Approximation algorithms for connectivity augmentation problems

Zeev Nutov
The Open University of Israel, nutov@openu.ac.il

Abstract

In Connectivity Augmentation problems we are given a graph $H = (V, E_H)$ and an edge set $E$ on $V$, and seek a min-size edge set $J \subseteq E$ such that $H \cup J$ has larger edge/node connectivity than $H$.

In the Edge-Connectivity Augmentation problem we need to increase the edge-connectivity by 1.

In the Block-Tree Augmentation problem $H$ is connected and $H \cup S$ should be $2$-connected.

In Leaf-to-Leaf Connectivity Augmentation problems every edge in $E$ connects minimal deficient sets. For this version we give a simple combinatorial approximation algorithm with ratio $5/3$, improving the $1.91$ approximation of [3] (see also [16]), that applies for the general case. We also show by a simple proof that if the Steiner Tree problem admits approximation ratio $\alpha$ then the general version admits approximation ratio $1 + \ln(4 - x) + \epsilon$, where $x$ is the solution to the equation $1 + \ln(4 - x) = \alpha + (\alpha - 1)x$. For the currently best value of $\alpha = \ln 4 + \epsilon$ [4] this gives ratio $1.942$. This is slightly worse than the ratio $1.91$ of [3], but has the advantage of using Steiner Tree approximation as a “black box”, giving ratio $< 1.9$ if ratio $\alpha \leq 1.35$ can be achieved.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases connectivity augmentation, approximation algorithm

Digital Object Identifier 10.4230/LIPIcs..2020.

1 Introduction

A graph is $k$-connected if it contains $k$ internally disjoint paths between every pair of nodes; if the paths are only required to be edge disjoint then the graph is $k$-edge-connected. In Connectivity Augmentation problems we are given an “initial” graph $G_0 = (V, E_0)$ and an edge set $E$ on $V$, and seek a min-size edge set $J \subseteq E$ such that $G_0 \cup J = (V, E_0 \cup J)$ has larger edge/node connectivity than $G_0$.

- In the Edge-Connectivity Augmentation problem we seek to increase the edge connectivity by one, so $G_0$ is $k$-edge-connected and $G_0 \cup J$ should be $(k + 1)$-edge connected.
- In the 2-Connectivity Augmentation problem we seek to make a connected graph 2-connected, so $G_0$ is connected and $G_0 \cup J$ should be 2-connected.

A cactus is a “tree-of-cycles”, namely, a 2-edge-connected graph in which every block is a cycle (equivalently - every edge belongs to exactly one simple cycle). By [5], the Edge-Connectivity Augmentation problem is equivalent to the following problem:

| CACTUS AUGMENTATION |
|----------------------|
| **Input:** A cactus $T = (V, E_T)$ and an edge set $E$ on $V$. |
| **Output:** A min-size edge set $J \subseteq E$ such that $T \cup J$ is 3-edge-connected. |

It is also known (c.f. [11]) that the 2-Connectivity Augmentation problem is equivalent to the following problem:

| BLOCK-TREE AUGMENTATION |
|--------------------------|
| **Input:** A tree $T = (V, E_T)$ and an edge set $E$ on $V$. |
| **Output:** A min-size edge set $F \subseteq E$ such that $T \cup F$ is 2-connected. |
A more general problem than Cactus Augmentation is as follows. Two sets \( A, B \) cross if \( A \cap B \neq \emptyset \) and \( A \cup B \neq V \). A set family \( \mathcal{F} \) on a groundset \( V \) is a crossing family if \( A \cap B, A \cup B \in \mathcal{F} \) whenever \( A, B \in \mathcal{F} \) cross; \( \mathcal{F} \) is a symmetric family if \( V \setminus A \in \mathcal{F} \) whenever \( A \in \mathcal{F} \). The 2-edge-cuts of a cactus form a symmetric crossing family, with the additional property that whenever \( A, B \in \mathcal{F} \) cross and \( A \cup B, B \setminus A \) are both non-empty, the set \((A \setminus B) \cup (B \setminus A)\) is not in \( \mathcal{F} \); such a symmetric crossing family is called proper \cite{6}. Dinitz, Karzanov, and Lomonosov \cite{5} showed that the family of minimum edge cuts of a graph \( G \) can be represented by 2-edge cuts of a cactus. Furthermore, when the edge-connectivity of \( G \) is odd, the min-cuts form a laminar family and thus can be represented by a tree. Dinitz and Nutov \cite{6} Theorem 4.2] (see also \cite{14} Theorem 2.7]) extended this by showing that an arbitrary symmetric crossing family \( \mathcal{F} \) can be represented by 2-edge cuts and specified 1-node cuts of a cactus; when \( \mathcal{F} \) is a proper crossing family this reduces to the cactus representation of \cite{5}. We say that an edge \( f \) covers a set \( A \) if \( f \) has exactly one end in \( A \). The following problem combines the difficulties of the Cactus Augmentation and the Block-Tree Augmentation problems, see \cite{16}.

\begin{tabular}{|l|}
\hline
**Crossing Family Augmentation**
\hline
**Input:** A graph \( G = (V, E) \) and a symmetric crossing family \( \mathcal{F} \) on \( V \).
\hline
**Output:** A min-size edge set \( J \subseteq E \) that covers \( \mathcal{F} \).
\hline
\end{tabular}

In this problem, the family \( \mathcal{F} \) may not be given explicitly, but we require that certain queries related to \( \mathcal{F} \) can be answered in polynomial time, see \cite{16}. Block-Tree Augmentation and Crossing Family Augmentation admit ratio 2 \cite{13} \cite{8}, that applies also for the min-cost versions of the problems.

The inclusion minimal members of a set family \( \mathcal{F} \) are called leaves. In the Leaf-to-Leaf Crossing Family Augmentation problem, every edge in \( E \) connects two leaves of \( \mathcal{F} \). In the Leaf-to-Leaf Block-Tree Augmentation problem, every edge in \( E \) connects two leaves of the input tree \( T \).

**Theorem 1.** The leaf-to leaf versions of Crossing Family Augmentation and Block-Tree Augmentation admit ratio 5/3.

Better ratios are known for two special cases. In the Tree Augmentation problem the family \( \mathcal{F} \) is laminar, namely, any two sets in \( \mathcal{F} \) are disjoint or one contains the other; this problem can be also defined in connectivity terms - make a spanning tree 2-edge-connected by adding a min-size edge set \( J \subseteq E \). This problem was vastly studied; see \cite{1} \cite{10} \cite{12} \cite{2} \cite{15} and the references therein for additional literature on the Tree Augmentation problem.

In the Leaf-to-Leaf Tree Augmentation problem, every edge in \( E \) connects two leaves of the tree; this problem admits ratio 17/12 \cite{13}. The Cycle Augmentation problem is a particular case of the Cactus Augmentation problem when the cactus is a cycle; in this case the leaves are the singleton nodes. The Cycle Augmentation problem admits ratio \( \frac{3}{2} + \epsilon \) \cite{9}; our algorithm from Theorem \cite{1} uses some ideas from \cite{9}.

Byrka, Grandoni, and Ameli \cite{3} showed that Cactus Augmentation admits ratio \( 2 \ln 4 - \frac{967}{1720} + \epsilon < 191 \), breaching the natural 2 approximation barrier. This was extended to Crossing Family Augmentation and Block Tree Augmentation in \cite{10}.

In the Steiner Tree problem we are given a graph \( G = (V, E) \) with edge costs and a set \( R \subseteq V \) of terminals, and seek a min-cost subtree of \( G \) that spans \( R \). We prove the following.

**Theorem 2.** If Steiner Tree admits ratio \( \alpha \) then Crossing Family Augmentation and Block-Tree Augmentation admit ratio \( 1 + \ln(4 - x) + \epsilon \), where \( x \) is the solution to the equation \( 1 + \ln(4 - x) = \alpha + (\alpha - 1)x \).
Currently, $\alpha = \ln 4 + \epsilon [2];$ in this case we have ratio 1.942 for the problems in the theorem. This is slightly worse than the ratio 1.91 of [3] (see also [10]), but our algorithm is very simple and has the advantage of using Steiner Tree approximation as a “black box”. E.g., if ratio $\alpha = 1.35$ can be achieved, then we immediately get ratio 1.895 $< 1.9$.

2 The leaf-to-leaf case (Theorem 1)

We prove Theorem 1 for the Crossing Family Augmentation problem, and later indicate the changes needed to adopt the proof for the Block-Tree Augmentation problem. We need some definition to describe the algorithm. Let $\mathcal{F}$ be a set family on $V$. We say that $A \in \mathcal{F}$ separates $u, v \in V$ if $|A \cap \{u, v\}| = 1$; $u, v$ are $\mathcal{F}$-separable if such $A$ exists and $u, v$ are $\mathcal{F}$-inseparable otherwise. Similarly, $A$ separates edges $f, g$ if one of $f, g$ has both ends in $A$ and the other has no end in $A$; $f, g$ are $\mathcal{F}$-separable if such $A \in \mathcal{F}$ exists, and $\mathcal{F}$-inseparable otherwise. The relation $\{(u, v) \in V \times V : u, v \text{ are } \mathcal{F}\text{-inseparable}\}$ is an equivalence, and we call its equivalence classes $\mathcal{F}$-classes. W.l.o.g. we will assume that all $\mathcal{F}$-classes are singletons and that no edge in $E$ has both ends in the same class; in particular, the leaves of $\mathcal{F}$ are singletons, and we denote the leaf set of $\mathcal{F}$ by $L$. We will also often abbreviate the notation for singleton sets and write $v, e$ instead of $\{v\}, \{e\}$. Given $J \subseteq E$, the residual instance $((V^J, E^J), \mathcal{F}^J)$ is defined as follows.

- The residual family $\mathcal{F}^J$ of $\mathcal{F}$ w.r.t. $J$ consists of all members of $\mathcal{F}$ that are uncovered by the edges in $J$. It is known that $\mathcal{F}^J$ is crossing (and symmetric) if $\mathcal{F}$ is.

- $V^J$ is the set of $\mathcal{F}^J$-classes (w.l.o.g. each of them can be shrunk into a single element).

- $E^J$ is obtained from $E \setminus J$ by removing all edges that have both ends in the same $\mathcal{F}^J$-class.

In addition, given a set $R \subseteq V$ of terminals, the residual set of terminals $R^J$ is the set of $\mathcal{F}^J$-classes that contain some member of $R$. For illustration see Fig. 1(a,b,c).

For any edge $e = uv$, there is an $\mathcal{F}^e$-class that contains both $u$ and $v$; denote this class by $C(\mathcal{F}, e)$. Given a set $R$ of terminals (a subset of $\mathcal{F}$-classes), the $(R, E, \mathcal{F})$-incidence graph $H = (U, E_H)$ has node set $U = E \cup R$ and edge set

$$E_H = \{ee' : e, e' \in E \text{ are } \mathcal{F}\text{-inseparable}\} \cup \{er : r \in R, e \in E, r \in C(\mathcal{F}, e)\}.$$ 

Let $R \subseteq V$ and let $H$ be the $(R, E, \mathcal{F})$-incidence graph. Note that $R$ is an independent set in $H$. It was shown in [10] that for $R = L$ being the set of leaves of $\mathcal{F}$, an edge set $J \subseteq E$ is a feasible solution to CROSSING FAMILY AUGMENTATION if and only if the subgraph $H[J \cup R]$ of $H$ induced by $J \cup R$ is connected. The proof in [10] extends to any $R \subseteq V$ that contains $L$. This implies that CROSSING FAMILY AUGMENTATION admits an approximation ratio preserving reduction to the following problem (see [16], [2] for more details).

**Subset Steiner Connected Dominating Set (SS-CDS)**

**Input:** A graph $H = (U, E_H)$ and a set $R \subseteq U$ of independent terminals.

**Output:** A min-size node set $S \subseteq U \setminus R$ such that $H[S]$ is connected and $S$ dominates $R$.

Given a SS-CDS instance and $s \in S = U \setminus R$ let $R(s) = R_H(s)$ denote the set of neighbors of $s$ in $H$ that belong to $R$. Let opt be the optimal solution value of a problem instance at hand. Before describing the algorithm, we will prove the following lemma.

**Lemma 3.** Let $I = (H, R)$ be a SS-CDS instance such that $\|R(s)\| = 2$ for all $s \in S = U \setminus R$. Then one of the following holds:

(i) There are adjacent $a, b \in S$ with $R(a) \cap R(b) = \emptyset$.

(ii) $\text{opt} \geq |R| - 1$. 
Figure 1 Illustration of definitions for a Crossing Family Augmentation instance where $\mathcal{F}$ is represented by a cactus. Here $A \in \mathcal{F}$ if and only if $A$ is a connected component obtained by removing a pair of edges that belong to the same cycle of the cactus. The edges in $E$ are shown by dashed arcs and the terminals in $R$ are shown by gray circles. The cactus of the residual family w.r.t. a single edge is obtained by “squeezing” the cycles along the path of cycles between the ends of the edge. (a) The original instance. (b) The residual instance w.r.t. $e$. (c) The residual instance w.r.t. $f$. (d) The $(R, E, \mathcal{F})$-incidence graph of the instance in (a).

Figure 2 Illustration to the proof of Lemma 3.
Proof. Assume that (i) does not hold for $\mathcal{I}$; we will prove that then (ii) holds. The proof is by induction on $|\mathcal{R}|$. In the base case $|\mathcal{R}| = 2$ (ii) holds. Assume that the statement is true for $|\mathcal{R}| - 1 \geq 2$. Let $T$ be an optimal solution tree and $S$ the set of non-terminals in $T$. Root $T$ at some node and let $s \in S$ be a non-terminal farthest from the root. The children of $s$ are terminal leaves, and assume w.l.o.g. that $R(s) = \{u, v\}$ is the set of children of $s$; if $s$ has just one child in $T$, then it has another terminal neighbor in $H$, that can be attached to $s$.

Consider the residual instance $\mathcal{I}' = (G' = (V', E'), R')$ and the tree $T'$ obtained by contracting $R(s)$ into the new terminal $s'$, and deleting any $z \in U \setminus (R + s)$ with $R(z) = R(s)$. Then $|R'| = |R| - 1$, $|R'(z)| = 2$ for all $z \in R'$, $T'$ is an optimal solution for $\mathcal{I}'$, and $S' = S - s$ is the set of non-terminals of $T'$.

If (i) does not hold for the new instance $\mathcal{I}'$ then (ii) holds for $\mathcal{I}'$, by the induction hypothesis. Then $|S| = |S'| + 1 \geq (|R'| - 1) + 1 = |R| - 1$, and we get that (ii) holds for $\mathcal{I}$. Assume henceforth that (i) holds for $\mathcal{I}'. We obtain a contradiction by showing that then (i) holds for $\mathcal{I}$. Let $a, b \in V' \setminus R'$ be such that $R'(a) \cap R'(b) = \emptyset$, see Fig. 2. If $s' \notin R'(a) \cup R'(b)$ then clearly (ii) holds for $\mathcal{I}$. Otherwise, if say $s' \in R'(a)$, then we have two cases. If one of $u, v$, say $v$, is a neighbor of $a$ in $G$ (see Fig. 2(a)) then $R(a) \cap R(b) = \emptyset$. Otherwise (see Fig. 2(a)), $R(a) \cap R(b) = \emptyset$. In both cases, we obtain a contradiction to the assumption that (i) does not hold for $\mathcal{I}$. □

We also need the following known lemma.

Lemma 4. Any inclusion minimal cover $J$ of a set family $\mathcal{F}$ is a forest.

Proof. Suppose to the contrary that $J$ contains a cycle $C$. Since $P = C \setminus \{e\}$ is a uv-path, then for any $A$ covered by $e$, there is $e' \in P$ that covers $A$. This implies that $J \setminus \{e\}$ also covers $\mathcal{F}$, contradicting the minimality of $J$. □

The algorithm starts with a partial solution $J = \emptyset$ and has two phases. Phase 1 consists of iterations. At the beginning of each iteration, construct the $(E, R^J, F^J)$-incidence graph $H^J$, where initially $R$ is the set of leaves of $\mathcal{F}$. Then, do one of the following:

1. If $H^J$ has a node $e \in E$ with $|R^J(e)| \geq 3$, then add $e$ to $J$.
2. Else, if there are $e, f \in E$ with $R^J(e) \cap R^J(f) = \emptyset$, then add both $e, f$ to $J$.

If none of the above two cases occurs, then we apply Phase 2, in which we add to $J$ an inclusion minimal cover of $\mathcal{F}^J$; note that all edges in $E^J$ have both endnodes in $R^J$. A more formal description is given in Algorithm 1.

We show that the algorithm achieves ratio $5/3$. Note that:

- Adding an edge $e$ as in step 4 reduces the number of terminals by at least 2.
- Adding an edge pair $e, f$ as in step 5 reduces the number of terminals by at least 3.

Algorithm 1: $(G = (V, E), F, R)$

1. $J \leftarrow \emptyset$
2. repeat
3. let $H^J$ be the $(E^J, R^J, F^J)$-incidence graph
4. if $H^J$ has a node $e \in E$ with $|R^J(e)| \geq 3$ then $J \leftarrow J \cup \{e\}$
5. else if $H^J$ has node pair $e, f \in E$ with $R^J(e) \cap R^J(f) = \emptyset$ then $J \leftarrow J \cup \{e, f\}$
6. until no edge $e$ or an edge pair $e, f$ as above exists;
7. find an inclusion minimal $F^J$-cover and add it to $J$
8. return $J$
Thus the approximation ratio is bounded by \( \ell \). Then \( H \) if the paths \( J \) coincides with the proof given for 2. On the other hand, \( \psi \) is a tree. The block-tree mapping \( \psi : V \rightarrow C_G \cup B_G \) of \( G \) is defined by \( \psi(v) = v \in C_G \) and \( \psi(v) \) is the block that contains \( v \) if \( v \in V \setminus C_G \). Given a Block-Tree Augmentation instance \( (T = (V, E_T), E) \) and \( J \subseteq E \), the residual instance \( (T^J = (V^J, E^J_T), E^J) \) is defined as follows.

- \( T^J \) is the block tree of \( T \cup J \).
- \( E^J = \{ \psi(u)\psi(v) : uv \in E \setminus J, \psi(u) \neq \psi(v) \} \), where \( \psi \) is the the block-tree mapping of \( T \cup J \).

For a set \( R \subseteq V \) of terminals, the residual set of terminals is \( R^J = \psi(R) = \cup_{r \in R} \psi(r) \). For an edge \( e = uv \) let \( T_e \) denote the unique \( uv \)-path in \( T \). We say that \( e, f \in E \) are T-inseparable if the paths \( T_e, T_f \) have an edge in common. The \((R, E, T)\)-incidence graph \( H = (U, E_H) \) has node set \( U = E \cup R \) and edge set

\[
E_H = \{ ef : e, f \in E \text{ are T-inseparable} \} \cup \{ er : r \in R, e, r \in T_e \}.
\]

It was shown in [10] that for \( R = L \) being the set of leaves of \( F \), an edge set \( J \subseteq E \) is a feasible solution to Block-Tree Augmentation if and only if the subgraph \( H[J \cup R] \) of \( H \) induced by \( J \cup R \) is connected. The proof in [10] extends to any \( R \subseteq V \) that contains \( L \). This implies that Crossing Family Augmentation admits an approximation ratio preserving reduction to SS-CDS, see [10] for details. Lemma[4] also extends to this case, as it is known that an if \( J \) is an inclusion minimal edge set whose addition makes a connected graph 2-connected, then \( J \) is a forest.

With these definitions and facts, the rest of the proof for the Block-Tree Augmentation coincides with the proof given for Crossing Family Augmentation, concluding the proof of Theorem[4].
3 The general case (Theorem 2)

Recall that each of the problems Crossing Family Augmentation and Block-Tree Augmentation admits an approximation ratio preserving reduction to the SS-CDS problem with \( R = L \) being the set of terminals. The SS-CDS instances that arise from this reduction have the following property; see [3, 10]:

\[(*) \text{ The neighbors of every } r \in R \text{ induce a clique.}\]

In fact, SS-CDS with property (\(*)\) is equivalent to the Node Weighted Steiner Tree problem with property (\(*)\) with unit node weights for non-terminals (the terminals have weight zero). Clearly, any SS-CDS solution is a feasible Node Weighted Steiner Tree solution; for the other direction, note that if property (\(*)\) holds, then the set of non-terminals in any feasible Node Weighted Steiner Tree solution is a feasible SS-CDS solution. The relation to the ordinary Steiner Tree problem is given in following lemma.

Lemma 5 ([3]). Let \( S \) be a SS-CDS solution and \( T = (U, J) \) a Steiner Tree solution on instance \((G, R)\) with unit edge costs. Then:

(i) If (\(*)\) holds then \( T \) can be converted into a SS-CDS solution \( S_J \) with \(|S_J| = |J| - |R| + 1.\)

(ii) \( S \) can be converted into a Steiner Tree solution \( T_S = (U_S, J_S) \) with \(|J_S| = |S| + |R| - 1.\)

Proof. We prove (i). Any Steiner Tree solution \( T' = (U', J') \) can be converted into a solution \( T = (U, J) \) such that \(|J| = |J'|\) and \( R \) is the leaf set of \( T' \). For this, for each \( r \in R \) that is not a leaf of \( T' \), among the edges incident to \( r \) in \( T' \), choose one and replace the other edges by a tree on the neighbors of \( r \); this is possible by (\(*)\). The non-leaf nodes of such \( T \) form a a SS-CDS as required. For (ii), taking a tree on \( S \) and for each \( r \in R \) adding an edge from \( r \) to \( S \) gives a Steiner Tree solution as required.

Let \( J^* \) be an optimal and \( J \) an \( \alpha \)-approximate Steiner Tree solutions. Let \( S_J, S^* \) be SS-CDS solutions, where \( S_J \) is derived from \( J \) and \( S^* \) is an optimal one. Then

\[|S_J| + R - 1 = |J| \leq \alpha|J'| \leq \alpha|J_S| = \alpha(|S^*| - 1 + |R|) = \alpha|S^*| + \alpha(|R| - 1).\]

This implies that if Steiner Tree admits ratio \( \alpha \) then SS-CDS with property (\(\alpha\)) admits a polynomial time algorithm that computes a solution \( S \) of size \(|S| \leq \alpha^{\text{opt}} + (\alpha - 1)|L| \) and achieves ratio \( \alpha + (\alpha - 1) \frac{|L|}{\alpha^{\text{opt}}} = \alpha + (\alpha - 1)x \), where \( x = \frac{|L|}{\alpha^{\text{opt}}} \), \( 0 < x \leq 2 \). We will prove the following.

Theorem 6. Crossing Family Augmentation and Block-Tree Augmentation admit ratio \( 1 + \ln \left(4 - \frac{|L|}{\alpha^{\text{opt}}} \right) + \epsilon.\)

From Lemma 5 and Theorem 6 it follows that we can achieve ratio

\[\max \{ \alpha + (\alpha - 1)x, 1 + \ln(4 - x) \} + \epsilon \text{ where } x = \frac{|L|}{\alpha^{\text{opt}}}\]

The worse case is when these two ratios are equal, which gives the Theorem 2 ratio. In the case \( \alpha = \ln 4 + \epsilon \) we have \( x \approx 1.4367 \), so \( L \approx 1.4367^{\text{opt}} \) and \( \text{opt} \approx 0.69L \). The ratio in this case is \( 1 + \ln(4 - x) + \epsilon < 1.942 \).
A set function $f$ is **increasing** if $f(A) \leq f(B)$ whenever $A \subseteq B$; $f$ is **decreasing** if $-f$ is increasing, and $f$ is **sub-additive** if $f(A \cup B) \leq f(A) + f(B)$ for any subsets $A, B$ of the ground-set. Let us consider the following algorithmic problem:

**MIN-COVERING**

**Input:** Non-negative set functions $\nu, \tau$ on subsets of a ground-set $U$ such that $\nu$ is decreasing, $\tau$ is sub-additive, and $\tau(\emptyset) = 0$.

**Output:** $A \subseteq U$ such that $\nu(A) + \tau(A)$ is minimal.

We call $\nu$ the **potential** and $\tau$ the **payment**. The idea behind this interpretation and the subsequent greedy algorithm is as follows. Given an optimization problem, the potential $\nu(A)$ is the (bound on the) value of some “simple” augmenting feasible solution for $A$. We start with an empty set solution, and iteratively try to decrease the potential by adding a set $B \subseteq U \setminus A$ of minimum “density” – the price paid for a unit of the potential. The algorithm terminates when the price $\geq 1$, since then we gain nothing from adding $B$ to $A$. The ratio of such an algorithm is bounded by $1 + \ln \nu(\emptyset)$ (assuming that during each iteration a minimum density set can be found in polynomial time). So essentially, the greedy algorithm converts ratio $\alpha = \frac{\nu(\emptyset)}{\nu(A)}$ into ratio $1 + \ln \alpha$.

Fix an optimal solution $A^*$. Let $\nu^* = \nu(A^*)$, $\tau^* = \tau(A^*)$, so $\text{opt} = \tau^* + \nu^*$. The quantity $\frac{\tau^*}{\nu(A)}$ is called the **density** of $B$ (w.r.t. $A$); this is the price paid by $B$ for a unit of potential. The **Greedy Algorithm** (a.k.a. **Relative Greedy Heuristic**) for the problem starts with $A = \emptyset$ and while $\nu(A) > \nu^*$ repeatedly adds to $A$ a non-empty augmenting set $B \subseteq U$ that satisfies the following condition, while such $B$ exists:

**Density Condition:**

$$\frac{\tau(B)}{\nu(A) - \nu(A \cup B)} \leq \min \left\{ 1, \frac{\tau^*}{\nu(A) - \nu^*} \right\}.$$

Note that since $\nu$ is decreasing, $\nu(A) - \nu(A \cup A^*) \geq \nu(A) - \nu(A^*) = \nu(A) - \nu^*$; hence if $\nu(A) > \nu^*$, then $\frac{\tau^*}{\nu(A) - \nu(A^*)} \leq \frac{\tau^*}{\nu(A) - \nu^*}$ and there exists an augmenting set $B$ that satisfies the condition $\frac{\tau(B)}{\nu(A) - \nu(A \cup B)} \leq \frac{\tau^*}{\nu(A) - \nu^*}$, e.g., $B = A^*$. Thus if $B^*$ is a minimum density set and $\frac{\tau(B^*)}{\nu(A) - \nu(A \cup B^*)} \leq 1$, then $B^*$ satisfies the Density Condition; otherwise, the density of $B^*$ is larger than 1 so no set can satisfy the Density Condition. The following statement is known, c.f. an explicit proof in [17].

**Theorem 7.** The **Greedy Algorithm** achieves approximation ratio $1 + \frac{\tau^*}{\text{opt}} \ln \frac{\nu(\emptyset) - \nu^*}{\tau^*}$.

This applies also in the case when we can only compute a $\rho$-approximate minimum density augmenting set, while invoking an additional factor $\rho$ in the ratio.

To use the framework of Theorem 7 we need to define $\tau$ and $\nu$. Let $J \subseteq E$ be an edge set. The payment $\tau(J) = |J|$ is just the size of $J$. The potential of $J$ is defined by $\nu(J) = |R^J| - 1$, where $R$ is a set of terminals such that $L \subseteq R \subseteq V$, defined in the following lemma. For an edge set $F$ let $F_{LL}$ be the set of edges in $F$ with both ends in $L$, and $F_L$ the set of edges in $F$ that have exactly one end in $L$.

**Lemma 8.** Let $F$ be an optimal solution to **Crossing Family Augmentation** instance and let $c$ be a cost function on $E$ defined by $c(e) = 0$ if $e \in E_{LL}$, $c(e) = 1$ if $e \in E_L$, and $c(e) = 2$ otherwise. Let $J$ be a 2-approximate $c$-costs solution and let $R$ be the set of ends of the edges in $J$. Then $|R| \leq c(J) + L \leq 4|F| - |L| = 4\text{opt} - |L|$.
Proof. Clearly, $|R| \leq c(J) + |L|$. We show that $c(J) \leq 4|F| - 2|L|$. Let $F'$ be the set of edges in $F$ that have no end in $L$. Since $|F'| = |F| - |F_L| - |F_{LL}|$ and $2|F_{LL}| + |F_L| \geq L$

$$c(F) = |F_L| + 2|F'| = |F_L| + 2(|F| - |F_L| - |F_{LL}|) = 2|F| - ((|F_L| + 2|F_{LL}|) \leq 2|F| - |L|.$$ 

Since $c(J) \leq 2c(F)$, the lemma follows.

It is easy to see that $\nu$ is decreasing and $\tau$ is subadditive. The next lemma shows that the obtained $\text{MIN-COVERING}$ instance is equivalent to the $\text{CROSSING FAMILY AUGMENTATION}$ instance, and that we may assume that $\tau^* = \text{opt}$ and $\nu^* = 0$.

Lemma 9. If $J$ is a feasible solution to $\text{CROSSING FAMILY AUGMENTATION}$ then $\nu(J) = 0$. If $J$ is a feasible $\text{MIN-COVERING}$ solution then one can construct in polynomial time a feasible $\text{CROSSING FAMILY AUGMENTATION}$ solution of size $\leq \tau(J) + \nu(J)$. In particular, both problems have the same optimal value, and $\text{MIN-COVERING}$ has an optimal solution $J^*$ such that $\nu(J^*) = 0$ and $\tau(J^*) = \text{opt}$.

Proof. If $J$ is a feasible $\text{CROSSING FAMILY AUGMENTATION}$ solution then $|R^I| = 1$ and thus $\nu(J) = 0$. Let $I$ be a $\text{MIN-COVERING}$ solution such that every edge in $I$ has both ends in $R$; e.g., $I$ can be as in Lemma 8. Then $I^J$ is a feasible solution to the residual problem w.r.t. $J$ and every edge in $I^J$ has both ends in $R^J$. Let $I' \subseteq I^J$ be an inclusion minimal edge set such that $J \cup I'$ is a feasible solution. By Lemma 8, $I'$ is a forest, hence $|I| \leq |R^J| - 1$. Consequently, $J \cup I'$ is a feasible solution of size at most $|J| + |I'| \leq |J| + |R^J| - 1 = \tau(J) + \nu(J)$.  

Recall also that $\nu(\emptyset) \leq 4\text{opt} - |L|$, by Lemma 8. We will show how to find for any $\epsilon > 0$, a $(1 + \epsilon)$-approximate best density set in polynomial time. It follows therefore that we can apply the greedy algorithm to produce a solution of value $1 + \epsilon$ times of

$$1 + \frac{\tau^*}{\text{opt}} \ln \frac{\nu(\emptyset) - \nu^*}{\tau^*} = 1 + \ln \frac{4\text{opt} - |L|}{\text{opt}} = 1 + \ln \left(\frac{4 - |L|}{\text{opt}}\right).$$

In what follows note that if $a_1, \ldots, a_q$ and $b_1, \ldots, b_q$ are positive reals, then by an averaging argument there exists an index $1 \leq i \leq q$ such that $a_i/b_i \leq \sum_{j=1}^q a_j/\sum_{i=1}^q b_j$.

Given a $\text{CROSSING FAMILY AUGMENTATION}$ instance, a set $R \supseteq L$ of terminals, and $F \subseteq E$, consider the corresponding $\text{SS-CDS}$ instance $(H = (U, E_H), R)$ and the set of non-terminals $Q$ that corresponds to $F$. The density of $F$ is $\frac{|F|}{|R| - |L|}$, and in the $\text{SS-CDS}$ instance this is computed by taking a maximal forest in the graph induced by $Q$ and the terminals that have a neighbor in $Q$; then the density is $|Q|$ over the number of trees in this forest. So in what follows we may speak of a density of a subforest of $H$. Let $T_i = (S_i \cup R_i, E_i), i = 1, \ldots, q$, be the connected components of such a forest, $(R_i$ is the set of terminals in $T_i$) and let $s_i = |S_i|$ and $r_i = |R_i|$, where $r_i \geq 2$. The density of the forest is $\sum_{i=1}^q s_i/r_i \sum_{i=1}^q (r_i - 1)$ while the density of each $T_i$ is $s_i/(r_i - 1)$. By an averaging argument, some $T_i$ has density not larger than that of the forest. Consequently, we may assume that the minimum density is attained for a tree, say $T$.

Let $T = (S \cup R, E)$ be a tree with leaf set $R$. The density of $T$ is $\frac{r}{s + 1}$, where $r = |R|$ is the number of terminals ($R$-nodes) and $s$ is the number of non-terminals ($S$-nodes) in $T$. The usual approach is to show that for any $k$ there exists a subtree $T'$ of $T$ with $k$ terminals (or $k$ non-terminals) such that the density of $T'$ is at most $1 + f(k)$ times the density of $T$, where $\lim_{k \to \infty} f(k) = 0$. The decomposition lemma that we prove is not a standard one. The difficulty can be demonstrated by the following examples. Consider the case when $T$ is a star with $n$ leaves. Then the density of $T$ is $1/(n - 1)$, while a subtree with $k$ leaves has density $1/(k - 1)$. If $T$ is a path with $n$ non-terminals, then the density of $T$ is $n$, while
a subtree with $k < n$ non-terminals has density $k/0 = \infty$. In both cases, the density of the subtree may be arbitrarily larger than that of $T$. To overcome this difficulty, we will decompose $T$ w.r.t. a certain subset $P$ of the non-terminals. Let $P \subseteq S$. Let $s = |S|$, $r = |R|$, and $p = |P|$. For a subtree $T'$ of $T$ let $S(T')$, $R(T')$, and $P(T')$ denote the set of $S$-nodes, $R$-nodes, and $P$-nodes in $T'$, respectively. We prove the following.

**Lemma 10.** Let $k \geq 2$. If $p \geq 3k + 1$ then there exists subtrees $T_1, \ldots, T_q$ of $T$ such that the following holds.

1. $\sum_{i=1}^{q} |S(T_i)| \leq s + q$.
2. Every $R$-node belongs to exactly one subtree, hence $\sum_{i=1}^{q} |R(T_i)| = r$.
3. $|P(T_i)| \in [k, 3k]$ for all $i$ and $q \leq \frac{p}{k-1}$.

**Proof.** Root $T$ at some node in $S$. For any $v \in S$ chosen as a “local root”, the subtree $T'$ rooted at $v$ is a subtree of $T$ that consist of $v$ and its descendants. Let $T'$ be an inclusion minimal rooted subtree of $T$ such that $|P(T')| \geq k + 1$. Note that $v \in P$. Let $B_1, \ldots, B_m$ be the branches hanging on $v$ and let $p_j = |P(B_j)|$. By the definition of $T_r$, each $p_j$ is in the range $[0, k]$ and $\sum_{j=1}^{m} p_j \geq k$. We claim that $\{p_1, \ldots, p_m\}$ can be partitioned such that the sum of each part plus 1 is in the range $[k, 3k]$. To see this, apply a greedy algorithm for **Multi-Bin Packing** with bins of capacity $2k$; at the end there is at most one bin with sum $\leq k - 1$ (as two such bins can be joined), and joining this bin to any other bin gives a partition as required. Now we remove $T'$ and the $S$-nodes on the path from $v$ to its closest terminal ancestor, and apply the same procedure on the remaining tree. If the last rooted subtree $T''$ considered has $|P(T'')| \leq k - 1$, then this tree can be joined to a subtree $T_i$ with $|P(T_i)| \leq 2k$ derived in previous iteration. Finally, $q \leq \frac{p+2k}{k-1}$ by the construction and since $|P(T_i)| \geq k$ for all $i$; this implies $q \leq \frac{p}{k-1}$.

Now we let $P = P_1 \cup P_2$, where $P_1$ is the set of nodes that have degree at least 3 in $T$ and $P_2$ is the of nodes that have a terminal neighbor in $T$. Note that $|P_1| \leq r$ and $|P_2| \leq r$. Hence $p \leq 2r$, and clearly $p \leq s$. By an averaging argument and Lemma 10 the density of some $T_i$ is bounded by $s_i/(r_i - 1) \leq \sum_{j=1}^{q} s_j / \sum_{j=1}^{q} (r_j - 1) \leq (s + q)/(r - q)$. Thus for $k \geq 3$ we get

$$\frac{s_i}{r_i - 1} \cdot \frac{r - 1}{s} \leq \frac{s + q}{r - q} \cdot \frac{r}{s} = \frac{s + p/(k-1)}{r - p/(k-1)} \cdot \frac{r}{s} = \frac{1 + 1/(k-1)}{1 - 2/(k-1)} = \frac{k}{k-3} = 1 + \frac{3}{k-3}.$$  

This implies that we can find a $(1 + \epsilon)$-approximate min-density tree by searching over all trees $T'$ with $|P(T')| \in [k, 3k]$, where given $\epsilon > 0$ we let $k = [3/\epsilon] + 3$. Specifically, for every $P' \subseteq S$ with $|P'| \in [k, 3k]$, we find an MST $T'$ in the metric completion of the current incidence graph, and then add to $T'$ all the terminals that have a neighbor in $P'$. Among all subtrees we choose one of minimum density. The time complexity is $n^{3k}$ which is polynomial for any fixed $\epsilon > 0$.

The process of adjusting the proof to the **Block-Tree Augmentation** is identical to the one in the proof of Theorem 1. This concludes the proof of Theorem 6 and thus also the proof of Theorem 2 is complete.

---

**References**

1. D. Adjiashvili. Beating approximation factor two for weighted tree augmentation with bounded costs. In *SODA*, pages 2384–2399, 2017.
2. M. Basavaraju, F. V. Fomin, P. A. Golovach, P. Misra, M. S. Ramanujan, and S. Saurabh. Parameterized algorithms to preserve connectivity. In *ICALP*, Part I, pages 800–811, 2014.
J. Byrka, F. Grandoni, and A. J. Ameli. Breaching the 2-approximation barrier for connectivity augmentation: a reduction to steiner tree. In STOC, pages 815–825, 2020. For the full version see. URL: https://arxiv.org/abs/1911.02259.

J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanitá. Steiner tree approximation via iterative randomized rounding. J. ACM, 60(1):6:1–6:33, 2013.

E. A. Dinitz, A. V. Karzanov, and M. V. Lomonosov. On the structure of a family of minimal weighted cuts in a graph. Studies in Discrete Optimization, page 290–306, 1976.

Y. Dinitz and Z. Nutov. A 2-level cactus model for the system of minimum and minimum+1 edge-cuts in a graph and its incremental maintenance. In STOC, pages 509–518, 1995.

S. Fiorini, M. Groß, J. Könemann, and L. Sanitá. A \( \frac{5}{3} \)-approximation algorithm for tree augmentation via chvátal-gomory cuts. In SODA, pages 817–831, 2018.

L. Fleischer, K. Jain, and D. P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. J. Comput. Syst. Sci., 72(5):838–867, 2006.

W. Gálvez, F. Grandoni, A. J. Ameli, and K. Sornat. On the cycle augmentation problem: Hardness and approximation algorithms. In WAOA, pages 138–153, 2019.

F. Grandoni, C. Kalaitzis, and R. Zenklusen. Improved approximation for tree augmentation: saving by rewiring. In STOC, pages 632–645, 2018.

S. Khuller. Approximation algorithms for finding highly connected subgraphs. In D. Hochbaum, editor, Approximation Algorithms for NP-hard problems, chapter 6, pages 236–265. PWS, 1995.

G. Kortsarz and Z. Nutov. A simplified 1.5-approximation algorithm for augmenting edge-connectivity of a graph from 1 to 2. ACM Transactions on Algorithms, 12(2):23, 2016.

Y. Maduel and Z. Nutov. Covering a laminar family by leaf to leaf links. Discrete Applied Mathematics, 158(13):1424–1432, 2010.

Z. Nutov. Structures of Cuts and Cycles in Graphs; Algorithms and Applications. PhD thesis, Technion, Israel Institute of Technology, 1997.

Z. Nutov. On the tree augmentation problem. In ESA, pages 61:1–61:14, 2017. To appear in Algorithmica.

Z. Nutov. 2-node-connectivity network design. CoRR, abs/2002.04048, 2020. To appear in WAOA. URL: https://arxiv.org/abs/2002.04048.

Z. Nutov, G. Kortsarz, and E. Shalom. Approximating activation edge-cover and facility location problems. In MFCS, pages 20:1–20:14, 2019.

R. Ravi and D. P. Williamson. An approximation algorithm for minimum-cost vertex-connectivity problems. Algorithmica, 18(1):21–43, 1997.