Tree Embeddings for Two-Edge-Connected Network Design

Anupam Gupta*   Ravishankar Krishnaswamy*   R. Ravi†

Abstract

The group Steiner problem is a classical network design problem where we are given a graph and a collection of groups of vertices, and want to build a min-cost subgraph that connects the root vertex to at least one vertex from each group. What if we wanted to build a subgraph that *two-edge-connects* the root to each group—that is, for every group $g \subseteq V$, the subgraph should contain two edge-disjoint paths from the root to some vertex in g? What if we wanted the two edge-disjoint paths to end up at *distinct* vertices in the group, so that the loss of a single member of the group would not destroy connectivity?

In this paper, we investigate tree-embedding techniques that can be used to solve these and other 2-edge-connected network design problems. We illustrate the potential of these techniques by giving poly-logarithmic approximation algorithms for two-edge-connected versions of the group Steiner, connected facility location, buy-at-bulk, and the k-MST problems.

1 Introduction

Edge survivability has long been a desired property in network design, and problems enforcing higher edge-connectivity have been well studied in the literature. We now have very strong approximation results for some of the basic problems, like the edge-survivable (and element-survivable) network design problems [29, 19], which have been recently extended to the case of vertex connectivity as well [12]. The techniques that have proved useful for these results are primal-dual algorithms (which were used for the first few results here) and subsequently, iterative rounding, which gave much stronger results.

However, higher-connectivity versions of several other network design problems still lack good approximations: let us consider the *group Steiner tree* problem, where given a rooted undirected graph, and subsets of vertices (called groups), the goal is to find a minimum cost subgraph that contains paths from the root to at least one vertex in each group. *What if we wanted two edge-disjoint paths to at least one vertex in each group?* A key difficulty in addressing this problem is that all known solution methods for the singly-connected version first reduce the given problem instance to one where the graph is a tree which approximately preserves pairwise distances; one can then either write a LP relaxation and round it, or use a clever greedy algorithm and dynamic programming, to obtain an approximation. In fact, it has been a long-standing open problem to obtain a logarithmic approximation guarantee in polynomial time that does not use the method of tree-approximations. Note that reducing to a tree instance is bad for us when a 2-edge-connected graph is desired, since we have lost the higher connectivity in the very first (but crucial) step.

In earlier work [23] on online survivable network design problems, we observed that approximating the given graph by a random spanning tree [1], we *need not discard* the non-tree edges, but can just raise their lengths to match the distance along the tree between their end-points. Hence the random tree-embedding can now be viewed as a random embedding into a *backboned graph*: one that has a "backbone" spanning tree such that the cost of a non-tree edge is at least that of the tree path between its end vertices. This enables us to write linear programming relaxations as in the singly-connected versions, and moreover, the modified costs on non-tree edges gives us the additional structure we can use to achieve 2-connectivity. Using this approach, in Section 3, we give a $O(\log^3 n \log q)$-approximation algorithm for instances of the 2-edge-connected group Steiner tree problem with q groups.

We show how similar ideas can be used to solve other 2-edge-connectivity problems. In Section 4, we consider the two-edge-connected version of the *connected facility location problem* (2-CFL). As in facility location, we are given clients with demands in an undirected network, and must open a set of facilities (paying facility opening costs) and assign the clients to some open facility (paying a connection cost equal to the shortest-path distance between them). But we must also build a two-edge-connected network (the "core") on

the open facilities, paying M times the cost of the edges in this 2-connected core. The motivation is a commonly-faced one for network designers: it is crucial to achieve fault-tolerance in the core of the network. This problem has been studied when the network is a complete graph and the costs satisfy the triangle inequality, but nothing was known for the general graph case even for the simple case $M = 1$ [45]. We use our general technique to reduce the problem to backboned networks, where we give a constant approximation—hence giving us an $\widetilde{O}(\log n)$ approximation for general graphs. We also give a poly-logarithmic approximation for the case where the connection costs and the core network costs are unrelated.

In Section 5, we give a $O(\log^2 n)$-approximation algorithm for the 2-edge-connected buy-at-bulk problem with concave scaling costs for buying cables. In this problem, we are given a graph and a set of demand pairs (s_i, t_i) that require 2-edge-connectivity from s_i to t_i. A feasible solution is a collection of two edge-disjoint paths for every (s_i, t_i) pair, and the cost incurred by an edge e in such a solution is $c(e) \cdot \Phi(l(e))$ where $c(e)$ is the length/distance of edge e, $l(e)$ is the *load* on edge e (the number of demand pairs using e), and $\Phi(\cdot)$ is a concave scaling function that models the economies of scale phenomenon. The goal is then to minimize the total cost on all the edges used. This problem was first studied (in the more general 2-vertex-connectivity setting) by Antonakopoulos et al. [2],where they showed a $O(\log^3 n)$-approximation for the (single-sink) buy-at-bulk problem, when there is only one cable type. What we show is that the additional properties of backboned graphs can be leveraged in order to separate the problem into that of buying tree paths and covering them appropriately. This structure enables us to get our $O(\log^2 n)$-approximation algorithm for the 2-edge-connectivity version of (multi-commodity) buy-at-bulk under any concave scaling function.

Finally, in Section 6 we also show how essentially the same techniques can be used to give a poly-logarithmic approximation for the k-2EC problem, which is a generalization of k-MST to higher connectivity. Here, we want to find a minimum-cost subgraph of a given graph G that contains at least k nodes and is 2-edge-connected. The first approximations to this problem were given only recently by Lau et al. [38], and improved by Chekuri and Korula [6] (whose solution also works for the node-connected case). We show that our framework also gives an $\tilde{O}(\log^3 n)$-approximation for the k-2EC problem: while our guarantees are quantitatively worse than those in the previous results, our proof shows how simple ideas can be used to obtain results in the same ballpark.

1.1 Related Work

1.1.1 Higher Connectivity problems. There is a huge body of work on higher connectivity problems. A long stream of work has studied the 2-edge-connected spanning subgraph problem: Frederickson and Ja'Ja' [20] gave the first 3-approximation algorithm by augmenting a minimum spanning tree, showing also in the process that the problem of augmenting any spanning tree to make it 2-edge-connected can be approximated within a factor of 2. This was subsequently improved by Khuller and Vishkin [33], who showed a 2-approximation for the general k-edge-connected spanning subgraph problem. Then, primal dual algorithms [34, 49, 22] were used to obtain $O(\log k)$ approximations for more general k-connectivity problems. Jain [29] gave an iterative rounding based 2-approximation algorithm for the general edge-connectivity survivable network design problem. These techniques have also been employed recently to obtain tight results for network design with degree constraints [38, 40, 3]. The element-connectivity and $\{0, 1, 2\}$-node connectivity versions were solved by Fleischer et al. [18, 19]. The generalized vertex connectivity problems are less well understood: [32, 9, 37, 17] give approximations for the k-vertex-connected spanning subgraph problem, while Cheriyan and Vetta [10] consider the general problem on instances with a complete metric. Recent papers [4, 7, 11, 43, 44] have shed more insight into the subset k-node-connectivity case, and very recently, Chuzhoy and Khanna [12] show that element connectivity can be used as a black-box to give good approximations for the generalized vertex-connectivity version via an elegant sampling idea. From the inapproximability side, Kortsarz et al. [36] give $\Omega(2^{\log^{1-\varepsilon} n})$ hardness for node-connected SNDP, and [4] give T^ε hardness for node-connecting T pairs.

1.1.2 Group Steiner Tree (GST). An LP rounding algorithm for the group Steiner problem with was given by Garg et al. [21]; an alternate greedy algorithm avoiding the LP rounding was given by Chekuri et al [5]. Similar poly-logarithmic approximations are also known for the *covering Steiner problem*, a generalization of the group Steiner tree problem where a requirement r_i is given with each group g_i and we require a minimum cost subgraph that (one-)connects at least r_i terminals from each group g_i to the root [35, 26]. Note the covering Steiner problem *does not* solve the 2-ECGS problem we consider here, since the paths from the root to two nodes from a group may share edges. Poly-logarithmic integrality gaps and hardness results are known for all these group and covering Steiner problems [27, 28]. Very

recently and independent of our work, Khandekar et al. [31] also consider 2-connected group Steiner problems and give an $O(k \log^2 n)$-approximation for the setting when vertex-disjoint paths were required to two distinct vertices, when groups have size at most k.

1.1.3 Connected Facility Location (CFL). This problem has been very widely studied in the approximation algorithms literature—here we want facilities to be (singly-)connected together by a Steiner tree. Several constant-factor approximations are known, based on ideas like LP rounding [45, 24], reduction to classical facility location [30], primal-dual methods [47], and random sampling of facilities [25, 13, 14, 48, 15]. We note that a special case of the 2-connected version we study here (called the "ring-star" problem or "tour-CFL") in which the underlying graph is a complete graph and the edge costs satisfy the triangle inequality was studied in [45]. The observation that an Euler tour can give a TSP with cost twice the Steiner tree cost implies that this is essentially equivalent to the 1-connected CFL. This proof breaks down when the graph is not complete.

A different version of the two-connected CFL problem can also be formulated, where we have to pick two edge-disjoint paths to connect each demand to its facility, and also build a two-connected subgraph on the facilities. A constant-factor approximation for this problem can be obtained from previous random sampling techniques; we give the details in the full version. Again, these techniques do not seem to extend to our case: loosely, this new version implies that demands are cheaply two-connected to each other, and hence opening up a subset of them may be a feasible solution; this is certainly not the case for the 2-CFL problem we study.

1.1.4 The two-edge-connected buy-at-bulk problem. Antonakapoulos et al. [2] first studied fault tolerant versions of the *single cable* buy-at-bulk problem. They showed a constant approximation for the single-sink case and a $O(\log^3 n)$-approximation for the multi-commodity setting of 2-vertex-connected buy-at-bulk Subsequently, Chekuri and Korula showed an $O(\log |T|^b)$-approximation algorithm for the *single-sink* 2-vertex-connected buy-at-bulk problem with b cable types and any set of T demand pairs. In this work, we show an $O(\log^2 n)$-approximation algorithm for the multiple cable multicommodity problem. However, we should note that the previous approximations hold for the more general setting of 2-vertex-connectivity, while our algorithm solves the 2-edge-connectivity problem.

1.1.5 The k-two-edge-connected subgraph problem. The k-2EC problem was first studied by Lau et al. [38] who claimed an $O(\log^3 k)$-approximation algorithm; this was corrected to an $O(\log n \log k)$-approximation [39]. Independently, Chekuri and Korula gave an $O(\log n \log k)$-approximation for the 2-node-connectivity version of the problem. At a high level both these algorithms use the idea of finding repeatedly dense subgraphs, and then pruning the resulting graph to have the right number of terminals. These ideas give better approximation guarantees than we do, but require more machinery; we show how simple ideas can give non-trivial approximations.

2 Backboned Graphs

In [23] we noted that the standard techniques used for approximating graph metrics by distributions over their subtrees implied that graphs could be well-approximated by random graphs with "nice" structure, which we called backboned graphs. While this is a trivial observation, it opens up the possibility of leveraging the added structure to design LP rounding algorithms, much like tree embeddings have been used. In this section, let us give the basic definitions we will use in the rest of the paper.

2.1 Backboned Graphs and Tree Embeddings

DEFINITION 2.1. ([23]) *A graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_{\geq 0}$ is called a* backboned graph *if there exists a spanning tree $T = (V, E(T))$ such that all edges $e = \{u, v\} \notin E(T)$ have the property that $c(e) \geq d_T(u, v)$, where $d_T(u, v)$ is the distance between u and v along T. In this case, T is called the* base tree *of G.*

The following result is a simple consequence of the results of Elkin et al. and Abraham et al. [16, 1]. We give the proof in Appendix A for completeness.

THEOREM 2.1. *Given a network-design problem Π whose objective function is linear in the edge-costs, any β-approximation algorithm for the problem Π on backboned graphs implies a randomized $\beta \times \widetilde{O}(\log n)$-approximation algorithm for Π on general graphs.*

Given this reduction, for the subsequent sections we will assume that the input graph is a backboned graph, and will use its properties to design our algorithms.

2.2 A Covering Lemma on Backboned Graphs

We begin with some notation. Let G be a backboned graph with base tree T. For any non-tree edge $f = \{u, v\}$, let $P_T(u, v)$ denote the base tree path from u to v, and let O_f denote the fundamental cycle with respect to T; i.e. $O_f = \{f\} \cup P_T(u, v)$. Because G is backboned graph, observe that the cost of the cycle

$c(O_f) \leq 2c(f)$, where $c(A) = \sum_{e \in A} c(e)$. A subgraph H is said to be *cycle-closed* if it satisfies the property that $f \in E(H) \setminus E(T) \Leftrightarrow O_f \in E(H)$; i.e., if a non-tree edge f is present in H, then the entire cycle O_f is in H.

OBSERVATION 2.1. *For any subgraph H, there exists a cycle-closed subgraph H' such that $H \subseteq H'$ and the cost of H' is at most $2c(H)$.*

OBSERVATION 2.2. *For any two vertices $u, v \in V$, if a cycle-closed subgraph H contains some $u - v$ path, then H also contains the base tree path $P_T(u,v)$.*

The first observation is true because we can include the entire cycle O_f in H' for every non-tree edge $f \in H$, and the cost at most doubles. The second observation follows from the definition of *cycle-closed* subgraphs: for any non-tree edge $\{x, y\} \in H$, we know that the path $P_T(x, y) \subseteq O_{\{x,y\}} \subseteq H$. Hence, by transitivity, for any u-v path in H, the path $P_T(u, v) \subseteq H$.

We now prove a simple but crucial property of 2-edge-connected subgraphs on backboned graphs.

LEMMA 2.1. (COVERING LEMMA) *Let H be any cycle-closed subgraph that 2-edge-connects a vertex r with a vertex $v \neq r$. Then for any edge e on the base tree path $P_T(r, v)$, there exists an edge $f = \{x, y\} \in E(H)$ such that $e \in O_f$. Therefore, r and v are connected in $(P_T(r, v) \cup O_f) \setminus \{e\}$.*

Proof. Consider an edge e on the base tree path $P_T(r, v)$. Removing the edge e would separate the base tree T into two components, one containing r (which we call C_r) and the other containing v (denoted by C_v). Since r and v are 2-edge-connected in the subgraph H and e is the only tree edge crossing C_r and C_v, there must exist a non-tree edge $f = \{x, y\} \in E(H) \setminus E(T)$ such that one end vertex of f is in C_r and the other is in C_v; otherwise e would be a cut edge separating r and v in H. But then, since x and y are in different components of $T \setminus e$, it follows that $e \in P_T(x, y) \subseteq O_f$. This completes the proof. ∎

3 2-Edge-Connected Group Steiner

In this section, we consider the 2-edge-connectivity extension of the *group Steiner* problem, which we call 2-ECGS, and give an $O(\log^2 n \log q)$-approximation algorithm for instances with backboned graphs, where q is the number of groups. Formally, we are given a graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}$, a set of groups $\mathcal{G} = \{g_1, g_2, \ldots, g_q\}$ where $g_i \subseteq V$, and a designated root $r \in V$. The objective is to find a minimum cost subgraph H and identify representatives $r_i \in g_i$ (for

$1 \leq i \leq q$) such that r_i and r are connected by *two edge-disjoint paths* in H. (One can consider a variant of the problem where it is sufficient have two edge-disjoint paths to the group g_i, possibly to different vertices: we consider this in Section 3.2.)

At a high level, our techniques for solving the 2-ECGS problem use the underlying base tree in the backboned graph to set up a linear program using ideas from the LP relaxations for group Steiner trees [21] and the tree augmentation problem [8] (where non-tree edges must be added to 2-edge-connect a tree). Our LP identifies terminals that will be fractionally 1-connected from the root *along the base tree*; the non-tree edges then (fractionally) 2-connect these terminals, which is enforced by tree augmentation constraints. Our algorithm then employs the group Steiner rounding, and follows this up with a second stage of choosing non-tree edges to 2-connect the first stage subtree. The crux of our analysis is to show that the expected cost of the second stage solution is no more than an extra logarithmic factor of the original LP cost, and this argument uses the level structure of the group Steiner LP rounding in a careful way.

3.1 An $O(\log^2 n \log q)$ Approximation for 2-ECGS on Backboned Graphs. Consider the following linear program (LP$_{\text{2GS}}$) for a 2-ECGS instance $\mathcal{I}$ on a backboned graph $G = (V, E)$ with edge costs $c(\cdot)$ and base tree T. The variable x_e is an indicator variable for whether tree edge e is present in the solution or not, and y_f is an indicator for whether or not the edges of the base cycle O_f are included. Call a set S "valid" *iff* there exists a group g_i such that $g_i \subseteq S$ and $r \notin S$. Let ∂S to denote the set of edges crossing the cut $S, V \setminus S$.

$$(\text{LP}_{\text{2GS}}) \quad \min \sum_{e \in E(T)} c(e) x_e + \sum_{f \in E \setminus E(T)} c(O_f) y_f$$

$$(3.1) \quad \text{s.t} \sum_{e \in \partial S \cap E(T)} x_e \geq 1 \quad \forall \text{ valid } S \subseteq V$$

$$(3.2) \quad \sum_{f \mid e \in O_f} y_f \geq x_e \quad \forall e \in E(T)$$

$$(3.3) \quad x_e, y_f \in [0, 1] \quad \forall e \in E(T), f \in E \setminus E(T)$$

Though the above LP has exponentially many constraints, it can be solved near optimally in polynomial time as there is an efficient min-cut based separation oracle to verify feasibility. It is also (almost) a relaxation:

LEMMA 3.1. *The cost of an optimal solution LPOpt of the above linear program is at most $4c(\text{Opt})$, where Opt is a minimum cost solution to the 2-ECGS instance $\mathcal{I}$.*

Proof. Let Opt be some optimal solution for the given instance, and let r_i be the representative from group g_i which is 2-edge-connected to the root r. From Observation 2.1, we can construct a *cycle-closed* subgraph Opt′ such that $c(\text{Opt}') \leq 2c(\text{Opt})$ and Opt $\subseteq$ Opt′. Also, since Opt′ is *cycle-closed*, we know (from Observation 2.2) that Opt′ contains the base tree path $P_T(r, r_i)$ for all $i \in [1, q]$. Therefore, for any valid cut S, there is a tree edge in Opt′ crossing it — this means that all constraints (3.1) would be satisfied by the integer solution corresponding to Opt′.

Furthermore, the Covering Lemma ensures that any edge e on the path $P_T(r, r_i)$ has a "covering cycle" $O_f \subseteq$ Opt′ such that $e \in O_f$ — this ensures that constraints (3.2) would also be satisfied. As a result, the solution corresponding to Opt′ is feasible to LP_{2GS}. As for the cost, the LP solution is charged $c(e)$ for any tree edge in e in Opt′ and is charged the cost of the entire cycle O_f corresponding to each non-tree edge in Opt′. Therefore, the value of the objective function for this solution is at most $2c(\text{Opt}') \leq 4c(\text{Opt})$. ∎

3.1.1   Rounding the LP solution. We first give the overview of the rounding procedure and then present the details in two stages.

- Firstly, constraints (3.1) ensure that the x_e variables form a feasible solution to the group Steiner LP on the base tree T, and so we round (in Stage 1) the x_e variables using one iteration of the Garg et al. [21] randomized rounding for the group Steiner problem (which we refer to as the GKR algorithm). At the end of Stage 1, we show our partial solution H_1 would 1-connect roughly $\Omega(1/\log n)$-fraction of the groups to the root.

- In Stage 2, we need to pick covering cycles such that each tree edge in the partial solution H_1 is covered by some cycle. To do this, we essentially use algorithms for Set Cover to get a low-cost collection of cycles covering all tree edges picked in the first stage. This ensures that there are no cut-edges in the subgraph H_1, and therefore the resulting subgraph 2-edge-connects all the groups connected to the root in H_1.

- Finally, we repeat these two stages independently $O(\log^2 n)$ times and output all the edges bought to get a feasible solution 2-connecting all groups to the root r with very high probability.

We now present the details of the two stages, as well as the analysis of the algorithm.

It remains to explain how to obtain the 2-approximate set cover in Step 3 (of Stage 2). The LP

Stage 1 - Picking Base Tree Edges

1: **solve** the linear program LP_{2GST}; **let** (x^*, y^*) denote an optimal solution.
2: **round up** each fractional x_e^* variable to the nearest power of 2. Then, set $x_e^* := 0$ if $x_e^* \leq \frac{1}{2n}$ and scale each non-zero x_e^* to $2x_e^*$.
3: **round** the x^* variables using one round of the GKR rounding scheme.
4: **let** H_1 denote the set of edges bought by the GKR algorithm.

Stage 2 - Picking Covering Cycles

1: **let** $H_2 := \emptyset$.
2: **setup** the following set cover instance:
   2a: **universe:** there is an element for each edge $e \in H_1$.
   2b: **sets:** there is a set S_f of cost $c(O_f)$ for each $f \in E \setminus E_T$,
   2c: **incidence:** element/edge e is covered by a set S_f if $e \in O_f$.
3: **obtain** a set cover $\mathcal{S}$ whose cost is *at most twice* the cost of the LP relaxation.
4: **for** each $S_f \in \mathcal{S}$, include all edges of O_f in H_2.

---

relaxation (LP_{H_1}) of the set cover problem to cover all edges in H_1 is the following (with a variable $y_f \in [0, 1]$ for non-tree edge each f).

$$(\text{LP}_{\text{H}_1}) \qquad \min \sum_{f \in E \setminus E(T)} c(O_f)\, y_f$$

$$(3.4) \qquad \text{s.t} \sum_{f\,|\,e \in O_f} y_f \geq 1 \qquad \forall\, e \in H_1$$

Consider a new instance obtained by replacing each non-tree edge $f = \{u, v\}$ with $\text{lca}(u, v) = a$ by two edges f_l and f_r, the former covering all edges on the path $P_T(u, a)$ and the latter covering edges on $P_T(a, v)$, and making both their costs equal $c(O_f)$. Setting y_{f_l} and y_{f_r} for the LP relaxation to this new instance equal to y_f in the old instance gives a solution of cost at most twice the value of LP_{H_1}. However, the constraint matrix in the LP for this new instance is a network matrix [46, Section 13.3], which is *totally unimodular*, and hence all optimal basic solutions are integral; moreover, any such integral solution is a 2-approximate set cover for the original instance.

3.1.2   Analysis of the LP Rounding. We first show that the subgraph $H_1 \cup H_2$ output by running both stages above 2-connects any fixed group to the root with non-trivial probability.

LEMMA 3.2. (SUCCESS PROBABILITY) *For each group*

g_i, the probability that a vertex from g_i is 2-edge-connected to r in $H_1 \cup H_2$ is $\Omega(1/\log n)$.

Proof. To show this, we observe the following properties of the subgraphs H_1 and H_2.

(i) For each group g_i, the probability that a vertex from g_i is connected to r in H_1 is $\Omega(1/\log n)$.

(ii) For each edge $e \in H_1$, there is an edge $f \in H_2$ such that $e \in O_f \subseteq H_2$.

The first part is a direct consequence of one round of the GKR group Steiner rounding algorithm. The second part follows from the way H_2 was obtained—each element/edge $e \in H_1$ has some set $S_f \in \mathcal{S}$ which covers it, and in Step 4 of Stage 2, we ensure that H_2 contains the entire cycle O_f.

Therefore, consider a group g_i which is connected to r in H_1. Let $\{u_1 = r, u_2\}, \{u_2, u_3\}, \ldots, \{u_{l-1}, u_l = v\}$ denote the tree path $P_T(r,v)$ (which is contained in H_1) from r to some vertex $v \in g_i$. Now, from property (ii) above, we know that u_j is 2-edge-connected to u_{j+1} in $H_1 \cup H_2$ for all $j \in [1, l-1]$. Therefore, from transitivity of edge-connectivity, we see that v is 2-edge-connected to r in $H_1 \cup H_2$. ■

Now we analyze the total expected cost of the subgraph $H_1 \cup H_2$. To this end, consider the optimal solution (x^*, y^*) to the linear program $\mathsf{LP_{2GST}}$. Define $\mathsf{LPOpt^e} = \sum_{e \in E(T)} c(e) x_e^*$ and $\mathsf{LPOpt^f} = \sum_{f \in E \setminus E(T)} c(O_f) y_f^*$. $\mathsf{LPOpt^e}$ and $\mathsf{LPOpt^f}$ denote the tree cost and the non-tree cost of the optimal fractional solution respectively. Let $\mathsf{LPOpt} = \mathsf{LPOpt^e} + \mathsf{LPOpt^f}$ denote the overall cost of the LP relaxation.

LEMMA 3.3. (STAGE 1 COST) *The expected cost of subgraph H_1 is at most $O(1)\mathsf{LPOpt^e}$.*

Proof. Because (x^*, y^*) is scaled by only a constant factor in Step 2 of Stage 1, the proof of this lemma follows directly from the properties of one round of the GKR rounding algorithm [21, Theorem 3.2]. ■

LEMMA 3.4. (STAGE 2 COST) *The expected cost of the subgraph H_2 is at most $O(\log n)\mathsf{LPOpt^f}$.*

Proof. For any fixed outcome of H_1, the cost of the subgraph H_2 is at most the cost of the set cover solution $\mathcal{S}$ in Step 3 of Stage 2 (which, in turn, is at most twice the cost of an optimal LP solution to $\mathsf{LP_{H_1}}$). Therefore, to prove the lemma, it would suffice to exhibit a fractional solution to the linear program $\mathsf{LP_{H_1}}$, whose expected cost is at most $O(\log n)\mathsf{LPOpt^f}$, the expectation being over the first stage randomization.

Consider a cycle O_f with $f = \{u, v\}$, and let a be the least common ancestor of u and v with respect to the
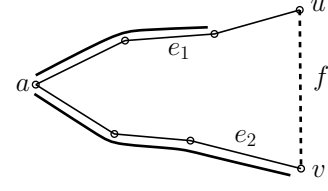

Figure 3.1: e_1 and e_2 are the edges furthest (along base tree) from a chosen in H_1

base tree T rooted at r (see Figure 3.1). If a is neither u nor v, then $P_T(u,v)$ is a disjoint union of subpaths $P_T(u,a)$ and $P_T(a,v)$. In this case, let e_1 and e_2 denote the edges furthest from the root r (along the base tree) on $P_T(a,u)$ and $P_T(a,v)$ that are included in H_1 by the rounding in Stage 1. We then set the value $\widetilde{y_f}$ in the following way:

- if $e_1 \neq \emptyset$, then set $y_f^1 := y_f^*/x_{e_1}^*$; set $y_f^1 := 0$ otherwise.
- if $e_2 \neq \emptyset$, then set $y_f^2 := y_f^*/x_{e_2}^*$; set $y_f^2 := 0$ otherwise.
- set $\widetilde{y_f} := y_f^1 + y_f^2$.

On the other hand, if the lca $a \in \{u, v\}$, then let e denote the edge furthest from the root r (along the base tree) on $P_T(u,v)$ which is included in H_1.

- if $e \neq \emptyset$, then set $\widetilde{y_f} := y_f^*/x_e^*$; set $\widetilde{y_f} := 0$ otherwise.

The fractional solution for the LP is then $\{\widetilde{y_f}, f \in E \setminus E(T)\}$.

CLAIM 3.1. (FEASIBILITY) *The fractional solution $\{\tilde{y_f}\}$ is feasible to $\mathsf{LP_{H_1}}$.*

Proof. Consider some edge $e \in H_1$, and let $f = \{u, v\}$ be any non-tree edge such that $e \in O_f$. Without loss of generality, we assume that the least common ancestor a of u and v is distinct from u and v, and that $e \in P_T(a,u)$ (the proof for other cases is similar).

Now, recall that when we set the value of y_f^1, we considered the edge e_1 furthest from the root on $P_T(a,u)$ that belonged to H_1, and then defined $y_f^1 = y_f^*/x_{e_1}^*$. But since the edge e is contained in $H_1 \cap P_T(a,u)$, this means e_1 is further from r than e along the base tree T (i.e., e_1 is a descendant of e on T). Therefore, we have that $x_{e_1}^* \leq x_e^*$ (from the structure of the group Steiner LP, edges further from the root have smaller x_e values than their ancestors). Consequently, $\widetilde{y_f} \geq y_f^1 \geq y_f^*/x_e^*$ for any edge f such that $e \in O_f$. Now, since (x^*, y^*) is a feasible solution to $(\mathsf{LP_{2GS}})$, constraint (3.2) ensures that $\sum_{f: e \in O_f} y_f^* \geq x_e^*$ for any tree edge e. Therefore, $\sum_{f: e \in O_f} \widetilde{y_f} \geq \sum_{f: e \in O_f} y_f^*/x_e^* \geq 1$ and hence, the values $\{\widetilde{y_f}\}$ comprise a feasible solution to the second stage LP $\mathsf{LP_{H_1}}$. ■

CLAIM 3.2. (EXPECTED COST) *The expected cost $\mathbb{E}[\sum_f c(O_f)\widetilde{y_f}]$ of the fractional solution $\{\widetilde{y_f}\}$ constructed above is $O(\log n)\mathsf{LPOpt}^\mathsf{f}$, where the expectation is taken over the randomization in Stage 1.*

Proof. In the following, let $\mathsf{parent}(e)$ (or $\mathsf{parent}(v)$) denote the *parent edge* of any given edge or vertex with respect to the base tree, i.e. the edge incident on the given edge or vertex that is closest to the root r. Also, for any tree edge e, say that $\mathsf{level}(e) = l$ if $x_e^* = 2^{-l}$ (after the scaling in Step 2 of Stage 1).

Consider any non-tree edge $f = \{u, v\}$, and let a denote the least common ancestor of u and v on the base tree T. We focus on the case where $a \notin \{u, v\}$; the other case when $a \in \{u, v\}$ is similar. Moreover, in order to bound the expected value of $\widetilde{y_f}$, it is sufficient to analyze the expected value of y_f^1; the analysis for y_f^2 is identical.

To this end, let $P_T(a, u) \equiv \{e_1' = \{a, u_1'\}, e_2' = \{u_1', u_2'\}, \ldots, e_t' = \{u_{t-1}', u\}\}$, with the edges ordered such that $\mathsf{parent}(e_j') = e_{j-1}'$ for $2 \leq j \leq t$. Also, let $P' = P_T(a, u) \setminus \{e_j' \mid \mathsf{level}(e_{j+1}') = \mathsf{level}(e_j')\}$: i.e., if an edge further from a along T has the same level as edge e, then e is not included in P'.

Let Z_e denote the event that an edge e is the edge furthest from r on the path $P_T(u, a)$ that was picked in H_1 by the Stage 1 algorithm. Then, we have

$$
\begin{aligned}
\mathrm{E}\left[y_f^1\right] &= \sum_{e \in \{e_1', \ldots, e_t'\}} \mathrm{E}\left[y_f^1 \mid Z_e\right] \Pr\left[Z_e\right] \\
&= \sum_{e \in P'} \mathrm{E}\left[y_f^1 \mid Z_e\right] \Pr\left[Z_e\right] \\
&= \sum_{e \in P'} \frac{y_f^*}{x_e^*} \Pr\left[Z_e\right] \\
&\leq \sum_{e \in P'} \frac{y_f^*}{x_e^*} x_e^* \\
&\leq \log n \cdot y_f^*
\end{aligned}
$$

Here, the second equality follows because if $\mathsf{level}(e_j') = \mathsf{level}(e_{j+1}')$, then whenever e_j' is picked by the GKR rounding, e_{j+1}' would also be selected (this is a property of the GKR algorithm, and this is why we rounded the x_e values in Step 2 of Stage 1). Therefore, the event $Z_{e_j'}$ can never occur (i.e. $\Pr\left[Z_e\right] = 0$ for $e \notin P'$). In the last-but-one inequality, we use $\Pr\left[Z_e\right] \leq x_e^*$ because the event Z_e is dominated by the event that e is picked by the GKR scheme, which happens with probability x_e^*. Finally, the last inequality holds because the edges in P' all belong to distinct levels, and there are at most $\log n$ levels.

We can bound the expected value of y_f^2 using a symmetric argument, and therefore, by linearity of expectation, we have $\mathrm{E}\left[\widetilde{y_f}\right] \leq (2 \log n)y_f^*$. Hence, the total expected cost of the fractional solution is $O(\log n)\mathsf{LPOpt}^\mathsf{f}$. ∎

Now Claims 3.1 and 3.2 show that the expected cost of a fractional solution to $\mathsf{LP_{H_1}}$ is $O(\log n)\mathsf{LPOpt}^\mathsf{f}$. Since we find an integer solution which is a 2-approximation to the LP cost, the proof of Lemma 3.4 is completed. ∎

Lemmas 3.3 and 3.4 together show that the expected cost of the subgraph $H_1 \cup H_2$ is $O(\log n)\mathsf{LPOpt}$, and each group is 2-edge-connected to the root with probability $\Omega(1/\log n)$. Therefore, if we independently repeat this process $O(\log n \log q)$ times, we get 2-edge-connectivity to the root for all groups with high probability, and the expected cost is $O(\log^2 n \log q)c(\mathsf{Opt})$. Thus we get the following theorem.

THEOREM 3.1. (2-ECGS THEOREM) *2-ECGS admits a randomized $O(\log^2 n \log q)$-approximation algorithm on backboned graphs, and hence a randomized $\tilde{O}(\log^3 n \log q)$-approximation on general graphs.*

There are two more natural variants of the 2-ECGS problem:

1. For each group g_i, we want two *distinct* vertices v_{i1} and v_{i2} and edge-disjoint paths P_{i1} and P_{i2} going from the root r to these two chosen vertices

2. For each group g_i, we want two edge-disjoint paths P_{i1} and P_{i2} going from the root r to any two vertices in g_i, *which may or may not be the same.*

In the following section, we show how we can solve the first variant requiring edge-disjoint paths to two distinct vertices, and explain later how we can reduce the latter variant to this setting.

3.2 2-ECGS with Distinct Vertices. Our algorithm is based on rounding an LP relaxation for this problem. The idea behind our LP formulation is the following structural property of any feasible solution: for any group g_i, suppose the representatives that are connected to the root are v_1 and v_2. Then the tree paths from the root r to v_1 and v_2 share a common prefix $P_T(r, v_f)$ till some vertex v_f, and then fork into two disjoint paths $P_T(v_f, v_1)$ and $P_T(v_f, v_2)$, where $v_1, v_2 \in g_i$. Furthermore, it is sufficient to cover the tree edges on the common prefix $P_T(r, v_f)$ to ensure that the group is 2-connected to r.

Now, before we explain the LP formulation, we need to slightly alter the problem instance for some technical issues which we will explain later. Given a backboned instance $\mathcal{I}$, we create a new instance $\mathcal{I}'$

in the following manner: $\mathcal{I}'$ is the same as $\mathcal{I}$ except for the following additional vertices: for each vertex $v \in V$, we add a dummy vertex v' and include an edge $\{v, v'\}$ of 0 cost. The edge $\{v, v'\}$ is also included in the base tree of the modified instance. Now for any group $g_i = \{v_1, v_2, \ldots, v_t\}$ in the original instance $\mathcal{I}$, the corresponding group in $\mathcal{I}'$ comprises of the dummy vertices $\{v_1', v_2', \ldots, v_t'\}$. It is easy to see that there is a bijection between feasible solutions to $\mathcal{I}$ and those to $\mathcal{I}'$, and the cost of the optimal solutions to both instances is the same, since the dummy edges added have 0 cost.

The reason we include these dummy vertices is to prevent an LP solution from finding 2-edge-disjoint paths from a single vertex of a group to the root. Such a solution is not possible in $\mathcal{I}'$ because for any vertex v, its corresponding dummy vertex v' cannot be 2-edge-connected to any other vertex in the new instance.

We are now ready to explain the LP relaxation (presented in Figure 3.2): x_e^d is an indicator variable for whether tree edge e is present on the un-forked portion of the tree path(s) from the root to some group: from the above discussion, only such edges need to be covered by non-tree cycles. The variable x_e^s indicates whether tree edge e is on a forked branch to some group representative. Finally, as used previously, y_f is the indicator variable for whether or not the base cycle O_f is included, and a set S is called "valid" *iff* there exists a group g_i such that $g_i \subseteq S$ and $r \notin S$. Notice that in any feasible solution, the variable x_e^d is 0 for any dummy edge e, since there are no covering cycles which contain it, and thus constraint 3.8 would be violated otherwise.

LEMMA 3.5. *The cost of an optimal solution* LPOpt *of the above linear program is at most* $4c(\mathsf{Opt})$, *where* Opt *is a minimum cost solution to the given instance* $\mathcal{I}'$ *of* 2-ECGS *with distinct vertices.*

Proof. Let Opt be some optimal solution for the given instance, and let v_i^1 and v_i^2 be the representatives from group g_i that have edge-disjoint paths to the root r. From Observation 2.1, we can construct a *cycle-closed* subgraph Opt' such that $c(\mathsf{Opt}') \leq 2c(\mathsf{Opt})$ and $\mathsf{Opt} \subseteq \mathsf{Opt}'$. Also, since Opt' is *cycle-closed*, we know (from Observation 2.2) that Opt' contains the base tree paths $P_T(r, v_i^1)$ and $P_T(r, v_i^2)$ for all $i \in [1, q]$. Let us create an LP solution in the following manner: for any tree edge e, if $e \in P_T(r, v_i^1) \cap P_T(r, v_i^2)$ for some $i \in [1, q]$, then set $x_e^d = 1$. Otherwise, if $e \in P_T(r, v_i^1)$ or $e \in P_T(r, v_i^2)$ for some i, then set $x_e^s = 1$. Set the y_f variables according to whether or not O_f is present in Opt'. Now, consider any valid cut S separating a group g_i from r. If there are 2 tree edges in $\partial S \cap (P_T(r, v_i^1) \cup P_T(r, v_i^2))$, then it is clear that constraint 3.5 is satisfied for this cut S. If only one tree edge e crosses ∂S, then it must be on the common

prefix of $P_T(r, v_i^1) \cap P_T(r, v_i^2)$, which means that x_e^d was set to 1, and therefore constraint 3.5 is satisfied for this cut S in this case as well.

Furthermore, the Covering Lemma ensures that any edge e on the path $P_T(r, v_i^1) \cap P_T(r, v_i^2)$ has a "covering cycle" $O_f \subseteq \mathsf{Opt}'$ such that $e \in O_f$ — otherwise we would have a cut edge separating group g_i from r. This ensures that constraints (3.8) would also be satisfied. Constraint 3.6 trivially holds in our solution since no tree edge e has both x_e^d and x_e^s set to 1, and constraint 3.7 holds because the common prefixes $P_T(r, v_i^1) \cap P_T(r, v_i^2)$ are all tree (sub-)paths anchored from the root, meaning that the set of variables with x_e^d set to 1 form a sub-tree rooted at r and are therefore downward non-increasing. As a result, the LP solution corresponding to Opt' is feasible to $\mathsf{LP_{2GSd}}$, and incurs a charge $c(e)$ for any tree edge in e in Opt' and the cost of the entire cycle O_f corresponding to each non-tree edge in Opt'. Therefore, the value of the objective function for this LP solution is at most $2c(\mathsf{Opt}') \leq 4c(\mathsf{Opt})$. ∎

We now present our rounding algorithm, which is essentially two stages of GKR rounding followed by a set covering phase.

Algorithm 3 2-ECGS with Distinct Vertices

1: **run** $O(\log n \log q)$ independent rounds of the GKR rounding scheme on variables $\widetilde{x_e} = \min(2x_e^d + x_e^s, 1)$. If all groups are not connected to r, then **stop**; else let H_1 be the set of all edges bought in this step.

2: **for** each group g_i, let $v_i \in g_i$ be some vertex connected to r in H_1, and let e_i be the edge closest to r in the tree path $P_T(r, v_i)$ with $x_e^d < 1/4$.

3: **create** group g_i' with those vertices in g_i that do not belong to the subtree subtended by e_i.

4: **run** $O(\log n \log q)$ independent rounds of GKR on variables $\widehat{x_e} = \min(\frac{4}{3}(2x_e^d + x_e^s), 1)$ for the group Steiner instance $\{g_1', g_2', \ldots, g_q'\}$. If all the (new) groups are not connected to r, then **stop**; else let H_2 be the set of all edges bought in this step.

5: **setup** the following set cover instance:
   5a: **universe:** for each edge e such that $x_e^d \geq 1/4$, we have an element.
   5b: **sets:** for each $f \in E \setminus E_T$, we have a set S_f, of cost $c(O_f)$.
   5c: **incidence:** an element e is covered by a set $S_{\{u,v\}}$ if $e \in O_{\{u,v\}}$.

6: **obtain** a set cover $\mathcal{S}$ whose cost is at most twice the cost of the LP relaxation.

7: **for** each $S_f \in \mathcal{S}$, include all edges of O_f in H_3.

8: **return** $H = H_1 \cup H_2 \cup H_3$.

In Step 2 of the above algorithm, we are guaranteed

$$(\mathsf{LP_{2GSd}}) \qquad \text{minimize} \sum_{e \in E(T)} c(e)(x_e^d + x_e^s) + \sum_{f \in E \setminus E(T)} c(O_f)y_f$$

$$(3.5) \qquad \text{subject to } \sum_{e \in \partial S \cap E(T)}(2x_e^d + x_e^s) \geq 2 \qquad \forall \text{ valid } S \subseteq V$$

$$(3.6) \qquad\qquad\qquad\qquad x_e^d + x_e^s \leq 1 \qquad \forall\, e \in E(T)$$

$$(3.7) \qquad\qquad\qquad\qquad x_e^d \leq x_{e'}^d \qquad \forall\, e, e' \in E(T) \text{ s.t } e' \text{ ancestor of } e$$

$$(3.8) \qquad\qquad\qquad \sum_{f \,|\, e \in O_f} y_f \geq x_e^d \qquad \forall\, e \in E(T)$$

$$\qquad\qquad\qquad\qquad x_e, y_f \in [0,1] \qquad \forall\, e \in E(T),\, f \in E \setminus E(T)$$

Figure 3.2: LP Relaxation for 2-ECGS with Distinct Vertices

to find an edge e_i with $x_{e_i}^d < 1/4$ because x_e^d is 0 for all dummy edges, and any path from r to a group vertex ends in a dummy edge. The following lemma is a direct consequence of the GKR rounding scheme.

LEMMA 3.6. *The expected cost of H_1 is at most $O(\log n \log q)$LPOpt. Furthermore, the algorithm stops after Step 1 with a small constant probability.*

LEMMA 3.7. *The variables $\widehat{x_e} = \frac{4}{3}(2x_e^d + x_e^s)$ form a feasible solution for the LP relaxation of the group Steiner instance $\{g_1', g_2', \ldots, g_q'\}$ created in Step 4.*

Proof. Consider any cut S (with $r \notin S$) separating group g_i' from root r. Create a new cut $S' = S \cup T_i$, where T_i is the subtree induced by e_i in T (e_i is as defined in Step 2). Since we include all vertices of T_i in S' and the root r is not contained in $T_i \cup S$, the only additional edge (if at all any) in $\delta S' \cap E(T) \setminus \delta S$ is e_i. Also, because the vertices in $g_i \setminus g_i'$ are all contained in in T_i, the cut S separates the entire group g_i from the root r. Therefore $\sum_{e \in \delta S' \cap E(T)}(2x_e^d + x_e^s) \geq 2$. However, by the definition of e_i, we know that $x_{e_i}^d < \frac{1}{4}$ and as a result, we have $2x_{e_i}^d + x_{e_i}^s < \frac{5}{4}$ (since for any edge e, $x_e^d + x_e^s \leq 1$). Therefore $\sum_{e \in \delta S \cap E(T)}(2x_e^d + x_e^s) \geq \sum_{e \in \delta S' \cap E(T) \setminus \{e_i\}}(2x_e^d + x_e^s) \geq \frac{3}{4}$, and thus $\sum_{e \in \delta S \cap E(T)} \frac{4}{3}(2x_e^d + x_e^s) \geq 1$. The cut constraint in the LP formulation for the group Steiner tree are all satisfied, and this completes the proof. $\blacksquare$

The following lemma is also then a consequence of the GKR scheme.

LEMMA 3.8. *The expected cost of H_2 is at most $O(\log n \log q)$LPOpt. Furthermore, the algorithm stops after Step 4 with a small constant probability.*

LEMMA 3.9. *Let $v_i \in g_i$ be the vertex connecting g_i to the root in H_1, and let $v_i' \in g_i'$ be any vertex connected to the root in H_2. Then, for any edge $e \in P_T(r, v_i) \cap P_T(r, v_i')$, we have $x_e^d \geq \frac{1}{4}$.*

Proof. Let $v_i \in g_i$ be the vertex connecting g_i to the root in H_1 (as chosen in Step 2), and let e_i be the edge closest to r on $P_T(r, v_i)$ which has $x_e^d < 1/4$. Then, by the way we defined our group g_i', any vertex in g_i' is not contained under the subtree beneath e_i. Therefore, the maximal extent to which the path $P_T(r, v_i')$ (for any $v_i' \in g_i'$) can overlap with $P_T(r, v_i)$ is until the parent edge e' of e_i (which has $x_{e'}^d \geq 1/4$ by definition). $\blacksquare$

LEMMA 3.10. *The expected cost of H_3 is $O(1)$LPOpt.*

Proof. The set H_3 is formed by solving a Set Cover relaxation for covering edges whose x_e^d value is at least $1/4$. Therefore, the cost of a feasible solution to the LP relaxation of the associated Set Cover problem is $O(1)$LPOpt. Furthermore, we can make the constraint matrix in the LP totally unimodular, like we did for the 2-ECGS algorithm (Section 3.1.1), which implies that the cost of solution H_3 is $O(1)$LPOpt. $\blacksquare$

From Lemma 3.9, we know that for any group, the x_e^d values on any edge e which belongs to the common tree path until the fork is high (at least $1/4$). But all such edges are covered in H_3 by cycles. Therefore, the subgraph $H_1 \cup H_2 \cup H_3$ is feasible to the given instance, and Lemmas 3.6 ,3.8 and 3.10 bound the expected cost.

THEOREM 3.2. *The above algorithm is a randomized $O(\log n \log q)$-approximation algorithm to 2-ECGS with distinct vertices on backboned graphs, and an $\tilde{O}(\log^2 n \log q)$-approximation algorithm on general graphs.*

Finally, Khandekar et al. ([31], Section 1.2) observe that the variant where the two edge-disjoint paths to any group could be to the same vertex or to distinct vertices, can be reduced to the setting where we force the paths to end up in distinct vertices. Therefore our $\tilde{O}(\log^2 n \log q)$-approximation for the distinct vertices setting carries over to this variant as well.

4 2-Edge-Connected Facility Location

In the standard connected facility location (CFL) problem we are given a set of clients that we assign to some facilities that we open, and then we connect these opened facilities together by a Steiner tree (which can be thought of as the *core* of the network). However, the network designer would ideally like the core to be resilient to edge failures, and hence it is desirable to *two-edge-connect the facilities together*. In this section we give a constant-factor approximation for the 2-edge-connected CFL problem (2-CFL) on backboned networks, and hence an $\widetilde{O}(\log n)$-approximation for the problem on general graphs.

Formally, an instance of 2-CFL is a graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}$, a set of demands $\mathcal{D} \subseteq V$, facility opening costs $f : V \to \mathbb{R}$, and a scaling parameter $M \geq 1$. The goal is to open a set of facilities $F \subseteq V$, assign each demand $u \in \mathcal{D}$ to an open facility $\sigma(u) \in F$, and buy a subgraph H that 2-edge-connects the facilities in F. The cost of the solution is then

$$\sum_{v \in F} f(v) \quad + \quad M \sum_{e \in H} c(e) \quad + \quad \sum_{u \in \mathcal{D}} c(P_{u, \sigma(u)})$$

where $P_{u, \sigma(u)}$ is a shortest path from u to $\sigma(u)$ in G under edge costs $c(\cdot)$. We refer to the three terms in the above sum as the *facility opening cost*, the *Steiner cost* and the *client connection cost* [15].

4.1 2-CFL on Backboned Graphs.
Let G be a backboned graph with base tree T. As a first step towards writing an LP relaxation, we *guess* a facility which an optimal solution Opt opens and call it r. Also, Observation 2.1 says that if H^* is the Steiner subgraph Opt builds to 2-edge-connect the facilities, then there is a *cycle-closed* subgraph $H' \supseteq H^*$ with cost $c(H') \leq 2c(H^*)$; hence we seek to build a Steiner subgraph that is *cycle-closed*. The LP relaxation is then given in Figure 4.3.

The variable x_e is an indicator variable for whether the tree edge e is included in the Steiner subgraph or not, y_f indicates the inclusion of the cycle O_f, z_{uv} indicates if demand u is assigned to the facility at v, and z'_v corresponds to whether a facility is opened at v. Constraints (4.9) and (4.10) are the usual facility location constraints ensuring that clients are (fractionally) connected to some open facility, and (4.11) ensures that the "root" facility r is opened. Constraint (4.12) ensures that open facilities are connected to the root along the base tree: if some client is connected to facilities in $S \subseteq V \setminus \{r\}$, then we need to buy tree edges crossing the cut ∂S (such a tree path exists because we seek *cycle-closed* Steiner subgraphs). Finally, constraints (4.13) ensure that tree edges bought are "covered" by fundamental cycles—note that this is a valid constraint be-

cause of the Covering Lemma 2.1).

LEMMA 4.1. *The cost of an optimal solution* LPOpt *of the linear program* LP$_{\text{2CFL}}$ *is at most* $4c(\text{Opt})$, *where* Opt *is a minimum cost solution to the* 2-CFL *instance* $\mathcal{I}$.

Proof. Let Opt be some optimal solution for the given instance and let demand $u_i \in \mathcal{D}$ be connected to facility v_i. Create an LP solution in the following manner: set $z_{u_i v_i} = 1$, and set z_v to 1 if there is some demand connecting to it. Clearly constraints 4.9-4.11 are then satisfied by this assignment (recall we had guessed r to be one open facility in Opt. Now consider the Steiner subgraph H 2-edge-connecting the facilities in Opt. From Observation 2.1, we can construct a *cycle-closed* subgraph H' such that $c(H') \leq 2c(H)$ and $H \subseteq H'$. Also, since H' is *cycle-closed*, we know (from Observation 2.2) that H' contains the base tree path $P_T(r, v_i)$ for all $i \in [1, |\mathcal{D}|]$. Now set x_e to 1 is $e \in H'$ and y_f to 1 if $O_f \in H'$. Therefore, since for any cut S such that $v_i \in S$ and $r \notin S$, there is a tree edge in H' crossing it, constraint (4.12) is also satisfied by the integer solution corresponding to Opt'.

Finally, the Covering Lemma ensures that any edge e on the path $P_T(r, v_i)$ has a "covering cycle" $O_f \subseteq$ Opt' such that $e \in O_f$ — otherwise e would be a cut edge separating v_i from r. This ensures that constraints (3.2) are also satisfied. As a result, the solution corresponding to Opt' is feasible to LP$_{\text{2CFL}}$. As for the cost, the LP solution is charged $c(e)$ for any tree edge in e in Opt' and is charged the cost of the entire cycle O_f corresponding to each non-tree edge in Opt'. The facility opening and connection costs are identical to that incurred by Opt'. Therefore, the value of the objective function for this solution is at most $2c(\text{Opt}') \leq 4c(\text{Opt})$. ∎

4.1.1 Rounding the LP Solution.
The LP rounding algorithm works in four stages. We first filter the solution to make sure that clients are not fractionally connected to any distant facility. Then we identify disjoint balls that are within reasonable distance to all the clients. In the third stage, we temporarily open a (possibly expensive) facility in each such ball and 2-edge-connect it to the root. Finally, in the fourth phase, we identify cheap facilities in each ball and 2-edge-connect them to the nearby temporary facilities. Here are the details.

Stage I. Filtering: Let (x^*, y^*, z^*) denote an optimal LP solution. Filter on the client connection costs [41] as follows: For $u \in \mathcal{D}$, let $C_u^* := \sum_{v \in V} c(u, v) z_{uv}^*$. Set $z_{uv}^* = 0$ if $c(u, v) > 2C_u^*$, and "double" the resulting solution (x^*, y^*, z^*). That is, set $x_e^* = \min(2x_e^*, 1)$, $y_f^* = \min(2y_f^*, 1)$, $z_{uv}^* = \min(2z_{uv}^*, 1)$, and $z_v'^* = \min(2z_v'^*, 1)$. As usual, this ensures that any client

$$(\text{LP}_{\text{2CFL}}) \qquad \text{minimize } M\left(\sum_{e\in E(T)} c(e)x_e + \sum_{f\in E\setminus E(T)} c(O_f)y_f\right) + \sum_{u\in\mathcal{D}, v\in V} c(u,v)z_{uv} + \sum_{v\in V} f(v)z'_v$$

$$(4.9) \qquad \text{subject to } \sum_{v\in V} z_{uv} \geq 1 \qquad\qquad \forall\, u\in\mathcal{D}$$

$$(4.10) \qquad\qquad z_{uv} \leq z'_v \qquad\qquad \forall\, u\in\mathcal{D},\ v\in V$$

$$(4.11) \qquad\qquad z'_r = 1$$

$$(4.12) \qquad \sum_{e\in\partial S\cap E(T)} x_e \geq \sum_{v\in S} z_{uv} \qquad\qquad \forall\, S\subseteq V\setminus\{r\},\ u\in\mathcal{D}$$

$$(4.13) \qquad \sum_{f\,|\,e\in O_f} y_f \geq x_e \qquad\qquad \forall\, e\in E(T)$$

$$\qquad\qquad x_e, y_f \in [0,1] \qquad\qquad \forall\, e\in E(T),\ f\in E\setminus E(T)$$

Figure 4.3: LP Relaxation for 2-CFL

fractionally connects only to facilities which are within a distance at most twice the fractional connection cost paid by the LP. (To avoid proliferation of notation, we refer to the new solution also as (x^*, y^*, z^*).)

Stage II. Finding disjoint balls: In this step, we identify disjoint balls which are reasonably close to all clients. For each $u\in\mathcal{D}$, let $\mathcal{B}_u = \{v\in V\,|\,c(u,v)\leq 2C_u^*\}$ denote the ball of radius $2C_u^*$. Order the clients $u_1, u_2, \cdots, u_d$ such that $C_{u_1}^* \leq C_{u_2}^* \leq \cdots \leq C_{u_d}^*$, and create a subset $V_D\subseteq\mathcal{D}$ in the following fashion: (i) $u_1\in V_D$, and (ii) for every subsequent i, u_i is included in V_D if $\mathcal{B}_{u_i}\cap\mathcal{B}_{u_j}=\emptyset$ for all $j<i$ s.t $u_j\in V_D$. Now, observe that **(a)** $\mathcal{B}_u\cap\mathcal{B}_{u'}=\emptyset$ for $u, u'\in V_D$, and **(b)** for any client $u\notin V_D$, there exists a client $u'\in V_D$ s.t $c(u,u')\leq 2(C_u^*+C_{u'}^*)\leq 4C_u^*$.

In the next stage, we will *temporarily open* some facilities in each ball $\mathcal{B}_u$ and 2-edge-connect them to the root. However, these facilities may be very expensive compared to what the LP has fractionally opened. We resolve this issue in the final step by actually opening cheap facilities from each ball and 2-edge-connecting them to the temporary facilities. The transitivity of edge-connectivity ensures that the cheap facilities are 2-edge-connected to the root. The crux of the argument is in showing that these two steps can be successfully done without blowing up the cost.

Stage III. Opening *some* facilities: For a set $V'\subseteq V$, let $\mathsf{lca}(V')$ denote their least common ancestor in the base tree with respect to r. Consider the set $S = \{\mathsf{lca}(\mathcal{B}_u)\,|\,u\in V_D\}$ and find the minimum cost subgraph H (with edge costs scaled by M) which 2-edge-connects the vertices in $S\cup\{r\}$. Make H *cycle-closed* by adding all edges of the cycle O_f for any non-tree edge f, and include H in the Steiner subgraph bought.

In the above step, we crucially use the fact that for any ball $\mathcal{B}_u$, the vertex $\mathsf{lca}(\mathcal{B}_u)$ is also contained in $\mathcal{B}_u$.

To see why this is true, consider any vertex $x\in\mathcal{B}_u$. Since all shortest path distances are along the base tree in a backboned graph, we know that all vertices in the path $P_T(u,x)$ are also contained in $\mathcal{B}_u$. Thus for any pair of vertices in $\mathcal{B}_u$, all the vertices in the tree path $P_T(x,y)$ (an in particular, their lca) belongs to $\mathcal{B}_u$.

Stage IV. Opening *cheap* facilities: In this final stage, we identify cheap facilities inside these balls and open those. For each $u\in V_D$, choose a facility v_u in $\mathcal{B}_u$ that minimizes the sum of the facility cost $f(v_u)$ and the cost (with edge-costs scaled by M) of 2-edge-connecting v_u to $\mathsf{lca}(\mathcal{B}_u)$, *assuming the edges in H are already bought*. Open a facility at v_u and include the subgraph H_u that 2-edge-connects v_u to $\mathsf{lca}(\mathcal{B}_u)$ in the Steiner subgraph.

4.2   Analysis of the 2-CFL Rounding. In the following, let the value of the optimal LP solution be $\mathsf{LPOpt} = C^* + O^* + E^* + F^*$, where $C^* = \sum_{u\in\mathcal{D}} C_u^*$ is the fractional connection cost, $O^* = \sum_{v\in V} f(v)z_v'^*$ be the fractional facility opening cost, $E^* = M\cdot\sum_{e\in T} c(e)x_e^*$ denote the fractional tree edge cost, and $F^* = M(\sum_{f\in E\setminus T} c(O_f)y_f^*)$ constitute the fractional cycle cover cost. The following lemmas analyze the cost incurred by the different stages of the algorithm.

LEMMA 4.2. *After Stage I, the modified solution (x^*, y^*, z^*) is feasible to the LP, and the cost of the solution is at most $2\mathsf{LPOpt}$. Further, if $z_{uv}^* > 0$, then $c(u,v)\leq 2C_u^*$.*

Proof. From the definition of C_u^*, it must be that $\sum_{v'\in V\,|\,c(u,v')\leq 2C_u^*} z_{uv}^* \geq \frac{1}{2}$. Therefore, setting z_{uv}^* to 0 when $c(u,v) > 2C_u^*$ and scaling the solution by factor 2 would indeed be feasible to the LP and incur a cost of at most $2\mathsf{LPOpt}$. In fact, the fractional client connection cost, opening cost, and Steiner cost are all at most $2C^*$, $2(E^* + F^*)$, and $2O^*$ respectively. ∎

LEMMA 4.3. *After Stage II, if we ensure that we open a facility in $\mathcal{B}_u$ for each $u \in V_D$, then the total client connection cost is at most $6C^*$.*

Proof. If a client u belongs to V_D, then it must be that some facility is opened in $\mathcal{B}_u$ (in Stages III and IV), which means that the client connection cost for u is at most $2C_u^*$. If $u \notin V_D$, then by the way we constructed V_D, we know that there exists $u' \in V_D$ such that $\mathcal{B}_u \cap \mathcal{B}_{u'} \neq \emptyset$ and $C_{u'}^* \leq C_u^*$. Consequently, the client u can connect to the facility opened in $\mathcal{B}_{u'}$ and the connection cost would be at most $2C_u^* + 2C_{u'}^* + 2C_{u'}^* \leq 6C_u^*$. Therefore, the total client connection cost is at most $\sum_u 6C_u^* = 6C^*$. $\blacksquare$

LEMMA 4.4. *The cost of the subgraph H bought in Stage III is $8(E^* + F^*)$.*

Proof. Consider a client $u \in V_D$. In the feasible LP solution (x^*, y^*, z^*) obtained after the Stage I filtering, u is fractionally connected only to facilities in $\mathcal{B}_u$. Hence (4.12) ensures that $\mathsf{lca}(\mathcal{B}_u)$ can send *unit flow* to the root along the base tree using the x_e variables. Moreover, constraint (4.13) ensures that each tree edge on the path from $\mathsf{lca}(\mathcal{B}_u)$ to r is fractionally covered by fundamental cycles. Hence, for any $u \in V_D$, $\mathsf{lca}(\mathcal{B}_u)$ is fractionally 2-edge-connected to the root r, implying that the LP solution (x^*, y^*) (ignoring the facility opening component) is feasible to the problem of 2-edge-connecting the set of vertices $S = \{\mathsf{lca}(\mathcal{B}_u) \mid u \in V_D\}$ with the root, and the fractional cost is at most $2(E^* + F^*)$, the extra factor of 2 arising from doubling the variables during filtering. Since the flow-based LP formulation for edge-connectivity SNDP has an integrality gap of 2 ([29]), we can use the approximation algorithm of Jain [29] to build the subgraph H that 2-edge-connects $r \cup \{\mathsf{lca}(\mathcal{B}_u) \mid u \in V_D\}$ incurs a cost $4(E^* + F^*)$. Furthermore, the cost of the subgraph H at most doubles when we make it *cycle-closed*. $\blacksquare$

This brings us to the interesting part of the proof: showing that we can open cheap facilities in each ball and 2-connect them to previously opened facilities. For any ball $\mathcal{B}_u$, the LP solution (x^*, y^*, z^*) is fractionally feasible to the problem of opening a facility in $\mathcal{B}_u$ and 2-edge-connecting it to $\mathsf{lca}(\mathcal{B}_u)$. Indeed, u is fractionally connected to facilities in $\mathcal{B}_u$ that can send unit flow (along tree edges) to $\mathsf{lca}(\mathcal{B}_u)$, and the tree edges are fractionally covered to an equal extent by the fundamental cycles. Hence we can send 2 units of flow from the fractionally opened facilities in $\mathcal{B}_u$ to their least common ancestor—and if we chose one of these facilities (say, at random), the cost incurred to 2-connect it to the lca would be at most $O(\mathsf{LPOpt})$. But we cannot do this

analysis independently for all the balls, since that may cost LPOpt for each ball.

To resolve this problem, we now show that the LP solution can be decomposed into disjoint parts corresponding to the balls $\{\mathcal{B}_u, u \in V_D\}$. Consider a ball $\mathcal{B}_u$ for $u \in V_D$, and consider the LP relaxation (given in Figure 4.4) for the problem $\mathcal{P}_u$ of opening a facility in $\mathcal{B}_u$ and 2-edge-connecting it to $\mathsf{lca}(\mathcal{B}_u)$ in the cheapest possible way.

CLAIM 4.1. *There exist feasible solutions (x^{*u}, y^{*u}, z^{*u}) to $\mathsf{LP_u}$ such that $\sum_{u \in V_D} \mathsf{cost}(x^{*u}, y^{*u}, z^{*u}) \leq 2\mathsf{LPOpt}$.*

Proof. Consider some $u \in V_D$. Say that a variable in the solution (x^*, y^*, z^*) is *critical* for $\mathcal{B}_u$ if setting it to 0 would make the resulting solution infeasible for $\mathsf{LP_u}$. The fractional solution (x^{*u}, y^{*u}, z^{*u}) is then formed by taking all fractional variables critical for $\mathcal{B}_u$ (and setting all other variables to 0). Clearly, from definition of criticality, (x^{*u}, y^{*u}, z^{*u}) is feasible to $\mathsf{LP_u}$: we now show that these solutions are (nearly) disjoint. In the following, if a variable x_e or y_f is critical for $\mathcal{B}_u$, we will say that edges e or f are critical for $\mathcal{B}_u$.
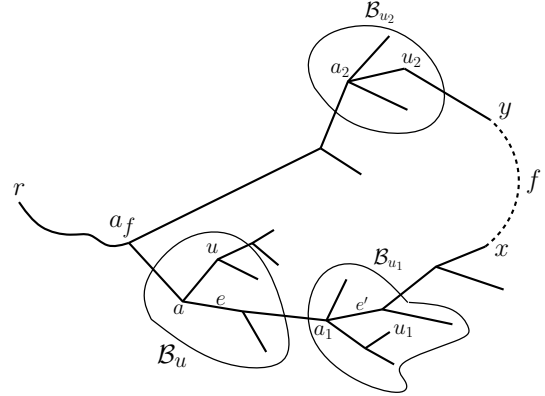

Figure 4.5: Non-tree edge f is critical for at most 2 balls

Since the balls $\mathcal{B}_u$ are all disjoint along the tree, each tree variables x_e can be critical only for the ball containing it. Likewise, the facilities contained in $\mathcal{B}_u$ are not in any other ball, so the z_v variables are also critical for only one ball. Hence, the only shared variables in the LP are the y_f variables. Consider any non-tree edge $f = \{x, y\}$ which is critical for $\mathcal{B}_u$, and let $a_f = \mathsf{lca}(x, y)$. Clearly f contributes towards constraint 4.16 of $\mathsf{LP_u}$ only if $\mathcal{B}_u$ contains edges in $P_T(x, y)$ (see Figure 4.5 for an illustration). Let e be such a tree edge for which constraint 4.16 in $\mathsf{LP_u}$ needs help from f—i.e., f is critical for $\mathcal{B}_u$ because of edge e. Without loss of generality, let $e \in P_T(a_f, x)$.

$$(\mathsf{LP_u}) \qquad \text{minimize } M\left(\sum_{e \in E(T)} c(e)x_e + \sum_{f \in E \setminus E(T)} c(O_f)y_f\right) + \sum_{v \in V} f(v)z_v$$

$$(4.14) \qquad \text{subject to } \sum_{v \in \mathcal{B}_u} z_v \geq 1$$

$$(4.15) \qquad \sum_{e \in \partial S \cap E(T)} x_e \geq \sum_{v \in S} z_v \qquad \forall\, S \subseteq \mathcal{B}_u \setminus \{\mathsf{lca}(\mathcal{B}_u)\}$$

$$(4.16) \qquad \sum_{f \mid e \in O_f} y_f \geq x_e \qquad \forall\, e \in E(T) \setminus E(H)$$

$$x_e, y_f \in [0,1] \qquad \forall\, e \in E(T),\, f \in E \setminus E(T)$$

Figure 4.4: LP Relaxation for subproblem $\mathcal{P}_u$

We now claim that the edge f is critical for $e \in \mathcal{B}_u$ only if there is no other ball $\mathcal{B}_{u_1}$ closer to x than $\mathcal{B}_u$ such that f is also critical for $\mathcal{B}_{u_1}$. Indeed, suppose there were such a ball, as in the figure. The fact that f is critical for $\mathcal{B}_{u_1}$ means that there is an edge $e' \in P_T(a_f, x)$ that is contained in $\mathcal{B}_{u_1}$. Hence, $a_1 = \mathsf{lca}(\mathcal{B}_{u_1})$ is an ancestor of x on the base tree T, and lies on the path $P_T(a_f, x)$. Now, since $\mathcal{B}_{u'} \cap \mathcal{B}_u = \emptyset$, the edge e must be on the path $P_T(a_1, a_f) \subseteq P_T(a_1, r)$. However, since the subgraph H bought in Stage III 2-edge-connects a_1 to r, there must be a cycle $O_{f'}$ bought in H that contains e. Hence the constraint 4.16 would not appear in $\mathsf{LP_u}$ since $e \in H$. This is a contradiction to the fact that f was critical for $\mathcal{B}_u$ because of e. Therefore, any edge f can be critical for at most 2 balls – the ones closest to the end vertices of f. This completes the proof of Claim 4.1. ∎

CLAIM 4.2. *The integrality gap of the LP relaxation* $\mathsf{LP_u}$ *for problem* $\mathcal{P}_u$ *is* $O(1)$.

Proof. Consider a subproblem $\mathcal{P}_u$ and the corresponding solution (x^{*u}, y^{*u}, z^{*u}) to the LP relaxation $\mathsf{LP_u}$. We know that this solution is feasible to the problem of (fractionally) opening a facility and two-edge-connecting it to $\mathsf{lca}(\mathcal{B}_u)$, assuming H is already bought in Stage III.

Now, suppose we simulate the facility opening component at any vertex v by adding a vertex v_f and including a tree edge $\{v, v_f\}$ of cost $f(v)/2$ and a covering non-tree edge $\{v, v_f\}$ of the same cost. It is easy to check that the problem $\mathcal{P}_u$ is identical to that of finding a minimum cost set of edges to augment to H to make some vertex v_f two-connected to $\mathsf{lca}(\mathcal{B}_u)$. Also, the solution (x^{*u}, y^{*u}) is a feasible solution to the new instance (when we set $x^{*u}_{\{v,v_f\}} = y^{*u}_{\{v,v_f\}} = z^{*u}_v$). But now, any fractional solution can be thought of as a 2-flow from $\mathsf{lca}(\mathcal{B}_u)$ to the collection of facility-vertices $\{v_f, v \in \mathcal{B}_u\}$. This can then be decomposed into a linear combination of integral 2-flows from these facility-vertices to $\mathsf{lca}(\mathcal{B}_u)$. Hence by an averaging argument, there exists an integral 2-flow of cost at most $c(x^{*u}, y^{*u}, z^{*u})$. ∎

The above two claims give us a bound the cost of Stage IV (summarized in Lemma 4.5 below), and combining Lemmas 4.2, 4.3, 4.4, and 4.5, we get the following theorem.

LEMMA 4.5. *The total cost incurred by Stage IV in opening a facility* v_u *in each ball* $\mathcal{B}_u$ *($u \in V_D$) and 2-edge-connecting* v_u *to* $\mathsf{lca}(\mathcal{B}_u)$ *is* $O(1)\mathsf{LPOpt}$.

THEOREM 4.1. 2-CFL *admits an* $O(1)$-*approximation algorithm on backboned graphs and a randomized* $O(\log n)$-*approximation algorithm on general graphs.*

5 2-Edge-Connected Buy-at-Bulk

We now consider the 2-connectivity generalization of the buy-at-bulk problem. We are given a (backboned) graph with costs on edges, a demand set $\mathcal{D}$ of s_i-t_i pairs, and a concave function $\Phi : \mathbb{Z} \to \mathbb{R}^+$. The goal is to identify a subgraph H and find 2-edge-disjoint paths from each s_i to t_i in H, such that the cost $c_{\mathsf{bab}}(H) = \sum_{e \in E} c(e) \cdot \Phi(l(e))$ is minimized, where $l(e)$ denotes the load on edge e, i.e., the number of s_i-t_i pairs which use e as part of their 2-flow.

The idea behind our algorithm is simple: if we first 1-connect each demand pair via the tree path, then it would suffice to buy covering cycles (to an appropriate extent to match the load on the tree edges) so that each s_i-t_i pair has 2-edge-disjoint paths between its end points. Therefore, with this in mind, let us for the moment assume that the tree path between each s_i-t_i pair has already been bought, and that we only need to buy the non-tree edges at bulk to *cover* these tree edges. To this end, consider the following problem of choosing non-tree edges (note that the constraints are linear, but the objective function is non-linear):

$$(\mathsf{NLP_{BaB}}) \qquad \min \sum_{f \in E \setminus E(T)} c(f)\Phi\left(\sum_i x^i_f\right)$$

$$(5.17) \qquad \sum_{f \mid e \in O_f} x^i_f \geq 1 \qquad \forall\, e \in P_T(s_i, t_i),\, \forall i$$

$$(5.18) \qquad x_f^i \in [0,1] \quad \forall f \in E \setminus E(T)$$

Lemma 5.1. *The optimal solution of the optimization problem* $\mathsf{NLP}_{\mathsf{BaB}}$ *has cost at most* $c_{\mathsf{bab}}(\mathsf{Opt})$, *where* Opt *is an optimal solution for the given buy-at-bulk instance on a backboned graph.*

Proof. Let us consider the optimal solution Opt, and set x_f^i to 1 whenever a non-tree edge f carries load on behalf of s_i-t_i. Clearly, this definition ensures that $\sum_i x_f^i$ is exactly equal to the total load on any non-tree edge f in Opt. Therefore, the total cost incurred by our solution in $\mathsf{NLP}_{\mathsf{BaB}}$ is at most $c_{\mathsf{bab}}(\mathsf{Opt})$.

To show that this is a feasible solution, suppose one of the constraints (5.17), corresponding to edge e and demand pair s_i-t_i is violated. Now, removing the edge e would separate the base tree T into two components, one containing s_i (which we call C_{s_i}) and the other containing t_i (denoted by C_{t_i}). Since s_i and t_i are 2-edge-connected in Opt and e is the only tree edge crossing C_{s_i} and C_{t_i}, there must exist a non-tree edge $f = \{x,y\} \in E(H) \setminus E(T)$ carrying load for s_i-t_i such that one end vertex of f is in C_{s_i} and the other is in C_{t_i}; otherwise e would be a cut edge separating s_i and t_i in H. Therefore, our solution would have set $x_f^i = 1$, which contradicts the assumption that this was a violated constraint. $\blacksquare$

Lemma 5.2. *Given an integer solution* $\widetilde{x}$ *to the optimization problem* $\mathsf{NLP}_{\mathsf{BaB}}$, *we can find a solution to the buy-at-bulk instance with cost at most* $2 \cdot c(f)\Phi(\sum_i \widetilde{x}_f^i)$, *i.e., twice the objective function.*

Proof. Let us incrementally create a subgraph $\widetilde{H}$ in the following manner: all edges begin with a load of 0. For each non-tree edge f, for each i, if $\widetilde{x}_f^i$ is 1, then increase the load in $\widetilde{H}$ on all edges of the cycle O_f by 1 (all these edges are made to carry load for s_i-t_i).

When the process has been completed, what this ensures is that for each s_i-t_i, for any tree edge $e \in P_T(s_i, t_i)$, there is a cycle O_f containing e which carries load for s_i-t_i. Therefore, by applying transitivity of edge-connectivity, it immediately follows that s_i and t_i are 2-edge-connected within the edges that carry load for the demand pair s_i-t_i.

We now compare the cost $c_{\mathsf{bab}}(\widetilde{H})$ with the cost of solution $\widetilde{x}$. For this, consider the step in the above process when non-tree edge f is being considered. Clearly, as the load on edge f increases from 0 to $\sum_i \widetilde{x}_f^i$, the load on each tree edge e also increases by the same amount $\sum_i \widetilde{x}_f^i$. Therefore, if $l'(\cdot)$ denotes the modified load on the edges (after f is completely processed) and $l(\cdot)$ the original load (before processing f), we have that the increase in cost of network

$\widetilde{H}$ is at most $\sum_{e \in O_f \cap E(T)} c(e)(\Phi(l'(e)) - \Phi(l(e))) + c(f)\Phi(l'(f))$. However, the concavity of the scaling function Φ ensures that $\Phi(l'(e)) - \Phi(l(e)) \leq \Phi(l'(f))$ for any tree edge $e \in O_f$. Therefore, the cost increment is at most $c(f)\Phi(l'(f)) + \sum_{e \in O_f \cap E(T)} c(e)(\Phi(l'(e)) - \Phi(l(e))) \leq c(f)\Phi(l'(f)) + \sum_{e \in O_f \cap E(T)} c(e)\Phi(l'(f)) \leq 2 \cdot c(f)\Phi(\sum_i \widetilde{x}_f^i)$. Therefore, the total cost $c_{\mathsf{bab}}(\widetilde{H})$ is at most twice the cost incurred by $\widetilde{x}$ in $\mathsf{NLP}_{\mathsf{BaB}}$. $\blacksquare$

Finally, it remains to show how we can get an approximately optimal *integral* solution for the problem $\mathsf{NLP}_{\mathsf{BaB}}$, since we can then use Lemma 5.2 above to convert it into a solution for buy-at-bulk. Since the problem has a concave objective function, we first convert it to one with a linear objective function via the reduction given by Meyerson et al. ([42, Section 5.7]). In particular, using their reduction, we lose a constant factor in the objective function, but get $T = |\mathcal{D}|$ copies/cable-types of each edge f with cable type $t \in [1, T]$ having a "fixed cost" of $c(f) \cdot A_t$ and an "incremental cost" of $c(f) \cdot B_t$. Now the following problem of choosing the cables (i.e. setting $X_{f,t}^i$ to 0/1 corresponding to selecting cable type t for edge f) is identical to $\mathsf{NLP}_{\mathsf{BaB}}$:

(i) constraints (5.17) are satisfied, and

(ii) $\sum_f \sum_t c(f)\big(A_{f,t} + B_{t_f} \sum_i X_f^i\big)$ is minimized.

Since this modified problem is now linear, we can write an LP relaxation. In the following, we have a variable $z_{f,t}$ for each edge/cable type which indicates whether we buy cable type t on edge f. Variable $x_{f,t}^i$ denotes whether edge f carries any load for s_i-t_i (using cable type t).

$$(\mathsf{LP}_{\mathsf{BaB}}) \qquad \min \sum_{f,t} c(f)\big(A_t z_{f,t} + B_t \sum_i x_{f,t}^i\big)$$

$$(5.19) \qquad \sum_{t,f \,|\, e \in O_f} x_{f,t}^i \geq 1 \qquad \forall e \in P_T(s_i, t_i), \forall i$$

$$(5.20) \qquad x_{f,t}^i \leq z_{f,t} \quad \forall f, \forall i, \forall t$$

$$(5.21) \qquad x_{f,t}^i, z_{f,t} \in [0,1] \quad \forall f, \forall i, \forall t$$

Finally, notice that this LP resembles that of the standard group Steiner tree problem on a 2-level tree (with the $z_{f,t}$ edges all connected to the root, and the $x_{f,t}^i$ edges hanging off the $z_{f,t}$ edges) with the groups appropriately defined based on each tree edge e needing to be covered for every terminal pair s_i-t_i such that $e \in P_T(s_i, t_i)$. Therefore, if we perform the GKR rounding algorithm, we would get an integer solution (to $\mathsf{LP}_{\mathsf{BaB}}$ and therefore to $\mathsf{NLP}_{\mathsf{BaB}}$) of cost at most a factor $O(\log n)$ of the optimal LP solution. This coupled with Lemmas 5.1 and 5.2 gives us the following theorem.

THEOREM 5.1. *The above algorithm is a randomized $O(\log n)$-approximation algorithm for 2-edge-connected buy-at-bulk on backboned graphs and consequently, an $\tilde{O}(\log^2 n)$-approximation on general graphs.*

6 The k-2EC problem

In the k-2EC problem, the goal is to find a minimum cost set of edges that 2-edge-connects at least k of some given set $X \subseteq V$ of terminals to the designated root vertex; Informally, this is the 2-connectivity variant of the well-studied k-MST problem. In [38], Lau et al. claimed a $O(\log^3 n)$ approximation algorithm for this problem, which was later shown to be incorrect. Subsequently, Lau et al. [39] gave an improved algorithm with approximation ratio $O(\log n \log k)$, and Chekuri and Korula [6] gave the same $O(\log n \log k)$ approximation for the more general 2-vertex-connectivity version, which implies an identical approximation for the k-2EC problem as well. In this section, we point out that applying techniques very similar to those for the 2-ECGS algorithm from Section 3 give us a simple algorithm for k-2EC problem, though with a weaker approximation guarantee of $O(\log^3 n)$.

6.1 The LP Relaxation and its Rounding.
We write an LP similar to the covering Steiner tree problem (there is one *universal* group which contains all the vertices and requires a connectivity of k) [26] along with covering constraints for the tree edges. For the following LP, we create a dummy leaf vertex l_v (corresponding to each vertex v) and connect it to v with an edge $\{v, l_v\}$ of 0 cost. There is also a parallel covering edge $\{v, l_v\}$ of 0 cost (just to make sure there is a feasible solution to 2-edge-connect k of the dummy vertices). The group X then comprises of the set $\{l_v \mid v \in V\}$. In the following, $\mathsf{parent}(v)$ denotes the parent edge of a vertex v along the base tree T, and $T(e)$ denotes all the vertices in the subtree subtended beneath edge e.

$$(\mathsf{LP_{k\text{-}2EC}}) \qquad \min \sum_{e \in E(T)} c(e)x_e + \sum_{f \in E \setminus E(T)} c(O_f)y_f$$

$$(6.22) \quad \text{s.t} \sum_{v \in T(e) \cap X} x_{\mathsf{parent}(v)} \leq k \cdot x_e \qquad \forall e \in E(T)$$

$$(6.23) \qquad x_e \leq x_{\mathsf{parent}(e)} \qquad \forall e \in E(T)$$

$$(6.24) \qquad \sum_{f \mid e \in O_f} y_f \geq x_e \qquad \forall e \in E(T)$$

$$(6.25) \qquad \sum_{v \in X} x_{\mathsf{parent}(v)} \geq k$$

$$x_e, y_f, z_v \in [0, 1]$$

Constraint 6.22 requires that if an edge e is part of the solution, there can be at most k terminals in the subtree $T(e)$ which require connectivity – this is trivially true in integer solutions but is used to cut-off bad fractional solutions (see [35, 26]). Constraint 6.23 re-

quires that the fractional solution be monotonically non-increasing as we move down the tree T. Constraint 6.24 requires that any tree edge included also be covered by a cycle – otherwise it would mean the solution has a cut-edge and is therefore not feasible. Finally, constraint 6.25 simply says that there are at least k terminals which are connected to r. Again, an argument identical to that for Lemma 3.1 shows that an optimal solution to this LP has cost $O(\mathsf{Opt})$.

If we wanted to settle for a $O(\log^2 n \log k)$ approximation algorithm on backboned graphs, we could round this LP exactly like in the 2-ECGS problem, except that instead of $O(\log^2 n)$ rounds of repetition, we repeat the two stages of rounding $O(\log n \log k)$ times—the reason for this change is simple. An application of Janson's inequality tells us that after a single round of Stage 1, at least $\frac{k}{2}$ vertices would be connected to the root by the solution H_1 with probability $\Omega(\frac{1}{\log n})$. (This proof can be found in [35, Section 3.2], where it is shown that the probability that we choose less than $(1 - \delta)k$ vertices is at most $1 - \delta^2\gamma$, with $\gamma = \Theta(1/\log n)$. Setting $\delta = \frac{1}{2}$, we get that we choose at least $k/2$ vertices with probability at least $\gamma/4$.) Therefore, we only need to repeat the two stages of the rounding $\Omega(\log n \log k)$ times to guarantee that we connect at least k terminals with high probability.

However, we can incorporate techniques used in [26] for the covering Steiner tree problem to get rid of a logarithmic factor. Consider the following changes to the rounding algorithm:

Case (1): If at least $k/2$ of the flow is reaching vertices that each receive at least $1/4$ units of flow: Scaling up the fractional solution by a factor of 4 ensures that at least $k/2$ nodes are connected deterministically in the scaled solution. Also, the covering constraints are satisfied completely for each edge bought entirely in the fractional solution — there is a good fractional solution to the set cover problem of covering each tree edge by cycles, which implies that there is an integral solution of at most twice the cost (recall from Section 3.1.1 that such set cover instances have a totally unimodular constraint matrix). Thus we can 2-edge-connect $k/2$ terminals to the root paying at most $O(\mathsf{LPOpt})$. Therefore, since we halve the requirement each time this case holds, there can be at most $O(\log k)$ times this case applies. The total cost of the edges bought whenever we execute this step is $O(\log k)c(\mathsf{Opt})$.

Case (2): Case (1) does not hold, but at least $3k/4$ flow reaches vertices that each receive at least $1/\log n$ units of flow: In this case, it must be that at least $(3k/4 - k/2) = k/4$ units of flow reach vertices that receive flow in the interval $[1/16 \log n, 1/4)$. But this must mean that the number of such vertices is at least

k. So scaling up the solution by $16 \log n$ will connect them all deterministically; and again, the y_f variables are just scaled by $O(\log n)$. Therefore, at a cost of $O(\log n)c(\mathsf{Opt})$, we have 2-edge connected k vertices to the root.

Case (3): Neither of the above cases hold: In this case at least $k/4$ of the flow reaches vertices that receive at most $1/16 \log n$ flow each. In this case, we scale up the flow by $O(\log n)$, and do the GKR randomized rounding. An argument similar to the one in Case (2) of [26] shows that we hit at least k vertices with constant probability. But in this case, the cost of a feasible set cover solution could be as large as $O(\log^2 n)c(\mathsf{Opt})$ — the original solution was scaled by $O(\log n)$, and furthermore, the expected cost of a fractional set cover solution costs $O(\log n)\mathsf{LPOpt}$ like in the 2-ECGS case, because we do a GKR style rounding. Therefore, in this case, we can cover k vertices at a cost of $O(\log^2 n)c(\mathsf{Opt})$.

Thus, the total cost in any case would be at most $O(\log^2 n)c(\mathsf{Opt})$. This gives us an $O(\log^2 n)$ approximation for backboned graphs and an $\tilde{O}(\log^3 n)$ approximation for general graphs.

THEOREM 6.1. *The above algorithm is an $O(\log^2 n)$ approximation algorithm for* k-2EC *on backboned graphs, and an $\tilde{O}(\log^3 n)$ approximation algorithm on general graphs.*

References

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Nearly tight low stretch spanning trees. In *FOCS'08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008.

[2] Spyridon Antonakopoulos, Chandra Chekuri, Bruce Shepherd, and Lisa Zhang. Buy-at-bulk network design with protection. In *FOCS '07: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 634–644, Washington, DC, USA, 2007. IEEE Computer Society.

[3] Nikhil Bansal, Rohit Khandekar, and Viswanath Nagarajan. Additive guarantees for degree bounded directed network design. In *STOC'08: Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 769–778, 2008.

[4] Tanmoy Chakraborty, Julia Chuzhoy, and Sanjeev Khanna. Network design for vertex connectivity. In *STOC'08: Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 167–176, 2008.

[5] Chandra Chekuri, Guy Even, and Guy Kortsarz. A greedy approximation algorithm for the group Steiner problem. *Discrete Appl. Math.*, 154(1):15–34, 2006.

[6] Chandra Chekuri and Nitish Korula. Pruning 2-connected graphs. In *FSTTCS'08: Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2008.

[7] Chandra Chekuri and Nitish Korula. Single-sink network design with vertex connectivity requirements. In *FSTTCS'08: Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, 2008.

[8] Joseph Cheriyan, Tibor Jordán, and R. Ravi. On 2-coverings and 2-packings of laminar families. In *Algorithms—ESA '99 (Prague)*, volume 1643 of *Lecture Notes in Comput. Sci.*, pages 510–520.

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

[10] Joseph Cheriyan and Adrian Vetta. Approximation algorithms for network design with metric costs. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 167–175. ACM, New York, 2005.

[11] Julia Chuzhoy and Sanjeev Khanna. Algorithms for single-source vertex-connectivity. In *FOCS'08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008.

[12] Julia Chuzhoy and Sanjeev Khanna. An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design. In *FOCS'09: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, volume abs/0812.4442, 2009.

[13] Friedrich Eisenbrand and Fabrizio Grandoni. An improved approximation algorithm for virtual private network design. In *16th SODA*, pages 928–932, 2005.

[14] Friedrich Eisenbrand, Fabrizio Grandoni, Gianpaolo Oriolo, and Martin Skutella. New approaches for virtual private network design. In *Automata, languages and programming*, volume 3580 of *LNCS*, pages 1151–1162. Springer, Berlin, 2005.

[15] Friedrich Eisenbrand, Fabrizio Grandoni, Thomas Rothvoß, and Guido Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. In *SODA'08: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1174–1183, 2008.

[16] Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 494–503, New York, NY, USA, 2005. ACM Press.

[17] Jittat Fakcharoenphol and Bundit Laekhanukit. An $O(\log^2 k)$-approximation algorithm for the k-vertex connected spanning subgraph problem. In *STOC'08: Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 153–158, 2008.

[18] Lisa Fleischer. A 2-approximation for minimum cost $\{0, 1, 2\}$ vertex connectivity. In *IPCO*, pages 115–129, 2001.

[19] Lisa Fleischer, Kamal Jain, and David P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *J. Comput. System Sci.*, 72(5):838–867, 2006.

[20] Greg N. Frederickson and Joseph Ja'Ja'. Approximation algorithms for several graph augmentation problems. *SIAM J. Comput.*, 10(2):270–283, 1981.

[21] Naveen Garg, Goran Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000. (Preliminary version in *9th SODA*, pages 253–259, 1998).

[22] Michel X. Goemans, Andrew V. Goldberg, Serge Plotkin, David B. Shmoys, Éva Tardos, and David P. Williamson. Improved approximation algorithms for network design problems. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, 1994)*, pages 223–232, New York, 1994. ACM.

[23] Anupam Gupta, Ravishankar Krishnaswamy, and R. Ravi. Online and stochastic survivable network design. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 685–694, New York, NY, USA, 2009. ACM.

[24] Anupam Gupta, Amit Kumar, Jon Kleinberg, Rajeev Rastogi, and Bülent Yener. Provisioning a Virtual Private Network: A network design problem for multicommodity flow. In *33rd STOC*, pages 389–398, 2001.

[25] Anupam Gupta, Amit Kumar, Martin Pál, and Tim Roughgarden. Approximation via cost sharing: simpler and better approximation algorithms for network design. *J. ACM*, 54(3):Art. 11, 38 pp. (electronic), 2007.

[26] Anupam Gupta and Aravind Srinivasan. An improved approximation ratio for the Covering Steiner problem. *Theory of Computing*, 2:53–64, 2006.

[27] Eran Halperin, Guy Kortsarz, Robert Krauthgamer, Aravind Srinivasan, and Nan Wang. Integrality ratio for group Steiner trees and directed Steiner trees. In *14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 275–284, January 2003.

[28] Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 585–594. ACM Press, 2003.

[29] Kamal Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001. (Preliminary version in *39th FOCS*, pages 448–457, 1998).

[30] David R. Karger and Maria Minkoff. Building Steiner trees with incomplete global knowledge. In *41th FOCS*, pages 613–623, 2000.

[31] Rohit Khandekar, Guy Kortsarz, and Zeev Nutov. The fault-tolerant group steiner problem. In *FSTTCS'09: To Appear*, 2009.

[32] Samir Khuller and Balaji Raghavachari. Improved approximation algorithms for uniform connectivity problems. *J. Algorithms*, 21(2):434–450, 1996.

[33] Samir Khuller and Uzi Vishkin. Biconnectivity approximations and graph carvings. *J. Assoc. Comput. Mach.*, 41(2):214–235, 1994.

[34] Philip N. Klein and R. Ravi. When cycles collapse: A general approximation technique for constrained two-connectivity problems. In *3rd IPCO*, pages 39–56, 1993.

[35] Goran Konjevod, R. Ravi, and Aravind Srinivasan. Approximation algorithms for the covering Steiner problem.

[36] Guy Kortsarz, Robert Krauthgamer, and James R. Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM J. Comput.*, 33(3):704–720 (electronic), 2004.

[37] Guy Kortsarz and Zeev Nutov. Approximating node connectivity problems via set covers. *Algorithmica*, 37(2):75–92, 2003.

[38] Lap Chi Lau, Joseph Naor, Mohammad R. Salavatipour, and Mohit Singh. Survivable network design with degree or order constraints. In *STOC'07: Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 651–660, 2007.

[39] Lap Chi Lau, Joseph Naor, Mohammad R. Salavatipour, and Mohit Singh. Survivable network design with degree or order constraints. *SIAM Journal on Computing*, 39(3):1062–1087, 2009.

[40] Lap Chi Lau and Mohit Singh. Additive approximation for bounded degree survivable network design. In *STOC'08: Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 759–768, 2008.

[41] Jyh-Han Lin and Jeffrey Scott Vitter. e-approximations with minimum packing constraint violation (extended abstract). In *STOC '92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 771–782, New York, NY, USA, 1992. ACM.

[42] Adam Meyerson, Kamesh Munagala, and Serge Plotkin. Cost-distance: Two metric network design. In *41st FOCS*, pages 624–630, 2000.

[43] Zeev Nutov. An almost $O(\log k)$-approximation for k-connected subgraphs. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 912–921. Society for Industrial and Applied Mathematics, 2009.

[44] Zeev Nutov. Approximating minimum cost connectivity problems via uncrossable bifamilies and spider-cover decompositions. In *FOCS'09: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, 2009.

[45] R. Ravi and F. S. Salman. Approximation algorithms for the traveling purchaser problem and its variants in network design. In *Algorithms—ESA '99 (Prague)*, volume 1643 of *Lecture Notes in Comput. Sci.*, pages 29–40. Springer, Berlin, 1999.

[46] Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin,

2003. Paths, flows, matchings, Chapters 1–38.

[47] Chaitanya Swamy and Amit Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.

[48] D. P. Williamson and A. van Zuylen. A simpler and better derandomization of an approximation algorithm for single-source rent-or-buy. *Operations Research Letters*, 2007.

[49] David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15(3):435–454, 1995. (Preliminary version in *25th STOC*, 1993).

A Proofs from Section 2

Proof of Theorem 2.1: Using the construction of [1], we can draw a random spanning tree $T = (V, E(T) \subseteq E)$ of G such that

- $d_T(x, y) \geq d_G(x, y)$ for all $x, y \in V$.
- For any $x, y \in V$, $\mathbb{E}[d_T(x,y)] \leq \widetilde{O}(\log n) \cdot d_G(x,y)$.

where the tree distance d_T is the defined as usual: if $P_T(u, v)$ is the unique u-v path in T, then $d_T(u, v) = \sum_{e \in P_T(u,v)} c(e)$. Now, suppose we consider the same graph G, but with the following edge costs instead: (i) tree edge $e \in T$ has cost $\widehat{c}_T(e) = c(e)$, and (ii) non-tree edge $e \in E \setminus E(T)$ has cost $\widehat{c}_T(e) = \max\{c(e), d_T(u, v)\}$. Then, it is simple to verify that G with edge costs $\widehat{c}_T(.)$ is backboned.

Consider a problem Π, and let the optimal solution to the given instance on G with edge costs $c(.)$ be a subgraph $H \subseteq G$. Then, from the low-stretch property of the random embedding, the expected cost of H under the cost function $\widehat{c}_T$ is $\mathbb{E}_T[\sum_{e \in H} \widehat{c}_T(e)] \leq \widetilde{O}(\log n) \cdot \sum_{e \in H} c(e)$. Therefore the expected cost of any optimal solution under edge costs $\widehat{c}_T$ is at most $\widetilde{O}(\log n) \cdot \sum_{e \in H} c(e)$. Consequently, any β-approximation algorithm for the problem Π on backboned graphs would return a subgraph $H' \subseteq G$, with expected cost (with respect to $\widehat{c}_T$) at most $(\beta \times \widetilde{O}(\log n)) \cdot \sum_{e \in H} c(e)$. Since $\widehat{c}(e) \geq c(e)$ for any edge $e \in E(G)$, the expected cost of the subgraph H' with respect to edge costs $c(.)$ is also at most $(\beta \times \widetilde{O}(\log n)) \cdot \sum_{e \in H} c(e) \leq (\beta \times \widetilde{O}(\log n)) c(H)$. Therefore, the solution H' is a randomized $\beta \times \widetilde{O}(\log n)$-approximate solution on the original edge costs $c(.)$. ∎

B Proofs from Section 4

B.1 2-CFL on Non-Metric Instances We now consider instances where the connection cost for the clients is given by some distance function $d(\cdot, \cdot)$ which may itself not satisfy triangle inequality, and the edge costs for building the 2-connected core is $c(\cdot)$.

We show how we can get poly-logarithmic approximations for the above "non-metric" 2-CFL problem us-

ing essentially the same techniques we used for 2-ECGS. We first guess one facility which the optimal solution opens and call it r. The LP is almost identical to the one given for 2-CFL on general graphs, except for the client-facility connection cost being some arbitrary function $d(\cdot, \cdot)$ instead of the tree distances $c(\cdot)$. Here is a brief overview of the rounding algorithm for 2-CFL. We skip the details of the proofs as they are very similar to the ones given in the earlier sections.

(i) Solve the LP relaxation optimally. Then filter the client connection costs: If we let $D_u^* = \sum_{v \in V} d(u, v) z_{uv}^*$, it must be that $\sum_{v \in V \mid d(u,v) \leq 2D_i^*} z_{uv}^* \geq \frac{1}{2}$. Set $z_{uv}^* \leftarrow 0$ if $d(u, v) > 2D_i^*$ and scale the solution by factor 2.

(ii) For each $u \in \mathcal{D}$, create a group $g_u = \{v \in V \mid z_{uv}^* \neq 0\}$ of facilities associated with this client. It is easy to check that the solution (x^*, y^*) is a feasible solution for the 2-ECGS LP with these groups.

(iii) Perform Stage I and Stage II of the 2-ECGS algorithm once; if a group g_u is 2-connected to the root, open a facility at the representative vertex v_{g_i}. Because the 2-ECGS algorithm ensures that $\Omega(\frac{1}{\log n})$ groups are 2-connected to the root, and we open facilities for these groups, we know that $\Omega(\frac{1}{\log n})$ clients have a facility opened near them. A similar analysis as the one for the 2-ECGS problem can be used to see that the total cost spent in this step is at most $O(\log n)\mathsf{LPOpt}$.

(iv) We can then repeat this process $O(\log^2 n)$ times and output the union of all previous partial solutions to guarantee with high probability a feasible solution to the 2-CFL problem.

THEOREM B.1. *Non-metric* 2-CFL *admits an* $O(\log^3 n)$ *approximation algorithm on backboned graphs, and an* $O(\log^4 n)$ *approximation algorithm for general graphs.*