defined by the redundant edges, if F is a 2-ECSS. We construct a total value of at least $\frac{3k}{4}$, where k is the total number of weak segments.

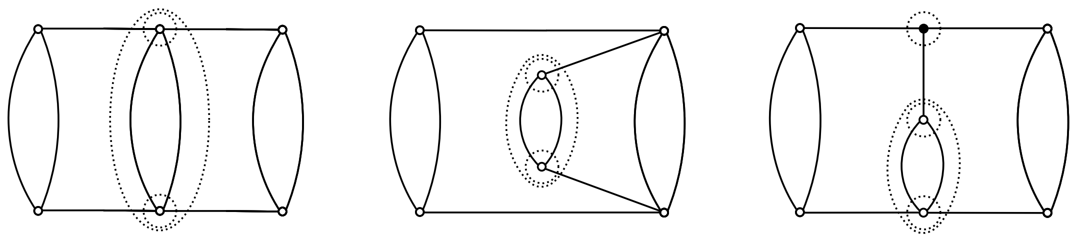
We first show this for a 2-ECSS by an induction on k. We thus consider the base case $k = 2$ in which the two weak segments do not share a common end vertex. This is depicted in Figure 11a. Note that this is indeed the base case, since weak segments do not share a common end vertex due to the first step of the algorithm, which computes a 2-VCSS, and there cannot be a single weak segment on a 2-ECSS. In this case assign $y_{\{v\}} = 1/4$ for each end vertex v of a weak segment, so that the total dual contribution defined for the weak segments is 2, which satisfies the result. In the inductive step one may introduce one, two, three, or four new weak segments by extending the graph in the induction hypothesis. These are given in Figure 11 and Figure 12, where we depict the extending subgraphs in their simplest form. We assume a dual assignment on the internal vertices of the segments in the extending subgraph as described in the first paragraph, so that all its segments are covered with ratio at least $\frac{3}{4}$. We then describe a further dual assignment to cover the newly introduced weak segments. In what follows, we mostly tacitly refer to the first claim of Lemma 3, which forbid the existence of edges that would violate the feasibility of the dual solution.

In Figure 11b and Figure 11c one new weak segment is introduced. Let u be a newly introduced



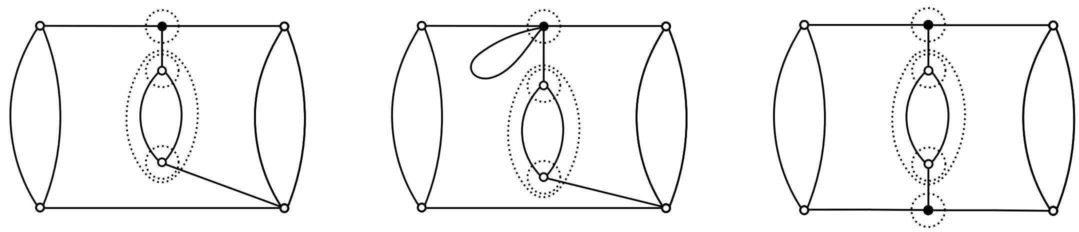
(a) **(b)** **(c)**

Figure 11



(a) (b) (c)

Figure 12



(a) (b) (c)

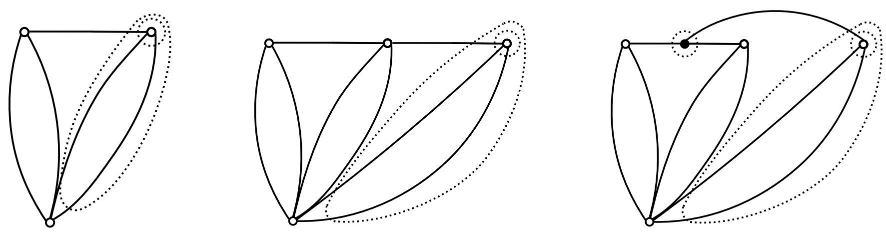
Figure 13

high-degree vertex. If there is no strong short segment with an end vertex u, we define $y_{\{u\}} = 1/2$. Otherwise, we define $y_{\{u\}} = 1/4$ and $y_S = 1/4$, where S is the set of vertices of strong short segments with an end vertex u. This is feasible by the third claim of Lemma 3, a fact that will also be used in the following cases. The duals are shown in the figures with dotted ovals. Note that the closed segment in Figure 11b cannot be a short segment by Lemma 2. In either case the dual contribution is 1, which optimally covers the new weak segment.

In Figure 12a two new weak segments are introduced. Let u and v be two newly introduced high-degree vertices, which are also the end vertices of the weak segments. If u and v are the end vertices of strong short segments, assign $y_{\{u\}} = y_{\{v\}} = 1/4$, and $y_S = 1/4$, where S is the set of vertices of strong short segments with an end vertex u. Otherwise, let $y_{\{u\}} = 1/2$. The total dual contribution by u and v together with y_S is at least $3/2$ in both cases, covering the two new weak segments with ratio at least $\frac{3}{4}$. We do not depict the generalization of Figure 12a, analogous to the one from Figure 11b to Figure 11c, which does not change the analysis.

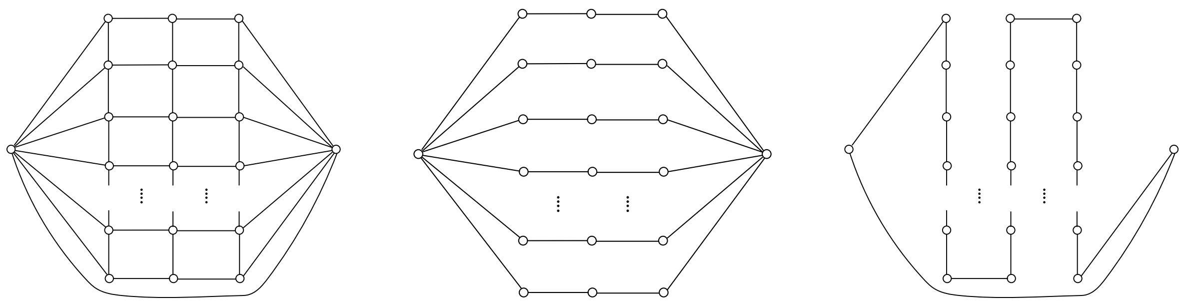
The argument for the configuration in Figure 12b is identical to that of Figure 12a. In Figure 12c, Figure 13a, and Figure 13b three new weak segments are introduced. If an end vertex is only incident to weak segments, it is assigned the dual value $1/2$ and hence contributes 1 (See the vertex depicted as a black dot in Figure 12c and Figure 13a). Otherwise, we define dual values around it as described for Figure 11b, which again contributes 1. This is depicted in Figure 13b. In both cases the other vertices contribute $3/2$, as described for Figure 12a, thus summing up to at least $5/2$. This covers the three newly introduced weak segments with ratio at least $\frac{5}{6}$. Figure 13c is a straightforward generalization of Figure 13a. The total new dual contribution is thus at least $\frac{7}{2}$, which covers the four new weak segments with ratio at least $\frac{7}{8}$. We do not depict the generalizations of Figure 13c, analogous to the one from Figure 13a to Figure 13b, which does not change the analysis. This completes the induction.

We now show the same result for a 2-VCSS. By definition, both the base case and the inductive step for a 2-ECSS are subsets of those for a 2-VCSS except the one depicted in Figure 11b. In addition, a 2-VCSS might have one trivial segment in the base case as depicted in Figure 14a. Let



(a) (b) (c)

Figure 14



(a) (b) (c)

Figure 15: A tight example for the algorithm: (a) Input graph; (b) A solution returned by the algorithm; (c) An optimal solution.

$e \in F_1$ be its edge, and fix an end vertex u of the segment. By the second constraint in (VC), there exists a high-degree vertex v and $W = (W_t, W_h)$ such that $V = \{v\} \cup W_t \cup W_h$, $u \in W_t$, and e covers W. If u is not shared by any strong short segment, define $y_{\{u\}} = 1/2$, which optimally covers e. Otherwise, define $y_{\{u\}} = 1/4$ and $y_S = 1/4$, where S is the set of vertices of strong short segments with an end vertex u, but excluding the vertex v. Note that this is feasible by the third claim of Lemma 3. These duals are shown as dotted ovals in the figure. The total value contributed by these duals is again 1, which optimally covers e.

The inductive step might introduce one or two new weak segments as depicted in Figure 14b and Figure 14c. In Figure 14b we define a dual value of $1/4$ on the newly introduced end vertex of the weak segment. We also define the value $1/4$ for the dual, which is defined as in the base case. These duals in total contribute 1, and hence optimally cover the new trivial segment. Figure 14c additionally contains a vertex incident to only weak segments on which we define a dual of value $1/2$, so that the two new weak segments are optimally covered. This completes the proof. □

Theorem 8.

$$|F| \leq \frac{4}{3} opt(G).$$

Proof. Follows from Lemma 7 and the fourth claim of Lemma 3. □

5 A Tight Example

A tight example for the algorithm is given in Figure 15. The solution returned by the algorithm has cost $4k$. The optimal solution has cost $3k + 2$.

Acknowledgment

We would like to thank Berkay Tahsin Tunca for helping us by drawing the figures. This work was supported by Scientific and Technological Research Council of Turkey (TUBITAK) under the Grant Number 123E530. The author thanks TUBITAK for their supports.

References

[1] M. Bosch-Calvo, F. Grandoni, and A. J. Ameli. A 4/3 approximation for 2-vertex-connectivity. In *50th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 29:1–29:13, 2023.

[2] J. Cheriyan, A. Sebö, and Z. Szigeti. Improving on the 1.5-approximation of a smallest 2-edge connected spanning subgraph. *SIAM J. Discrete Math.*, 14(2):170–180, 2001.

[3] J. Cheriyan and R. Thurimella. Approximating minimum-size k-connected spanning subgraphs via matching. *SIAM J. Comput.*, 30(2):528–560, 2000.

[4] J. Cheriyan, S. Vempala, and A. Vetta. An approximation algorithm for the minimum-cost k-vertex connected subgraph. *SIAM J. Comput.*, 32(4):1050–1055, 2003.

[5] A. Çivril. Corrigendum to "A new approximation algorithm for the minimum 2-edge-connected spanning subgraph problem" [Theor. comput. sci. 943 (2023) 121-130]. *Theor. Comput. Sci.*, 963:113926, 2023.

[6] A. Çivril. A new approximation algorithm for the minimum 2-edge-connected spanning subgraph problem. *Theor. Comput. Sci.*, 943:121–130, 2023.

[7] B. Csaba, M. Karpinski, and P. Krysta. Approximability of dense and sparse instances of minimum 2-connectivity, TSP and path problems. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 74–83, 2002.

[8] A. Czumaj and A. Lingas. On approximability of the minimum-cost k-connected spanning subgraph problem. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 281–290, 1999.

[9] M. Garg, F. Grandoni, and A. J. Ameli. Improved approximation for two-edge-connectivity. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2368–2410, 2023.

[10] N. Garg, S. Vempala, and A. Singla. Improved approximation algorithms for biconnected subgraphs via better lower bounding techniques. In *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms (SODA)*, pages 103–111, 1993.

[11] P. Gubbala and B. Raghavachari. Approximation algorithms for the minimum cardinality two-connected spanning subgraph problem. In *Integer Programming and Combinatorial Optimization, 11th International IPCO Conference*, pages 422–436, 2005.

[12] K. Heeger and J. Vygen. Two-connected spanning subgraphs with at most $\frac{10}{7}$OPT edges. *SIAM J. Discrete Math.*, 31(3):1820–1835, 2017.

[13] C. Hunkenschröder, S. Vempala, and A. Vetta. A 4/3-approximation algorithm for the minimum 2-edge connected subgraph problem. *ACM Trans. Algorithms*, 15(4):55:1–55:28, 2019.

[14] R. Jothi, B. Raghavachari, and S. Varadarajan. A 5/4-approximation algorithm for minimum 2-edge-connectivity. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 725–734, 2003.

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

[16] P. Krysta and V. S. A. Kumar. Approximation algorithms for minimum size 2-connectivity problems. In *18th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 431–442, 2001.

[17] P.Gubbala. *Problems in Graph Connectivity*. PhD thesis, Department of Computer Science, University of Texas at Dallas, Richardson, TX, 2006.

[18] A. Sebö and J. Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014.

[19] S. Vempala and A. Vetta. Factor 4/3 approximations for minimum 2-connected subgraphs. In *Approximation Algorithms for Combinatorial Optimization, Third International Workshop (APPROX)*, pages 262–273, 2000.