

Tight Analysis of the Primal-Dual Method for Edge-Covering Pliable Set Families

Zeev Nutov 

The Open University of Israel, Ra'anana, Israel

Abstract

A classic result of Williamson, Goemans, Mihail, and Vazirani [STOC 1993: 708–717] states that the problem of covering an uncrossable set family by a min-cost edge set admits approximation ratio 2, by a primal-dual algorithm with a reverse delete phase. Bansal, Cheriyan, Grout, and Ibrahimpur [ICALP 2023: 15:1–15:19] showed that this algorithm achieves approximation ratio 16 for a larger class of so called γ -pliable set families, that have much weaker uncrossing properties. The approximation ratio 16 was improved to 10 in [11]. Recently, Bansal [3] obtained approximation ratio 8 for γ -pliable families and also considered an important particular case of the family of cuts of size $< k$ of a graph H . We will improve the approximation ratio to 7 for the former case and give a simple proof of approximation ratio 6 for the latter case. Furthermore, if H is λ -edge-connected then we will show a slightly better approximation ratio $6 - \frac{1}{\beta+1}$, where $\beta = \left\lfloor \frac{k-1}{\lceil (\lambda+1)/2 \rceil} \right\rfloor$. Our analysis is supplemented by examples indicating that these approximation ratios are asymptotically tight for the primal-dual algorithm.

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms

Keywords and phrases primal dual method, pliable set family, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.82

1 Introduction

For an edge set or a graph J on node set V and disjoint node subsets $S, T \subseteq V$ let $\delta_J(S, T)$ denote the set of edges in J between S and T , and let $d_J(S, T) = |\delta_J(S, T)|$ be their number; we let $\delta_J(S) = \delta_J(S, V \setminus S)$ and $d_J(S) = d_J(S, V \setminus S)$. An edge set J **covers** S if $d_J(S) \geq 1$. The following generic meta-problem captures dozens of specific network design problems, among them STEINER FOREST, k -CONSTRAINED FOREST, POINT-TO-POINT CONNECTION, STEINER NETWORK AUGMENTATION, and many more.

SET FAMILY EDGE COVER

Input: A graph $G = (V, E)$ with edge costs $\{c_e : e \in E\}$, a set family \mathcal{F} on V .

Output: A min-cost edge set $J \subseteq E$ such that $d_J(S) \geq 1$ for all $S \in \mathcal{F}$.

In this problem the family \mathcal{F} may not be given explicitly, but we will require that some queries related to \mathcal{F} can be answered in time polynomial in $n = |V|$. An inclusion-minimal set in \mathcal{F} is called an \mathcal{F} -**core**, or just a **core**, if \mathcal{F} is clear from the context. Following previous work, we will require that for any edge set J , the cores of the **residual family** $\mathcal{F}^J = \{S \in \mathcal{F} : d_J(S) = 0\}$ of \mathcal{F} (the family of sets in \mathcal{F} that are uncovered by J) can be computed in time polynomial in $n = |V|$.

Agrawal, Klein and Ravi [2] designed and analyzed a primal-dual algorithm for the STEINER FOREST problem, and showed that it achieves approximation ratio 2. A classic result of Goemans and Williamson [9] from the early 90's shows by an elegant proof that the same algorithm applies for proper set families, where \mathcal{F} is **proper** if it is **symmetric** ($A \in \mathcal{F}$ implies $V \setminus A \in \mathcal{F}$) and has the **disjointness property** (if A, B are disjoint and $A \cup B \in \mathcal{F}$ then $A \in \mathcal{F}$ or $B \in \mathcal{F}$). Slightly later, Williamson, Goemans, Mihail, and Vazirani [12] (henceforth WGMV) further extended this result to the more general class of **uncrossable**



© Zeev Nutov;

licensed under Creative Commons License CC-BY 4.0

50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025).

Editors: Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak; Article No. 82; pp. 82:1–82:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

families ($A \cap B, A \cup B \in \mathcal{F}$ or $A \setminus B, B \setminus A \in \mathcal{F}$ whenever $A, B \in \mathcal{F}$), by adding to the algorithm a novel reverse-delete phase. They posed an open question of extending this algorithm to a larger class of set families and combinatorial optimization problems. However, for 30 years, the class of uncrossable set families remained the most general generic class of set families for which the WGMV algorithm achieves a constant approximation ratio.

Bansal, Cheriyan, Grout, and Ibrahimpur [4] (henceforth BCGI) analyzed the performance of the WGMV algorithm [12] for the following generic class of set families that arise in variants of capacitated network design problems.

► **Definition 1.** Two sets A, B **cross** if all the four sets $A \cap B, V \setminus (A \cup B), A \setminus B, B \setminus A$ are non-empty. A set family \mathcal{F} is **pliable** if $\emptyset, V \notin \mathcal{F}$ and for any $A, B \in \mathcal{F}$ at least two of the sets $A \cap B, A \cup B, A \setminus B, B \setminus A$ belong to \mathcal{F} . We say that \mathcal{F} is γ -**pliable** if it has the following **Property** (γ): For any edge set I and sets $S_1 \subset S_2$ in the residual family \mathcal{F}^I , if \mathcal{F}^I -core (an inclusion minimal set in \mathcal{F}^I) C crosses each of S_1, S_2 , then the set $D = S_2 \setminus (S_1 \cup C)$ is either empty or belongs to \mathcal{F}^I .

BCGI showed that the WGMV algorithm achieves approximation ratio 16 for γ -pliable families, and that Property (γ) is essential – without it the cost of a solution found by the WGMV algorithm can be $\Omega(\sqrt{n})$ times the cost of an optimal solution. Another generalization of uncrossable families is considered in [10]. A set family \mathcal{F} is **semi-uncrossable** if for any $A, B \in \mathcal{F}$ we have that $A \cap B \in \mathcal{F}$ and one of $A \cup B, A \setminus B, B \setminus A$ is in \mathcal{F} , or $A \setminus B, B \setminus A \in \mathcal{F}$. One can verify that semi-uncrossable families are sandwiched between uncrossable and γ -pliable families. The WGMV algorithm achieves the same approximation ratio 2 for semi-uncrossable families, and [10] shows that many problems can be modeled by semi-uncrossable families that are not uncrossable.

The approximation ratio 16 of BCGI [4] for γ -pliable families was improved to 10 in [11]; in fact, the analysis in [11] immediately implies approximation ratio 9, see Lemma 11. Recently Bansal [3] stated an approximation ratio of 8. Here we improve the approximation ratio to 7, and show that this bound is likely to be asymptotically tight for the WGMV algorithm.

► **Theorem 2.** The SET FAMILY EDGE COVER problem with a γ -pliable set family \mathcal{F} admits approximation ratio 7.

A set family \mathcal{F} is **sparse** if for any edge set J , every set $S \in \mathcal{F}^J$ crosses at most one \mathcal{F}^J -core. A particular important case of γ -pliable families arise from the SMALL CUTS COVER problem, in which we seek to cover by a min-cost edge set the set family $\mathcal{F} = \{\emptyset \neq S \subset V : d_H(S) < k\}$ of cuts of size $< k$ of a graph H . It is known that this family is γ -pliable and Bansal [3] made an important observation that this family is sparse. To see this, note that if $S \in \mathcal{F}$ crosses an \mathcal{F} -core C then $C \cap S \notin \mathcal{F}$ and $C \setminus S \notin \mathcal{F}$ by the minimality of C , hence $d_H(C \cap S) \geq k$ and $d_H(C \setminus S) \geq k$. Thus we have $2d_H(C \cap S, C \setminus S) = d_H(C \cap S) + d_H(C \setminus S) - d_H(C) \geq k + 1$. One can see that the cores are pairwise disjoint, hence if S crosses two cores C_1, C_2 then $d_H(S) \geq d_H(C_1 \cap S, C_1 \setminus S) + d_H(C_2 \cap S, C_2 \setminus S) \geq 2\lceil(k+1)/2\rceil$, contradicting that $d_H(S) < k$. Since \mathcal{F}^J is the family of cuts of size $< k$ of the graph $H \cup J$, we get that this \mathcal{F} is sparse.

Bansal [3] stated an approximation ratio of 5 for γ -pliable sparse families, but his proof relies only on the sparsity property and has an error [6]. We note that in one of earlier versions v2 of his arXiv draft [3], along with the 5-approximation for SMALL CUTS COVER he also states a 6-approximation for γ -pliable sparse families. The proof provided relies on several complex reductions and decompositions and many phases of token distribution. We give a relatively simple proof of a 6-approximation for sparse γ -pliable families, that relies on a clear combinatorial statement (Lemma 13), and also give an example that this bound is likely to be asymptotically tight for the WGMV algorithm.

We will also investigate the dependence of the approximation ratio on the “inverse” parameter – the maximum number of pairwise disjoint sets in \mathcal{F}^J that a single core can cross. We say that a set family \mathcal{F} is β -crossing for an integer $\beta \geq 1$ if for any edge set J , an \mathcal{F}^J -core crosses at most β pairwise disjoint sets in \mathcal{F}^J .

► **Theorem 3.** *The SET FAMILY EDGE COVER problem with a γ -pliable sparse set family \mathcal{F} admits approximation ratio 6. If in addition \mathcal{F} is β -crossing then the approximation ratio is $6 - \frac{1}{\beta+1}$.*

The family of cuts of size/capacity $< k$ of a λ -edge-connected graph H is β -crossing for $\beta = \left\lfloor \frac{k-1}{\lceil (\lambda+1)/2 \rceil} \right\rfloor$. To see this, note that if $S \in \mathcal{F}$ crosses an \mathcal{F} -core C then $d_H(S \cap C) \geq k$ since $S \cap C \notin \mathcal{F}$, $d_H(S \setminus C) \geq \lambda$ since H is λ -edge-connected, and $d_H(S) \leq k-1$ since $S \in \mathcal{F}$. Thus we have $2d_H(S \cap C, S \setminus C) = d_H(S \cap C) + d_H(S \setminus C) - d_H(S) \geq \lambda + 1$. If each of p disjoint sets S_1, \dots, S_p in \mathcal{F} crosses C , then each S_i contributes to $\delta_H(C)$ the set $F_i = \delta_H(S_i \cap C, S_i \setminus C)$ of at least $|F_i| \geq \lceil (\lambda+1)/2 \rceil$ edges. The edge sets F_i are pairwise disjoint, thus $k-1 \geq d_H(C) \geq p \cdot \lceil (\lambda+1)/2 \rceil$. Combined with Theorem 3 we get:

► **Corollary 4.** *The SMALL CUTS COVER problem with λ -edge-connected graph H admits approximation ratio $6 - \frac{1}{\beta+1}$, where $\beta = \left\lfloor \frac{k-1}{\lceil (\lambda+1)/2 \rceil} \right\rfloor$.*

The proofs of Theorems 2 and 3 rely on a new structural property of inclusion-minimal solutions that was not known prior to this paper, see Lemma 13.

For additional applications of γ -pliable families for the so called FLEXIBLE GRAPH CONNECTIVITY problems see, for example, [1, 7, 8, 4, 11, 3, 5]. In particular, the second part of Theorem 3 can be used to improve approximation ratios for this problem, c.f. [3].

The rest of this paper is organized as follows. In the next section we will describe the WGMV primal-dual algorithm for pliable set families and show that its approximation ratio is determined by a certain combinatorial problem. Theorems 2 and 3 are proved in Sections 3 and 4, respectively.

2 The WGMV algorithm and pliable families

We start by describing the WGMV algorithm for an arbitrary set family \mathcal{F} . Recall that an inclusion-minimal set in \mathcal{F} is called an \mathcal{F} -core, or just a **core**, if \mathcal{F} is clear from the context; let $\mathcal{C}_{\mathcal{F}}$ denote the family of \mathcal{F} -cores. Consider the following LP-relaxation **(P)** for SET FAMILY EDGE COVER and its dual program **(D)**:

$$\begin{array}{ll}
 \min & \sum_{e \in E} c_e x_e \\
 \text{(P)} \quad \text{s.t.} & \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \in \mathcal{F} \\
 & x_e \geq 0 \quad \forall e \in E
 \end{array}
 \qquad
 \begin{array}{ll}
 \max & \sum_{S \in \mathcal{F}} y_S \\
 \text{(D)} \quad \text{s.t.} & \sum_{\delta(S) \ni e} y_S \leq c_e \quad \forall e \in E \\
 & y_S \geq 0 \quad \forall S \in \mathcal{F}
 \end{array}$$

Given a solution y to **(D)**, an edge $e \in E$ is **tight** if the inequality of e in **(D)** holds with equality. The algorithm has two phases.

Phase 1 starts with $J = \emptyset$ and applies a sequence of iterations. At the beginning of an iteration, we compute the family $\mathcal{C} = \mathcal{C}_{\mathcal{F}^J}$ of \mathcal{F}^J -cores. Then we raise the dual variables corresponding to the \mathcal{F}^J -cores uniformly (possibly by zero), until some edge $e \in E \setminus J$ becomes tight, and add e to J . Phase 1 terminates when $\mathcal{C}_{\mathcal{F}^J} = \emptyset$, namely when J covers \mathcal{F} .

Phase 2 is a “reverse delete” phase, in which we process edges in the reverse order that they were added, and delete an edge e from J if $J \setminus \{e\}$ still covers \mathcal{F} . At the end of the algorithm, J is output.

The produced dual solution is feasible, hence $\sum_{S \in \mathcal{F}} y_S \leq \text{opt}$, by the Weak Duality Theorem. To prove an approximation ratio of ρ , it is sufficient to prove that at the end of the algorithm the following holds for the returned solution J and the dual solution y :

$$\sum_{e \in J} c(e) \leq \rho \sum_{S \in \mathcal{F}} y_S .$$

As any edge in the solution J returned by the algorithm is tight, this is equivalent to

$$\sum_{e \in J} \sum_{\delta_J(S) \ni e} y_S \leq \rho \sum_{S \in \mathcal{F}} y_S .$$

By changing the order of summation we get:

$$\sum_{S \in \mathcal{F}} d_J(S) y_S \leq \rho \sum_{S \in \mathcal{F}} y_S .$$

It is sufficient to prove that at any iteration the increase at the left hand side is at most the increase in the right hand side. Let us fix some iteration, and let \mathcal{C} be the family of cores at the beginning of this iteration. The increase in the left hand side is $\varepsilon \cdot \sum_{C \in \mathcal{C}} d_J(C)$, where ε is the amount by which the dual variables were raised in the iteration, while the increase in the right hand side is $\varepsilon \cdot \rho |\mathcal{C}|$. Consequently, it is sufficient to prove that

$$\sum_{C \in \mathcal{C}} d_J(C) \leq \rho |\mathcal{C}| .$$

Let us use the following notation.

- J_0 is the set of edges picked at Phase 1 before the current iteration.
- $I' = J \setminus J_0$ is the set of edges picked after J_0 and survived the reverse-delete phase.
- $I = \bigcup_{C \in \mathcal{C}} \delta_{I'}(C)$ is the set of edges in I' that cover some $C \in \mathcal{C}$.

► **Lemma 5.** *Let \mathcal{F}' be the residual family of \mathcal{F} w.r.t. $J_0 \cup (I' \setminus I)$. Then:*

- (i) *I is an inclusion-minimal cover of \mathcal{F}' .*
- (ii) *\mathcal{C} is the family of \mathcal{F}' -cores, namely, $\mathcal{C} = \mathcal{C}(\mathcal{F}')$.*

Proof. Let $\mathcal{F}_0 = \mathcal{F}^{J_0}$ be the residual family of \mathcal{F} w.r.t. J_0 , and note that \mathcal{F}' is the residual family of \mathcal{F}_0 w.r.t. $I' \setminus I$.

We prove (i). Since the edges were deleted in reverse order, the edges in I' were considered for deletion when all edges in J_0 were still present. Thus I' is an inclusion-minimal cover of \mathcal{F}_0 . This implies that I is an inclusion-minimal cover of the residual family of \mathcal{F}_0 w.r.t. $I' \setminus I$ (this is so for any $I \subseteq I'$), which is \mathcal{F}' .

We prove (ii). By the definition, \mathcal{C} is the family of \mathcal{F}_0 -cores. No $C \in \mathcal{C}$ is covered by $I' \setminus I$, hence $\mathcal{C} \subseteq \mathcal{C}(\mathcal{F}')$. This also implies that \mathcal{F}' has no other core $C' \notin \mathcal{C}(\mathcal{F}') \setminus \mathcal{C}$, as otherwise $C' \in \mathcal{F}_0$ and thus properly contains some $C \in \mathcal{C}$, which is a contradiction. ◀

Observing that $d_J(C) = d_I(C)$ for all $C \in \mathcal{C}$ (since no $C \in \mathcal{C}$ is covered by $J_0 \cup (I' \setminus I)$), we have the following.

► **Lemma 6.** *The WGMV primal-dual algorithm achieves approximation ratio ρ if for any residual family \mathcal{F}' of \mathcal{F} the following holds: If \mathcal{C} is the family of \mathcal{F}' -cores and I is an inclusion minimal cover of \mathcal{F}' such that every edge in I covers some $C \in \mathcal{C}$ then*

$$\sum_{C \in \mathcal{C}} d_I(C) \leq \rho |\mathcal{C}| . \tag{1}$$

One can see that if an edge e covers one of the sets $A \cap B, A \cup B, A \setminus B, B \setminus A$ then it also covers one of A, B . This implies the following.

► **Lemma 7.** *If \mathcal{F} is pliable or γ -pliable, then so is any residual family \mathcal{F}' of \mathcal{F} .*

Due to Lemmas 6 and 7, to prove that the WGMV algorithm achieves approximation ratio 7 for a γ -pliable family \mathcal{F} , it is sufficient to prove the following purely combinatorial statement.

► **Lemma 8.** *Let I be an inclusion minimal cover of a γ -pliable set family \mathcal{F} such that every edge in I covers some $C \in \mathcal{C}$. Then*

$$\sum_{C \in \mathcal{C}} d_I(C) \leq 7|\mathcal{C}|. \quad (2)$$

A set family \mathcal{L} is **laminar** if any two sets in \mathcal{L} are disjoint or one of them contains the other. Let I be an inclusion minimal edge cover of a set family \mathcal{F} . We say that a set $S_e \in \mathcal{F}$ is a **witness set** for an edge $e \in I$ if e is the unique edge in I that covers S_e , namely, if $\delta_I(S_e) = \{e\}$. We say that $\mathcal{L} \subseteq \mathcal{F}$ is a **witness family** for I if $|\mathcal{L}| = |I|$ and for every $e \in I$ there is a witness set $S_e \in \mathcal{L}$. By the minimality of I , there exists a witness family $\mathcal{L} \subseteq \mathcal{F}$. The following was proved in BCGI [4].

► **Lemma 9** (BCGI [4]). *Let I be an inclusion minimal cover of a pliable set family \mathcal{F} . Then there exists a witness family $\mathcal{L} \subseteq \mathcal{F}$ for I that is laminar.*

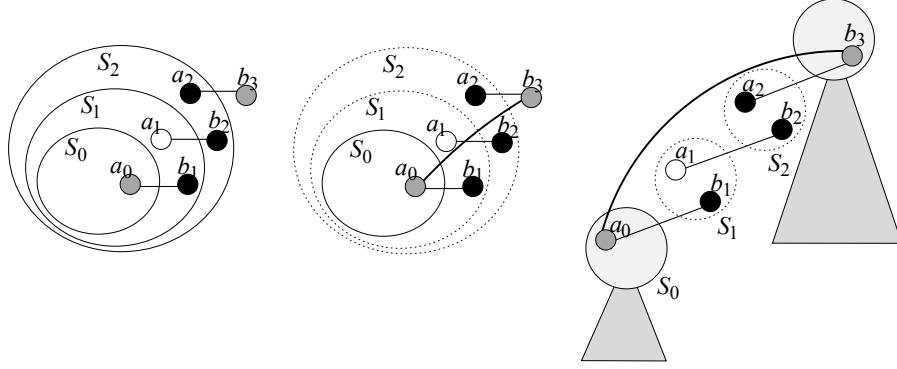
Augment \mathcal{L} by the set V . A set $S \in \mathcal{L}$ **owns** a set C if S is the inclusion-minimal set in \mathcal{L} that contains C . We assign colors to sets in \mathcal{L} as follows: a set is **black** if it owns some core and is **white** otherwise.

► **Definition 10.** *A sequence $\mathcal{S} = (S_1, \dots, S_\ell)$ of sets in $\mathcal{L} \setminus \{V\}$ is a **white chain** if each of S_1, \dots, S_ℓ is white and has exactly one child, where S_{i-1} is the child of S_i , $i = 2, \dots, \ell$. We denote the child of S_1 by S_0 . The edge set of \mathcal{S} is $I_{\mathcal{S}} = \{a_0b_1, \dots, a_\ell b_{\ell+1}\}$, where $a_i b_{i+1}$ is the unique edge in I that covers S_i and $a_i, b_i \in S_i$; see Fig. 2 and note that possibly $a_i = b_i$. The **weight** $w(e)$ of an edge $e \in I$ is the number of cores it covers. The **weight of a white chain** \mathcal{S} is $w(\mathcal{S}) = \sum_{C \in \mathcal{C}} d_{I_{\mathcal{S}}}(C)$; note that $w(e) \leq 2$ for any $e \in I$ and thus $w(\mathcal{S}) \leq 2(\ell + 1)$.*

The laminar family \mathcal{L} can be represented by a rooted tree \mathcal{T} with node set \mathcal{L} and root V , where the parent of S in \mathcal{T} is the smallest set in \mathcal{L} that properly contains S . The (unique) edge in I that covers S corresponds to the edge in \mathcal{T} from S to its parent. We use for nodes of \mathcal{T} the same terminology as for sets in \mathcal{L} ; specifically, nodes of \mathcal{T} are colored white and black accordingly, and a white chain in \mathcal{T} is a path from a node to its ancestor such that all its nodes are white and have degree 2.

Short-cutting a maximal white chain \mathcal{S} as in Definition 10 means removing from \mathcal{L} the sets S_1, \dots, S_ℓ and replacing in I the $\ell + 1$ edges in $I_{\mathcal{S}}$ by the single edge $e = a_0 b_{\ell+1}$ of weight $w(e) = w(\mathcal{S})$ that now has S_0 as the witness set; see Fig. 1. In the tree representation \mathcal{T} of \mathcal{L} this means that we replace the white chain – the edges in $I_{\mathcal{S}}$ and the nodes S_1, \dots, S_ℓ by a new “shortcut edge” e of weight $w(e) = w(\mathcal{S})$ between the set that owns a_0 and the set that owns b_ℓ .

Now let us consider the rooted weighted **shortcut tree** $T = (B \cup W, I), w, r$ (where B is the set of **black nodes** and W is the set of **white nodes**) obtained from \mathcal{T} by short-cutting every maximal white chain. Let L be the set of leaves of T . In what follows, note the following properties of T .



■ **Figure 1** Illustration to the shortcut of a white chain of length $\ell = 2$. Here, the black nodes belong to the same core C , the white node a_1 does not belong to any core. The weight $w(e)$ of the shortcut edge a_0b_3 equals to 3 plus the number of gray nodes that belong to some core. The gray triangles represent the corresponding subtrees in the tree T .

1. $w(I) = \sum_{C \in \mathcal{C}} d_I(C)$, namely, $w(I)$ equals the left-hand side of (1).
2. $|B| \leq |\mathcal{C}|$; every core is owned by exactly one set in \mathcal{L} , since $V \in \mathcal{L}$ and since \mathcal{L} is laminar.
3. In T , every leaf and every non-root node with exactly one child is black; we will call any tree that has this property a **black-white** tree. In particular, T has no white chain (a path of white nodes that have exactly one child each) and thus $|I| \leq 2|\mathcal{C}|$.
4. $|I| = |W| + |B| - 1 \leq 2|B| - 1$ and $|W| \leq |L| \leq |B|$, and if r is black or has at least 2 children then $|W| \leq |B| - 1$.

If the original tree has no white chain of length $> \ell$ then $w(e) \leq \ell$ for all $e \in I$, and thus $\sum_{C \in \mathcal{C}} d_I(C) = w(I) \leq 2(\ell + 1) \cdot 2|\mathcal{C}|$. BCGI [4] showed that the maximum possible length of a white chain is $\ell = 3$, which gives the bound $w(I) \leq 16|\mathcal{C}|$. To improve this bound the following was proved in [11].

► **Lemma 11** ([11]). $w(\mathcal{S}) \leq 5$ for any white chain \mathcal{S} and if $w(\mathcal{S}) = 5$ then S_0 is black.

This immediately implies $w(I) \leq 10$ (this is the bound that was explicitly stated in [11]), but in fact it also easily implies that $w(I) \leq 9|B|$. To see this, let t be the number of edges of weight 5. Then $t \leq |B|$, since by Lemma 11 the tail of every edge of weight 5 is black. Thus since $|W| \leq |B|$ we get

$$w(I) \leq 5t + 4(|W| + |B| - 1 - t) \leq t + 4(2|B| - 1) \leq 9|B| - 4.$$

In the next section we will describe how to improve the analysis of the approximation ratio from 9 to 7.

3 A 7-approximation for pliable families (Theorem 2)

Let $T = (B \cup W, I)$, w be a shortcut tree with root r and leaf set L . For two paths P, P' of T we will write $P \prec P'$ if the nodes of P are descendants of the nodes of P' . We will say that an edge of T is **heavy** if it has weight ≥ 3 . An ordered pair (e, e') of heavy edges is a **bad pair** if $e \prec e'$ and there is no black node between e and e' . Similarly, given two maximal white chains $\mathcal{S}, \mathcal{S}'$ we will write $\mathcal{S} \prec \mathcal{S}'$ if in \mathcal{T} the nodes of \mathcal{S} are descendants of the nodes of \mathcal{S}' , say that a maximal white chain \mathcal{S} is heavy if $w(\mathcal{S}) \geq 3$, and say that a pair of heavy maximal white chains $(\mathcal{S}, \mathcal{S}')$ is a bad pair if $\mathcal{S} \prec \mathcal{S}'$ and there is no black set between S_ℓ and S'_0 . The following lemma proves the desired bound in the case when there are no bad pairs.

► **Lemma 12.** *If T has no bad pair then $w(I) \leq 7|B| - 2$.*

Proof. Let t be the number of heavy edges. There are exactly $|I| - t = |W| + |B| - 1 - t$ non-heavy edges, hence since $|W| \leq |B|$ we have

$$w(I) \leq 5t + 2(|W| + |B| - 1 - t) = 3t + 2(|W| + |B| - 1) \leq 3t + 2(2|B| - 1) .$$

Since all leaves are black and since there is no bad pair, we can assign to every heavy edge the closest descendant black node, and no black node will be assigned twice. Consequently, $t \leq |B|$. Thus we get $w(I) \leq 3t + 2(2|B| - 1) \leq 7|B| - 2$, concluding the proof. ◀

We will prove the following.

► **Lemma 13.** *Let (e, e') be a bad pair. Then:*

1. $w(e) + w(e') \leq 7$.
2. *There is no heavy edge on the path between e and e' .*

Note that Lemma 13 does not imply that the bad pairs are pairwise disjoint; if (e, e') is a bad pair then (e, e') is the unique bad pair that contains e , but there can be many bad pairs $(e_1, e'), \dots, (e_q, e')$ that contain e' . Still, Theorem 2 easily follows from Lemmas 13 and 12 by a simple manipulation of weights. For every edge e' that appears as an upper edge in some bad pair, choose one such bad pair (e, e') and change the weights of e' to be 2 and the weight of e to be $w(e) + w(e') - 2 \leq 5$. This operation does not change the maximum weight nor the total weight, and after it there are no bad pairs, so there is a black node between any two ancestor-descendant heavy edges. Theorem 2 now follows from Lemma 12. Furthermore, the proof shows that if the bound $w(I) \leq 7|B|$ is asymptotically tight, then there exists a tight example without bad pairs.

In the rest of this section we prove Lemma 13. For that, we will need to analyze the white chains of a bad pair (e, e') as in the lemma. Note that in terms of white chains Lemma 13 says that if $(\mathcal{S}, \mathcal{S}')$ is a bad pair of white chains then:

1. $w(\mathcal{S}) + w(\mathcal{S}') \leq 7$.
2. *There is no heavy maximal white chain between \mathcal{S} and \mathcal{S}' .*

In the rest of this section we will prove this “white chains version” of Lemma 13.

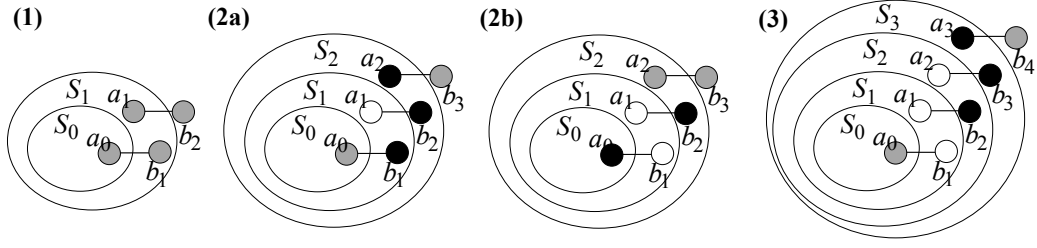
► **Lemma 14.** *Let \mathcal{F} be a pliable set family and let $S \in \mathcal{F}$ and $C \in \mathcal{C}_{\mathcal{F}}$ such that $C \cap S \neq \emptyset$. Then either $C \subseteq S$ or C, S cross and the following holds: $S \setminus C, S \cup C \in \mathcal{F}$ and $C \cap S, C \setminus S \notin \mathcal{F}$. Consequently, the members of $\mathcal{C}_{\mathcal{F}}$ are pairwise disjoint.*

Proof. Suppose that C is not a subset of S . Then $C \setminus S \neq \emptyset$. Also $S \setminus C \neq \emptyset$, since S cannot be a subset of C . By the minimality of C we must have $C \cap S, C \setminus S \notin \mathcal{F}$, thus since \mathcal{F} is pliable $S \setminus C, S \cup C \in \mathcal{F}$. In particular, S, C cross. ◀

In the proof of Lemma 13 we will use the following property of white sets.

► **Lemma 15.** *Let S_{i-1} be a child of a white set $S_i \in \mathcal{L}$ and let $C \in \mathcal{C}$. If $C \cap S_{i-1}$ and $C \setminus S_{i-1}$ are both non-empty then C crosses both S_i, S_{i-1} . Furthermore, if S_{i-1} is the unique child of S_i then $S_i \setminus S_{i-1} \subset C$.*

Proof. Since S_i is white (and thus doesn't own C), $C \setminus S_i \neq \emptyset$. Thus C crosses both S_{i-1}, S_i , by Lemma 14. Now suppose that S_{i-1} is the unique child of S_i . Let $D = S_i \setminus (C \cup S_{i-1})$. By property (γ) either $D = \emptyset$ or $D \in \mathcal{F}$. If $D = \emptyset$ then we are done. Else, $D \in \mathcal{F}$ and thus D contains a core $C' \in \mathcal{C}$, that is owned by a descendant of S_i disjoint to S_{i-1} . This contradicts that S_i has a unique child. ◀



■ **Figure 2** The cases in Lemma 17. Black nodes are in U , white nodes are not in U , while gray nodes may or may not be in U .

Let \mathcal{S} be a maximal white chain as in Definition 10 and let $C \in \mathcal{C}$. The following is proved in [11]; we provide a proof for completeness of exposition.

► **Lemma 16** ([11]). *If $S_0 \cap C \neq \emptyset$ then either $a_1, b_1 \in C$ or C is owned by S_0 or by a descendant of S_0 ; consequently, if $a_0 \in C$ then S_0 owns C . For $i \geq 1$ the following holds:*

- (i) *If $a_i \in C$ then $\ell = i$.*
- (ii) *If $a_i \notin C$ and $b_i \in C$ then $\ell \in \{i, i+1\}$; furthermore, if $\ell = i+1$ then $a_{i+1}, b_{i+1} \in C$.*

Proof. If $S_0 \cap C \neq \emptyset$ and C is not owned by S_0 or by a descendant of S_0 , then by Lemma 15, $\{a_1, b_1\} \subseteq S_1 \setminus S_0 \subset C$. Now assume that $a_0 \in C$ and suppose to the contrary that S_0 does not own C . Then $C \setminus S_0 \neq \emptyset$. By Lemma 15, $S_1 \setminus S_0 \subset C$, hence $b_1 \in C$. Thus the edge $a_0 b_1$ has both ends in C , contradicting the assumption the every edge in I covers some $C \in \mathcal{C}$.

We prove (i). If S_{i+1} exists then by Lemma 15 $b_{i+1} \in C$, contradicting the assumption that every edge in I covers some $C \in \mathcal{C}$.

We prove (ii). If S_{i+1} exists then by Lemma 15 $a_{i+1}, b_{i+1} \in C$, and $\ell = i+1$ follows from part (i). ◀

Let $U = \bigcup_{C \in \mathcal{C}} C$ be the set of those nodes that belong to some core. Using Lemma 16, we obtain the following partial characterization of heavy maximal white chains.

► **Lemma 17.** *If \mathcal{S} is a heavy maximal white chain then exactly one of the following holds.*

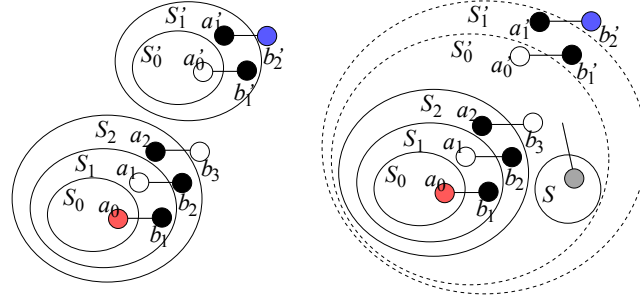
1. $\ell = 1$ and at least 3 among a_0, b_2, a_1, b_2 are in U .
2. $\ell = 2$, $a_1 \notin U$, and one of the following holds:
 - a. $b_1, b_2, a_2 \in C$ for some $C \in \mathcal{C}$.
 - b. $b_1 \notin U$, $a_0, b_2 \in U$, and at least one of a_2, b_3 is in U .
3. $\ell = 3$, $a_1, b_1 \notin U$, and $b_2, b_3, a_3 \in C$ for some $C \in \mathcal{C}$.

Proof. The case $\ell = 1$ is obvious. If $\ell = 2$ then $a_1 \notin U$, by Lemma 16. If $b_1 \in C$ for some $C \in \mathcal{C}$ then by Lemma 16 $a_2, b_2 \in C$ and we arrive at case (2a). Else, $b_1 \notin U$ and since $a_1 \notin U$ we must have $a_0, b_2 \in U$ (since every edge has at least one end in U), and we arrive at case (2b).

If $a_1, b_1, a_2 \notin U$, then $b_2 \in U$ (since the edge $a_1 b_2$ has an end in U), which by Lemma 16 implies $\ell = 3$ and $b_2, b_3, a_3 \in C$ for some $C \in \mathcal{C}$. ◀

► **Lemma 18.** *Let $\mathcal{S} = (S_0, \dots, S_\ell)$ and $\mathcal{S}' = (S'_0, \dots, S'_{\ell'})$ be two heavy white chains with edges $a_0 b_1, \dots, a_\ell b_{\ell+1}$ and $a'_0 b'_1, \dots, a'_{\ell'} b'_{\ell'+1}$, respectively. If $\mathcal{S} \prec \mathcal{S}'$ and there is no black set between \mathcal{S} and \mathcal{S}' then (see Fig. 3):*

1. $w(\mathcal{S}') = 3$, $\ell' = 1$, and $a'_0 \notin U$.
2. $w(\mathcal{S}) \leq 4$ and if $\ell = 1$ then $a_0 \in U$.



■ **Figure 3** Illustration of a bad pair $(\mathcal{S}, \mathcal{S}')$ with $w(\mathcal{S}) + w(\mathcal{S}') = 7$. Blue and red nodes belong to distinct cores, while all black nodes belong to the same core.

Proof. Consider the lower chain \mathcal{S} . By Lemma 17, one of the nodes $a_1, b_1, \dots, a_\ell, b_\ell$ is in C for some $C \in \mathcal{C}$. The core C is not owned by sets in $\mathcal{S} \cup \mathcal{S}'$ nor by sets between \mathcal{S} and \mathcal{S}' , since all these sets are white. Thus C crosses all sets in the upper chain \mathcal{S}' , and in particular the set S'_1 . By Lemma 15 $S'_1 \setminus S'_0 \subseteq C$, and in particular $a'_1 \in C$. This implies $\ell = 1$, by Lemma 16. Moreover, $a'_0 \notin U$, as otherwise by Lemma 16 S'_0 is black, contradicting the assumption that there is no black set between \mathcal{S} and \mathcal{S}' . This proves part 1.

For part 2, we claim that one of $a_\ell, b_{\ell+1}$ is not in U . Suppose to the contrary that $a_\ell \in C$ and $b_{\ell+1} \in C'$ for some distinct $C, C' \in \mathcal{C}$. Then each of C, C' crosses all the sets in \mathcal{S}' , and in particular the first set S'_1 . Thus by Lemma 15 $S'_1 \setminus S'_0 \subseteq C \cap C'$, and in particular $a'_1, b'_1 \in C \cap C'$. This contradicts that the cores are disjoint. Consequently, one of $a_\ell, b_{\ell+1}$ is not in U , which implies $w(\mathcal{S}) \leq 4$, by Lemma 17; note that $w(\mathcal{S}) = 5$ is possible only in case (3) of Lemma 17 when $a_3, b_4 \in U$. If $\ell = 1$, then $a_0 \in U$ as otherwise \mathcal{S} is not heavy. ◀

Lemma 18 already implies the first part 1 of Lemma 13, that $w(\mathcal{S}) + w(\mathcal{S}') \leq 7$. We will show that it also implies part 2. Suppose to the contrary that there is another maximal white chain \mathcal{S}'' between \mathcal{S} and \mathcal{S}' . To obtain a contradiction we apply Lemma 18 twice, as follows.

- Since $\mathcal{S} \prec \mathcal{S}''$, Lemma 18 implies $\ell'' = 1$ and $a''_0 \notin U$.
- Since $\mathcal{S}'' \prec \mathcal{S}'$, Lemma 18 implies that if $\ell'' = 1$ then $a''_0 \in U$.

In the first application $a''_0 \notin U$ while in the second $a''_0 \in U$, arriving at a contradiction.

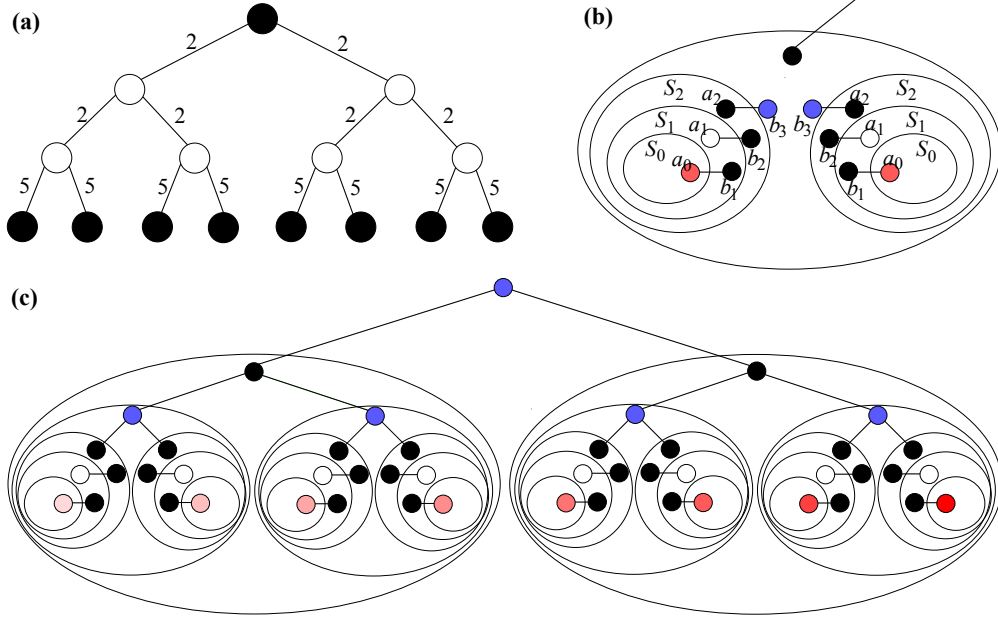
This concludes the proof of Lemma 13, and thus also of Lemma 8 and Theorem 2.

The following example shows that the bound in (2) is asymptotically tight. The shortcut-tree is a binary tree with black nodes $B = L \cup \{r\}$ and weights 5 for leaf edges while all the other edges have weight 2; see Fig. 4 for an illustration for the case $|L| = 8$. To materialize this tree in terms of the laminar family \mathcal{L} and a set \mathcal{C} of cores, do the following.

- Replace every leaf edge by the gadget as in case (2a) in Lemma 17 where $a_0, b_3 \in U$ belong to distinct cores.
- Every other edge will connect two distinct cores, when the same cores are used for distinct edges.

Every node colored by a shade of red is a core (we used distinct shades of red to indicate that these cores are distinct), and is a leaf in the laminar family \mathcal{L} . There are two additional cores – one contains all black nodes and the other all blue nodes; these two cores are owned by the root V . The number of cores is $|\mathcal{C}| = |L| + 2$, while the total weight of the edges in the shortcut tree is $5|L| + 2(|L| - 1) = 7|L| - 2$.

Note that this example shows only the edge set I , the laminar witness family \mathcal{L} , and the set \mathcal{C} of cores, but does not specify the entire γ -pliable set family \mathcal{F} . Such a family \mathcal{F} should have the following properties.



■ **Figure 4** Construction of a tree \mathcal{T} of weight $7|L| - 2$ and a set of $|L| + 2$ cores. (a) The shortcut tree. (b) The gadgets. (c) The laminar family \mathcal{L} and the set \mathcal{C} of cores.

- (i) \mathcal{F} contains the laminar family \mathcal{L} and the set \mathcal{C} of cores as in the example.
- (ii) I is an inclusion minimal cover of \mathcal{F} .
- (iii) \mathcal{F} is γ -pliable.

Such a family \mathcal{F} can probably be obtained by an iterative process that starts with $\mathcal{F} = \mathcal{L} \cup \mathcal{C}$ and repeatedly adds to \mathcal{F} at least two sets among $A \cap B, A \setminus B, B \setminus A, A \cup B$ for any pair $A, B \in \mathcal{F}$. We will not describe the full construction here.

4 Improved approximation for sparse families (Theorem 3)

Recall that a set family \mathcal{F} is sparse if for any edge set J , every set $S \in \mathcal{F}^J$ crosses at most one \mathcal{F}^J -core. This implies that if \mathcal{F} is sparse, then so is any residual family \mathcal{F}' of \mathcal{F} . Due to this and Lemmas 7 and 6, to prove that the WGMV algorithm achieves approximation ratio 6 for a γ -pliable sparse family \mathcal{F} , it is sufficient to prove the following.

► **Lemma 19.** *Let I be an inclusion minimal cover of a γ -pliable sparse set family \mathcal{F} such that every edge in I covers some $C \in \mathcal{C}$. Then*

$$\sum_{C \in \mathcal{C}} d_I(C) \leq 6|\mathcal{C}| - 2. \quad (3)$$

In the proof of Lemma 19 we will use the first part of the following lemma.

► **Lemma 20.** *If \mathcal{F} is sparse then for any edge e of the shortcut tree the following holds:*

- *If $w(e) = 5$ then both ends of e are black.*
- *If $w(e) = 4$ then at least one end of e is black.*

Proof. Let \mathcal{S} be a white chain. By Lemma 17, if $w(\mathcal{S}) = 5$ then $a_0, a_\ell, b_{\ell+1} \in U$; see cases (2a) and (3) in Figure 2 and Lemma 17. Thus by Lemma 16 S_0 is black (since $a_0 \in U$). Note that $a_\ell, b_{\ell+1}$ belong to distinct cores, say $a_\ell \in C$ and $b_{\ell+1} \in C'$. Let S be the parent of S_ℓ .

Since \mathcal{F} is sparse, at least one of C, C' cannot cross S , and thus is owned by S . Hence S is also black. If $w(\mathcal{S}) = 4$ then $a_0 \in U$ and then S_0 is black, or $a_\ell, b_{\ell+1} \in U$ and then the parent S of S_ℓ is black. \blacktriangleleft

Let $T = (W \cup B, I), r, w$ be the shortcut tree; recall that T is a black-white tree, namely, every non-root node with exactly one child is black. We already know that $w(e) \leq 5$ and if $w(e) = 5$ then e has its lower end in B (by Lemma 11); Lemma 20 adds the property that if $w(e) = 5$ then e has both ends in B . Furthermore, Lemma 18 implies the following.

► **Corollary 21.** *For any bad pair (e, e') of T the following holds.*

1. *There is no heavy edge between e and e' .*
2. *$w(e) \leq 4$, $w(e') = 3$ and e' has an upper end in B .*

For a heavy edge e let B_e be the set of black nodes $b \in B$ such that b is a descendant of e and there is no heavy edge between e and b . Note the following.

- $B_e = \emptyset$ if and only if e is an upper edge of a bad pair.
- $B_e \cap B_f = \emptyset$ for distinct e, f .

Let H^* be the set of heavy edges e such that e is not the upper edge of a bad pair, so $B_e \neq \emptyset$ for all $e \in H^*$. For every $e \in H^*$ choose one node $b_e \in B_e$ and let $B^* = \{b_e : e \in H^*\}$ be the set of chosen black nodes. Now we will prove the following lemma, that implies Lemma 19.

► **Lemma 22.** *$w(I) \leq 3|B| + |L| + 2|B^*| - 2$ and $w(I) \leq 3|B| + |L| + 2|B^*| - 4$ if r is black or has at least 2 children.*

Proof. Assign tokens to nodes in B as follows:

- 4 tokens to every node in L .
- 3 tokens to every node in $B \setminus L$.
- 2 additional tokens to every node in B^* .

The number of tokens is $4|L| + 3|B \setminus L| + 2|B^*| = 3|B| + |L| + 2|B^*|$. We will show that these tokens can be redistributed such that every $e \in I$ gets $w(e)$ tokens and some tokens remain.

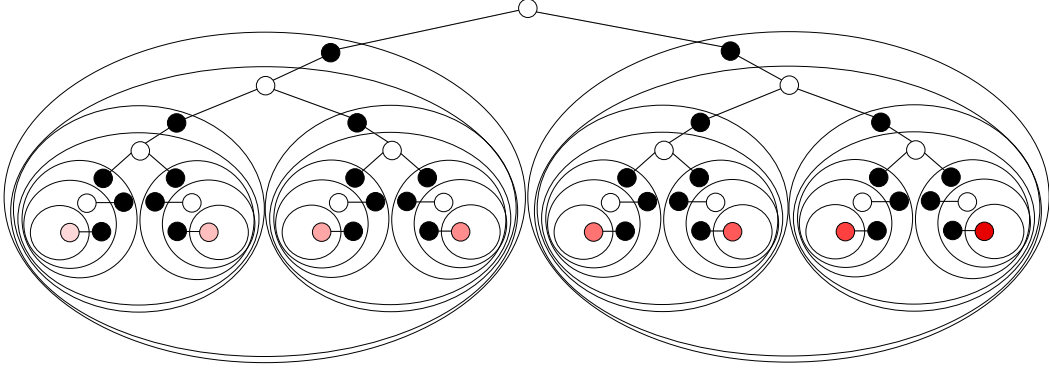
For each $e \in H^*$ use 2 tokens of b_e to reduce the weight of e by 2. Then we have the following.

- Every leaf has 4 tokens and every internal black node has 3 tokens.
- The maximum weight is 3 since initially the maximum weight was 5 and since the upper edge of every bad pair has weight exactly 3, by Corollary 21.
- Every edge of weight 3 has its upper end in B , since every edge of weight 5 and the upper edge of every bad pair has its upper end in B , by Lemma 20 and Corollary 21.

For $v \in V$ let T_v be the rooted subtree of T that consists of v and its descendants. We claim that for any $v \neq r$, the tokens of T_v can be redistributed such that every edge gets at least $w(e)$ tokens and the root v gets 4 tokens. The proof is by induction on the height of the tree. The induction base case, when v is a leaf, is trivial. Suppose that v is not a leaf and has $p \geq 1$ children. By the induction hypothesis each child of v has 4 tokens.

Suppose that v is white. Then $p \geq 2$. Each child of v is connected to v by an edge of weight ≤ 2 (since the upper end of every edge of weight 3 is black), and this child can pay for its parent edge and give 2 tokens to v . Thus v gets $2p \geq 4$ tokens.

Suppose that v is black, so $v \in B \setminus L$. Then v already has 3 tokens. Each child of v is connected to v by an edge of weight ≤ 3 . Thus in this case each child can pay for his parent edge and give 1 token to v . Thus v gets $3 + p \geq 4$ tokens.



■ **Figure 5** Construction of a tree T of weight $6|L| - 2$ and a set of $|L| + 1$ cores. Any two red nodes belong to distinct cores, while all black nodes belong to the same core. (a) The shortcut tree. (b) The gadgets. (c) The laminar family and the cores.

Now let us consider the root r of T that has $p \geq 1$ children. If r is white then it gets $2p \geq 2$ tokens. If r is black then it gets $3 + p \geq 4$ tokens, and if r has at least 2 children then it gets $3 + p \geq 4$ tokens. ◀

Lemma 22 implies that $w(I) \leq 6|B| - 2 \leq 6|C| - 2$, thus concluding the proof of the first part of Theorem 3. The following example shows that the bound $w(I) \leq 6|B|$ is asymptotically tight even when there are no bad pairs. The shortcut-tree is a binary tree with black nodes $B = L \cup \{r\}$ and weights 4 for leaf edges while all the other edges have weight 2; see Fig. 5 for an illustration for the case $|L| = 8$. To materialize this tree in terms of the laminar family and cores, do the following.

- Replace every leaf edge by the Lemma 17(2a) gadget, where $a_0 \in U$ and $b_3 \notin U$.
- Replace every other edge by the gadget as in case (1) in Lemma 17 where $a_1, b_1 \in U$ and $a_0, b_3 \notin U$; this is a “redundant” (non-heavy) white chain of weight 2.

Every node colored by a shade of red is a core (we used distinct shades of red to indicate that these cores are distinct), and there is one additional core – the one that contains all black nodes; this core is owned by the root V . The number of cores is $|L| + 1$, while the total weight is $4|L| + 2(|L| - 1) = 6|L| - 2$.

Note that in this example every member of the laminar family is crossed by at most one core, and that there are no bad pairs in this example.

The example again shows only the edge set I , the laminar witness family \mathcal{L} , and the set \mathcal{C} of cores, but does not specify the entire γ -pliable sparse set family \mathcal{F} . Such a family \mathcal{F} should have the following properties.

- (i) \mathcal{F} contains the laminar family \mathcal{L} and the set \mathcal{C} of cores as in the example.
- (ii) I is an inclusion minimal cover of \mathcal{F} .
- (iii) \mathcal{F} is γ -pliable and sparse.

Such a family \mathcal{F} can be obtained by an iterative process that starts with $\mathcal{F} = \mathcal{L} \cup \mathcal{C}$ and repeatedly adds to \mathcal{F} at least two sets among $A \cap B, A \setminus B, B \setminus A, A \cup B$ for any pair $A, B \in \mathcal{F}$. We will not describe the full construction here.

Note that the above example is not for the SMALL CUTS COVER problem, but rather for an arbitrary sparse γ -pliable family. In fact, Corollary 4 states that SMALL CUTS COVER admits a better approximation ratio $6 - \frac{1}{\beta+1}$, where $\beta = \left\lfloor \frac{k-1}{\lceil (\lambda+1)/2 \rceil} \right\rfloor$ and λ is the edge-connectivity of the input graph H . This approximation ratio is better than 6 when λ is not much smaller than k .

To prove the second part of Theorem 3 we prove the following.

► **Lemma 23.** *Let I be an inclusion minimal cover of a γ -pliable sparse β -crossing set family \mathcal{F} such that every edge in I covers some $C \in \mathcal{C}$. Then*

$$\sum_{C \in \mathcal{C}} d_I(C) \leq 5|\mathcal{C}| + \frac{|\mathcal{C}|}{1 + 1/\beta} = \left(6 - \frac{1}{\beta + 1}\right) |\mathcal{C}|. \quad (4)$$

Proof. We claim that

$$|\mathcal{C}| \geq |L| + |L \cap B^*|/\beta \geq |L \cap B^*| \cdot (1 + 1/\beta)$$

To see this consider the set of edges $I^* = \{e \in H^* : b_e \in L \cap B^*\}$; namely, $e \in I^*$ if $e \in H^*$ (so e is heavy but is not the upper edge of a bad pair) and the black node b_e assigned to e is a leaf. Note the following.

- For any two edges in I^* , none of them is a descendant of the other, since if $e \in I^*$ has a descendant edge $f \in I^*$, then b_e is between e and f contradicting that b_e is a leaf.
- For every $e \in I^*$ there is a core $C_e \in \mathcal{C}$ that crosses all the sets in the white chain of e . For every $e \in I^*$ chose some set S_e from the white chain of e . The sets S_e are pairwise disjoint. Since \mathcal{F} is β -crossing, in any set of $\beta + 1$ edges from I^* there are two edges e, f with $C_e \neq C_f$. Consequently

$$|\{C_e : e \in I^*\}| \geq |I^*|/\beta = |L \cap B^*|/\beta.$$

There are also $|L|$ cores contained in leaves of T . Thus

$$|\mathcal{C}| \geq |L| + |\{C_e : e \in I^*\}| \geq |L| + |L \cap B^*|/\beta.$$

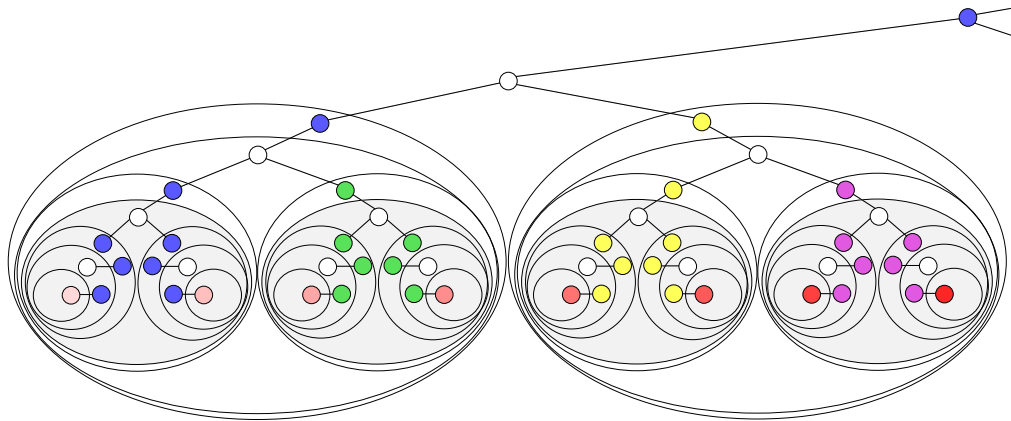
By Lemma 22 $w(I) - 3|B| - |B^*| \leq |L| + |B^*|$. Thus since $L \cup B^* \subseteq B$ we get

$$w(I) - 3|B| - |B^*| \leq |L| + |B^*| = |L \cup B^*| + |L \cap B^*| \leq |B| + |\mathcal{C}|/(1 + 1/\beta)$$

Consequently, $w(I) \leq 4|B| + |B^*| + |\mathcal{C}|/(1 + 1/\beta) \leq 5|\mathcal{C}| + |\mathcal{C}|/(1 + 1/\beta)$, as required. ◀

Note that we proved approximation ratio $\frac{6\beta+5}{\beta+1}$. We can provide an example without edges of weight 5 and without bad pairs such that $w(I) \approx \frac{6\beta}{\beta+1}$. Consider the example in Fig. 5. Suppose that $|L| = 2^i$ and $\beta = 2^j$ for some $0 \leq j < i$. Fig. 6 illustrates the construction for $j = 1$ and $i = 3$, where two nodes are colored by the same color if they belong to the same core. Consider the minimal rooted subtrees of T with exactly $\beta = 2^j$ leaves. Each such subtree will be exactly as in the example in Fig. 5 – the leaves are colored by distinct colors, while the other nodes in U all have the same color, which we call “the color of the subtree”. For each subtree, its root is not in U (the union of all cores), while the parent of the root is in U and has the color of the subtree. The grandparent of the root is again a node not in U and joins two subtrees; the parent of the grandparent will be colored by the color of one of these subtrees. In a similar manner we propagate the colors upwards, where a node not in U joins two subtrees and its parent is colored by the color of one of these two subtrees. Note that such “coloring” does not violate the β -crossing property. In the constructed example, $w(I) = 6|L| - 2$ (the weight is the same as in the example in Fig. 5) and $|\mathcal{C}| = |L| + |L|/\beta$, hence when $|L|$ is large $w(I)/|\mathcal{C}| \approx \frac{6}{1+1/\beta} = \frac{6\beta}{\beta+1}$.

Our paper still leaves an open questions concerning the SMALL CUTS COVER problem, whether an approximation significantly better than 6 is possible, by the primal-dual algorithm or by some other method.



■ **Figure 6** Illustration of the construction of a tree with $w(I) = 6|L| - 2$ and $|C| = |L| + |L|/\beta$ with $|L| = 2^3$ leaves and $\beta = 2^1$.

References

- 1 D. Adjiashvili, F. Hommelsheim, and M. Mühlenhaller. Flexible graph connectivity. *Mathematical Programming*, pages 1–33, 2021.
- 2 A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. on Computing*, 24(3):440–456, 1995. doi:10.1137/S0097539792236237.
- 3 I. Bansal. A global analysis of the primal-dual method for pliable families, July 2024. arXiv:2308.15714.
- 4 I. Bansal, J. Cheriyan, L. Grout, and S. Ibrahimpur. Improved approximation algorithms by generalizing the primal-dual method beyond uncrossable functions. In *ICALP*, pages 15:1–15:19, 2023.
- 5 I. Bansal, J. Cheriyan, S. Khanna, and M. Simmons. Improved approximation algorithms for flexible graph connectivity and capacitated network design, 2024. doi:10.48550/arXiv.2411.18809.
- 6 Ishan Bansal. personal communication.
- 7 S. C. Boyd, J. Cheriyan, A. Haddadan, and S. Ibrahimpur. Approximation algorithms for flexible graph connectivity. In *FSTTCS*, pages 9:1–9:14, 2021.
- 8 C. Chekuri and R. Jain. Approximation algorithms for network design in non-uniform fault models. In *ICALP*, volume 261, pages 36:1–36:20, 2023. doi:10.4230/LIPICS.ICALP.2023.36.
- 9 M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995. doi:10.1137/S0097539793242618.
- 10 Z. Nutov. Extending the primal-dual 2-approximation algorithm beyond uncrossable set families. In *IPCO*, pages 351–364, 2024.
- 11 Z. Nutov. Improved approximation algorithms for covering pliable set families and flexible graph connectivity. In *WAOA*, pages 151–166, 2025.
- 12 D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15(3):435–454, 1995. doi:10.1007/BF01299747.