Cost-Sharing Mechanisms for Network Design *

Anupam Gupta† Aravind Srinivasan‡ Éva Tardos§

Abstract

We consider a single source network design problem from a game-theoretic perspective. Gupta, Kumar and Roughgarden [4] developed a simple method for single source rent-or-buy problem that also yields the best-known approximation ratio for the problem. We show how to use a variant of this method to develop an approximately budget-balanced and group strategyproof cost-sharing method for the problem.

The novelty of our approach stems from our obtaining the cost-sharing methods for the rent-or-buy problem by carefully combining cost-shares for the simpler problem Steiner tree problem; we feel that this idea may have wider implications. Our algorithm is conceptually simpler than the previous such cost-sharing method due to Pál and Tardos[9], and has a much improved approximation factor of 4.6 (over the previously known 15).

1 Introduction

This paper studies the problem of giving good *cost-sharing mechanisms* for a single source network design problem. Imagine a general network design problem, where the participants (or *agents*) want to build a network connecting them to a common source (a server); however, they are autonomous and behave in a selfish (but non-malicious) fashion. Informally, a cost-sharing mechanism builds a network, and allocates the cost incurred among the agents, so that no group of agents is charged too much, thus precluding the possibility of their being unhappy and trying to secede from the system.

The type of problem we consider is where we are given an undirected graph $G = (V, E)$, and a set of *demands* $D \subseteq V$ that want to connect to a common source r. A cost-sharing method is an algorithm that builds a cheap network, and also specifies what portion of its cost is paid by which of the participants in the network; this is done in a manner that ensures that the cost paid by any subset of the participants is fair.

Given an instance of a network design game specified by a set of demands D, a cost-sharing scheme for the problem is simply a function $\xi(D, i)$ with $\xi(D, i) = 0$ for $i \notin D$. For a set D of demands we use $\mathsf{OPT}(D)$ to denote the minimum cost network serving the users in D, while $\mathsf{Alg}(D)$ denotes the cost, or expected cost, of the network computed by a given algorithm. There are three properties that we will be concerned with:

* Research done in part during the IMA workshop on *Network Management and Design* at the University of Minnesota, April 2003.

† Department of Computer Science, Carnegie Mellon University, Pittsburgh PA 15232. Email: anupamg@cs.cmu.edu.

‡ Department of Computer Science and University of Maryland Institute for Advanced Computer Studies, University of Maryland at College Park, College Park, MD 20742. This material is based upon work supported in part by the National Science Foundation under Grant No. 0208005. srin@cs.umd.edu

§ Department of Computer Science, Cornell University, Ithaca, NY 14853. eva@cs.cornell.edu. Research supported in part by ONR grant N00014-98-1-0589, and NSF grants CCR-0311333 and CCR-0325453.

1. **(β-approximate budget-balance)** For a set of demands D, we require that

$$\mathsf{OPT}(D) \geq \sum_{i \in D} \xi(D, i) \geq \mathsf{Alg}(D)/\beta$$

for some given parameter $\beta \geq 1$. Equivalently, by multiplying the shares up by β, we could require that the cost-shares are at least the total cost of a solution found, but do not exceed $\beta\ \mathsf{OPT}(D)$. If $\beta = 1$, we call the cost-sharing budget-balanced.

2. **(fairness)** For any $A \subseteq D$, $\sum_{i \in A} \xi(D, i) \leq \mathsf{OPT}(A)$. I.e., the cost paid by any subset of people should not exceed the optimal cost of connecting them alone and hence they have no incentive to secede.

3. **(cross-monotonicity)** For any $A \subseteq D$, and demand $i \in A$ we require that $\xi(D, i) \leq \xi(A, i)$. I.e., the cost-share of any demand should not go up due to other demands entering the system. This property is also known under the name of population-monotone.

Cross-monotonicity is a key ingredient used in solving the following type of mechanism design problems: consider the network design problem with a set of demand nodes D, with each user (or demand) d having an associated utility u_d. Since the users have limited utilities, the service provider has to now decide which subset of customers it must serve, in addition to designing the network and deciding how to share the cost between the served customers. A mechanism for solving this problem is called *group strategyproof* if no subset of users has an incentive to deviate from the protocol (e.g., by misreporting their utility) in the hope of improving the outcome for themselves (e.g., receiving the service at a cheaper cost). Moulin and Shenker [6] show that having a cross-monotone cost-sharing method for a problem naturally gives rise to a group strategyproof mechanism for the problem in the following way. We start with all the customers; if there is some customer whose cost share (computed w.r.t. the current set of customers) exceeds its utility, we drop it from the set, recompute cost shares and repeat. At the end, we are left with the desired set of customers and their cost-shares.

As an example of cross-monotonicity, let us consider the MINIMUM SPANNING TREE game on the complete graph $G = (V, E)$ with edge weights, given by a set of players D and a *root* $r \notin D$; the objective is to find the cheapest tree $\mathsf{MST}(D)$ spanning D and r. It must be emphasized that this game does not allow the use of Steiner vertices in $\mathsf{MST}(D)$, and r is not a player, and hence should have no cost-share. It is not difficult to verify that a budget-balanced and fair cost-sharing scheme for this game can be found thus: find an MST, root it at r, and set the cost-share for vertex $i \in D$ to be the cost of the edge from i to its parent. However, this scheme is not cross-monotone, and getting a cross-monotone cost-sharing scheme for the minimum spanning tree problem takes more work. Kent and Skorin-Kapov [5] and Jain and Vazirani [7] developed such budget-balanced and cross-monotone cost-sharing schemes for the spanning tree game using a directed branching game; let us denote this cost-sharing scheme by ξ_{MST}. Note that the values $\frac{1}{2}\xi_{MST}$ serve also as cross-monotone, 2-approximately budget-balanced cost-sharing for the corresponding Steiner tree game.

The Single Source Rent-or-Buy Game In this paper we will consider the *Single Source Rent-or-Buy* Network Design game; this combines features of Steiner tree and shortest paths. The game is defined as follows: we are given a complete undirected graph $G = (V, E)$ with edge costs c_e satisfying the triangle inequality, a special *source* (or *root*) vertex r, and a parameter $M \geq 1$. There are also a set of *users* (also called *players* or *demands*) $D \subseteq V$, each of which is identified with a vertex of the graph. We assume that there is a unique player at each vertex. Our results can be easily extended to multiple players, each with a *weight* d_j indicating its amount of demand; we leave the details for the final version of the paper.

The objective is to connect each player j to the source r via some path P_j, on which one unit of bandwidth has been allocated. What makes the game interesting is that aggregating paths is beneficial, in tune with the idea of economies of scale. Hence there are two different actions that can be performed by each edge: either the edge can be *bought* at cost Mc_e, but then an arbitrary amount of bandwidth can be sent on that edge (and hence an arbitrary number of paths P_j can use it); or bandwidth on the edge can be *rented* at cost c_e per unit bandwidth (and hence if the paths for some set S of players were using the edge, then the edge would cost $c_e \times |S|$). As usual, any edge can be used for any of the users. Our main theorem is the following:

Theorem 1.1 *There is a cross-monotone, fair cost-sharing method for the Single Source Rent-or-Buy network design game that is also β-budget-balanced, where $\beta = 4.6$. Furthermore, these cost-shares can be computed in deterministic polynomial time.*

This improves on the results of Pál and Tardos [9], who gave a 15-approximate cross-monotone cost-sharing scheme for the problem. They use a primal-dual algorithm to build the network and obtain cost-shares.

We construct the claimed cost-shares based on expected costs in the approximation algorithm of Gupta et al. [4]. Loosely speaking, that algorithm works by randomly reducing the network design problem to computing Steiner trees over subsets of terminals; we show that if we use the function $\frac{1}{2}\xi_{MST}$ to allocate approximate cost-shares to each vertex in this resulting Steiner game, then the expected cost share of a vertex (taken over the random reduction) gives us the cross-monotone cost-shares claimed in Therorem 1.1. However, since the variances of the random variables involved may be large in general, computing these expected costs in polynomial time is non-trivial. To this end, we give an alternate analysis of the algorithm of [4]; this allows us to give a derandomization using a small sample space, which has some additional useful properties that help ensure cross-monotonicity.

2 The Algorithm and the Cost-Sharing

We will use an algorithm SimpleCFL suggested by Gupta et al. [4] for the equivalent problem of connected facility location (without facility costs). First, let us recall the algorithm from [4]. Here $\alpha > 0$ is a constant that will be chosen later.

S1. *Mark* each demand j independently with probability α/M, and let D' be the set of marked demands.

S2. Construct a minimum spanning tree T on $F = D' \cup \{r\}$. (This is a 2-approximate Steiner tree on F.) The elements of F are called the *open facilities*.

S3. Assign each demand $j \in D$ to its closest demand $i(j)$ in F.

The algorithm suggests a simple and intuitive idea for the cost-shares: each player pays a cost proportional to the *expected* cost incurred by it on running the above algorithm. For a given a set of coin-tosses, the players in D' will pay for buying the MST, where their shares are derived from a cross-monotone cost-sharing scheme ξ_{MST} for the MST problem given in [5, 7]. All other players (players in $D \setminus D'$) will pay for their shortest paths to F. The cost-shares defined for a particular set of coin-tosses exactly cover the cost of the solution built. From [4] we know that this algorithm is a β-approximation algorithm for some constant β. To get a β-approximate cost-shares we divide the above defined shares by a β. Formally:

$$\xi(D, j) = \frac{1}{\beta}\, \mathrm{E}[\, M\, \xi_{MST}(F, j) + \ell(j, F)\,]\,, \tag{2.1}$$

where $\ell(j, S)$ is the length of a shortest-path from j to the closest vertex of the set S, and the expectation is over the coin tosses. (Note that the set $F = D' \cup \{r\}$ depends on the coin tosses.)

Outline of the Proof There are two parts to proving the result: we first have to show that the properties of the cost-shares claimed in Theorem 1.1 are indeed satisfied; i.e., they are β-budget-balanced, fair, and cross-monotone. These properties are proved in Section 3.

The technically involved part of the proof involves showing that our cost-shares can be computed in (deterministic) polynomial time. A little thought will convince the reader that this is not obvious, even with the results of Gupta et al. [4]. Indeed, we need to estimate the expectations in (2.1) for each of the players, but the random variables in (2.1) do not have small variance in general. Furthermore, it is unclear how to derandomize the proof of [4], since it relies on some severe conditioning. To take care of this problem, we give a proof of the performance guarantee of the SimpleCFL algorithm different from the one given by [4]; this is done in Section 4. Our new proof will yield a somewhat worse constant than that of [4], but allow us to derandomize the algorithm by marking the demands in Step (S1) in t-wise independent fashion for a constant t; this appears in Section 5.

Using t-wise independent random choices in Step (S1), allows us to use a polynomial-sized sample space, letting us compute expectations such as those of (2.1) in (deterministic) polynomial time, by considering all the points in the sample space. However, we need to make sure that the properties of the expectation proved in Section 3 (fair and cross-monotone) also hold for the expectation using t-wise independent random choices. Interestingly, the properties of a *particular* construction of t-wise independent random variables turns out to be crucial, as described in the proof of Theorem 3.5.

3 Properties of the Cost-Sharing Scheme

Recall that given an instance of the game specified by a set of demands D, a *cost-sharing scheme* for the game is simply a function $\xi(D, i)$ (with $\xi(D, i) = 0$ for $i \notin D$). We now need to show that the function defined in (2.1) is a cost-sharing scheme with the properties we care about, namely, approximate budget-balance, fairness, and cross-monotonicity. These three properties are, in fact, not independent. A cross-monotone and approximately budget balanced cost-sharing is always fair.

Lemma 3.1 *If a cost-sharing function ξ is approximately budget balanced and cross-monotone then it is also fair.*

Proof: Consider a subset $A \subseteq D$. By cross-monotonicity, it follows that $\sum_{i \in A} \xi(D, i) \leq \sum_{i \in A} \xi(A, i)$. Now using the approximate budget balance property gives us $\sum_{i \in A} \xi(A, i) \leq \mathsf{OPT}(A)$, and completes the proof of the lemma. ∎

We will need the following facts about the cost-sharing scheme ξ_{MST} :

Theorem 3.2 ([5, 7]) *There exists an efficiently computable cost-sharing scheme ξ_{MST} for the Minimum Spanning Tree game, that is budget-balanced, fair and cross-monotone.*

Since the spanning tree connects D **and** r, we will use the notation $\xi_{MST}(D \cup \{r\}, i)$ and $\xi_{MST}(D, i)$ interchangeably; however, we will always ensure that $\xi_{MST}(D \cup \{r\}, r) = 0$.

To prove approximate budget-balance, we will need the following result bounding the performance of the algorithm SimpleCFL; its proof appears in Sections 4 and 5.

Theorem 3.3 *The expected cost of the algorithm* SimpleCFL *on a set of demands D is at most $\beta\,\mathsf{OPT}(D)$ (where $\beta = 4.6$), even if the demands are marked in a t-wise independent fashion in Step (S1), for a suitably large constant t.*

Note that this is a stronger form of the performance guarantee proved in [4] (albeit with a worse constant), since it requires only constant-wise independence of the random variables. Armed with this result, we are now in a position to prove the properties of the function ξ defined in (2.1), and thus to prove Theorem 1.1.

The above two theorems immediately imply that ξ is β-approximately budget-balanced: for any random sample ω, ξ exactly shares the cost of the solution built. See Appendix B for the details.

Lemma 3.4 *The function ξ is a β-approximately budget-balanced.*

Details of the limited-independence marking: In order to prove that the expectation using t-wise independent random marking is cross-monotone, we need a particular choice of random marking. Let $\mathbb{F}$ be a field of size $\geq n$, with $|\mathbb{F}|$ chosen large enough so that $\lceil \alpha|\mathbb{F}|/M \rceil / |\mathbb{F}|$ is sufficiently close to α/M; for convenience, we will assume that this fraction equals α/M. (This approximation impacts our analysis in a negligible manner, as will be shown in the full version.) Let the elements of the field be $\{a_1, a_2, ..., a_{|\mathbb{F}|}\}$, and let the vertices V of the graph be labeled by the first n elements $\{a_1, a_2, \ldots, a_n\}$. Let S be any pre-specified subset of $\mathbb{F}$ with $|S| = \lceil \alpha|\mathbb{F}|/M \rceil$. To get a sample, generate $\omega = (x_0, x_1, \ldots, x_{t-1})$ uniformly at random from $\mathbb{F}^t$, and define Y_i to be 1 if $\sum_{j=0}^{t-1} x_j a_i^j$ lies in S, and $Y_i = 0$ otherwise. By construction, we have $\Pr[\, Y_i = 1 \,] \sim \alpha/M$; furthermore, it is well known that the Y_i generated thus are all t-wise independent. Note that the above distribution is generating n coin tosses Y_i, while we need only $|D|$ of them; however, we can just ignore all the Y_j for $j \notin D$.

For a fixed $\omega \in \mathbb{F}^t$, let F_ω denote the set of facilities; i.e., $F_\omega = \{r\} \cup \{\, j \in D \mid Y_j(\omega) = 1 \,\}$.

Theorem 3.5 *Assuming that the random marking is done using the t-wise independent random variables explained above, or using independent random variables, then the function ξ is cross-monotone, i.e., $\xi(A, i) \leq \xi(D, i)$ for any $A \subseteq D$.*

Proof: The particular type of t-wise independent distribution that we use has the following crucial property: for any choice of $\omega \in \mathbb{F}^t$, if the set of marked demands in the run of SimpleCFL on A is $A'(\omega)$ and the set of marked demands in the run on D is $D'(\omega)$, then $A'(\omega) \subseteq D'(\omega)$. Define the joint probability $p(F, E)$ to be the probability of selecting an element of the sample space ω such that the marked demands in the run of SimpleCFL on A is F and on D it is E. Note that if $p(F, E) > 0$, then $F \subseteq E$. A joint probability with this property can also be defined for the case when the marking is fully independent.

With this additional property, we can prove that the scheme ξ is cross-monotone. To see this, note that

$$\xi(A, i) = \frac{1}{\beta} \sum_{F,E} p(F, E)[M\, \xi_{MST}(F, i) + \ell(i, F)] \geq \frac{1}{\beta} \sum_{F,E} p(E, F)[M\, \xi_{MST}(E, i) + \ell(i, E)] = \xi(D, i),$$

where the inequality uses the fact that the support of $p(F, E)$ is only on pairs with $F \subseteq E$, and the cross-monotonicity of ξ_{MST}. ■

4 A New Analysis of the Performance of SimpleCFL

Suppose first that the marking is done fully independently in Step (S1); we now give a way of analyzing SimpleCFL that is different from the one of [4]. We will then use this proof in Section 5 to show that our estimate of β changes negligible when we conduct a t-wise independent marking fro a high enough value of t. Recall that a solution to the Connected Facility Location Problem with demands D is given by the facilities F and the Steiner tree on them, with the cost being $\sum_{j \in D} \ell(j, i(j)) + Mc(T)$. Of the two terms in the sum, the former cost is referred to as the *connection* cost, and the latter is the *Steiner* cost. Let OPT be an optimal solution with facilities F^* and tree T^*, and let C^* and $S^* = M\,c(T^*)$ be the connection and Steiner costs in OPT; also let $Z^* = C^* + S^*$.

The bound for the Steiner cost C is the same as in [4]: we consider the optimal solution OPT, and compute the expected cost assuming that we construct the Steiner tree in Step (S2) using the paths of the optimal solution. The proof is included in Appendix B for completeness.

Lemma 4.1 *The expected cost of Step (S2) of* SimpleCFL *is at most*

$$S \doteq \mathrm{E}[\,M\,c(T)\,] \leq 2\,(S^* + \alpha C^*). \tag{4.2}$$

Massaging the optimal solution: Before we proceed to bound the connection costs, let us modify OPT to get another solution OPT', which costs more than OPT, but which has some more structure that allows us to complete the argument. The new solution has the following properties:

P1. OPT' opens facilities at $F' \subseteq F^*$, and these are connected by a *cycle* T' instead of a tree T^*; each demand j is now connected to the facility $i'(j)$, which is not necessarily its closest facility.

P2. The number of demands assigned to a facility in F', except perhaps the root r, is a multiple of M. (We will call this the *mod-M* constraint).

Lemma 4.2 *There is a solution* OPT' *with the properties above, which has connection cost $C' \leq C^* + S^*$ and Steiner cost $S' \leq 2S^*$. Hence, the total cost of* OPT' *is $Z' \leq C^* + 3S^*$.*

Proof: Modifying the assignment of demands to nodes in F^* to satisfy the mod-M constraint is fairly simple. Given the tree T^*, we process this in a two-pass fashion. The first pass goes bottom-up, starting at the leaves which must all be in F^*, and making sure that each node satisfies the mod-M property by sending some $b < M$ demand from the node to its parent in the tree. OPT has nodes that are not in F^*, and hence cannot be facilities in F'. We eliminate these nodes during a top-down pass, where we distribute their demand among their children. See Appendix B for the details. ■

Back to Bounding the Connection Cost: From now on, we shall only consider the modified solution OPT'. For simplicity, by making copies of nodes if necessary, we can assume that each node in F' has been assigned M demands. Recall that T' is a cycle which contains r. (See Figure 1.) Starting from r, let us name the facilities on T' in (say) clockwise order $r = f_0, f_1, \dots, f_k, f_{k+1} = r$. (Hence referring to f_l makes sense for all l, since we can just consider l mod $(k+1)$.)

Let D_l be the set of demands assigned to f_l, and hence $|D_l| = M$ for $l \neq 0$ by assumption (P2). Let us abuse notation and add r to D_0. Let P'_l be the portion of the cycle T' joining f_l and f_{l+1}, and let $c(P'_l)$ denote the length of P'_l, and hence $S' = M\,\sum_{l=0}^{k} c(P'_l)$. Let C'_l be the total connection cost of the demands in D_l in the solution OPT' (with $C' = \sum_{l=0}^{k} C'_l$). Our algorithm chooses, in Step (S3), the cheapest assignment of demands to nodes in F. Thus, to bound the expected connection cost, it suffices to bound the expected connection cost of an arbitrary experiment that assigns demands to nodes in F. We present one such experiment next and analyze it. In Appendix A we present and briefly analyze an improved experiment.

A Candidate Assignment: Here we consider the following experiment: if $D_l \cap F \neq \emptyset$, then we assign *all* the demands in D_l to the element in $D_l \cap F$ that is closest to f_l. Else we have to assign these demands to some other open facility (i.e., "go outside D_l"). In this case, consider the smallest t such that $D_{l+t} \cap F \neq \emptyset$; let $s = l + t$. Note that $r \in D_0$, and so $t \leq k$. We now send all the demands in D_l to the facility in D_s closest

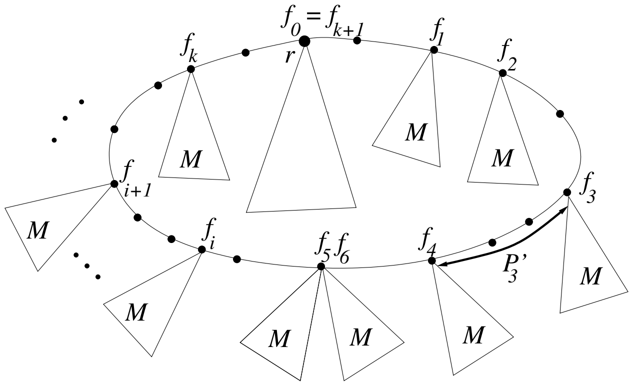

Figure 1: *The transformed instance* OPT'

to f_s. If we assign demands in D_l to a marked node in D_{l+t}, then a path for each demand in D_l goes through $f_l, P'_l, P'_{l+1}, \ldots, P'_{l+t-1}$, and then from f_{l+t} to the element in $D_{l+t} \cap F$ closest to it. We will bound the expected cost of these paths, which in turn will bound the expected connection cost of SimpleCFL.

Let us define some random variables. Let X_i be the indicator variable that $D_i \cap F = \emptyset$ in our algorithm. (Note that $X_0 = 0$ with probability 1.) Let A_i be the distance from f_i and the closest element of $D_i \cap F$. (If this is empty, A_i can be set to 0.) By the above arguments, the assignment cost of the M demands in D_l is at most

$$C'_l + M \sum_{i=l}^{k} (X_l \cdots X_i)\, c(P'_i) + M \sum_{i=l}^{k} (X_l \cdots X_{i-1})\,(1 - X_i)\, A_i \tag{4.3}$$

Indeed, the first term is the distance traveled by the demands in D_l to reach f_l, the second term expresses the fact that demands use P_i (and pay $M\, c(P'_i)$) if and only if $D_l, \ldots, D_i$ all do not intersect F, and the third term implies that we incur a cost of $M\, A_i$ if we assign demands in D_l to the closest member of $D_i \cap F$.

Note that X_i and X_j are independent for $i \neq j$, and that $\mathrm{E}[\, X_i \,] = (1 - \alpha/M)^M = \mathbf{q}$. To bound $M \cdot \mathrm{E}[\,(1 - X_i)A_i\,]$, let the distances from f_i to the members of D_i be $a_1 \leq a_2 \leq \ldots \leq a_M$ with $\sum_j a_j = C'_i$. Now,

$$\mathrm{E}[\,(1 - X_i)A_i\,] = \sum_{j=1}^{M} a_j \times \alpha/M \times (1 - \alpha/M)^{j-1}.$$

Note that the coefficients of a_j decrease as j increases, and hence subject to the constraints above, the expectation is maximized when all the a_j's are equal to C'_i/M. This yields

$$\mathrm{E}[\,(1 - X_i)A_i\,] \leq (C'_i/M)\,[1 - (1 - \alpha/M)^M] = (C'_i/M)\,(1 - \mathbf{q}). \tag{4.4}$$

Let C_l be the expected connection cost of the demands in D_l in our current experiment. Combining the above-seen facts with the inequality (4.3), we get that for $l > 0$,

$$C_l \leq C'_l + M \sum_{i=l}^{k} c(P'_i)\, \mathbf{q}^{i-l+1} + \sum_{i=l}^{k} C'_i\,(1 - \mathbf{q})\, \mathbf{q}^{i-l}. \tag{4.5}$$

Note that $C_0 = C'_0$; adding this to the sum of (4.5) over all l, the total expected connection cost is

$$C \leq \sum_{l=0}^{k} C'_l + M \sum_{i=1}^{k} c(P'_i) \sum_{l=1}^{i} \mathbf{q}^{i-l+1} + \sum_{i=1}^{k} C'_i \, (1-\mathbf{q}) \sum_{l=1}^{i} \mathbf{q}^{i-l} \tag{4.6}$$

$$\leq C' + M \sum_{i=1}^{k} c(P'_i) \frac{\mathbf{q}}{1-\mathbf{q}} + \sum_{i=1}^{k} C'_i \, (1-\mathbf{q}) \frac{1}{1-\mathbf{q}}$$

$$\leq 2\, C' + \frac{\mathbf{q}}{1-\mathbf{q}}\, S' \tag{4.7}$$

However, $\mathbf{q} \leq 1/e^{\alpha}$, and hence the second term is at most $1/(e^{\alpha}-1)$. Now using Lemma 4.2 to replace C' by $C^* + S^*$ and S' by $2S^*$, we see that the expected connection cost of our algorithm can be bounded by:

$$C \leq 2\, C^* + \frac{2\, e^{\alpha}}{e^{\alpha}-1}\, S^*. \tag{4.8}$$

Adding with (4.2), the total expected cost is at most $C + S \leq 2(1+\alpha)C^* + (2 + \frac{2e^{\alpha}}{e^{\alpha}-1}) \cdot S^*$. Since the optimal cost is $Z^* = C^* + S^*$, we choose $\alpha \approx 1.35$ to minimize the approximation ratio, and get the following result.

Theorem 4.3 *The algorithm* SimpleCFL *is a* $\beta = 4.7$*-approximation algorithm for* CFL.

We present an improved bound of a 4.6 approximation in Appendix A. If we were to use an 1.55-approximation for Steiner tree [10] to buy edges in Step (S2), we would be getting an improved approximation ratio of 4.2 for CFL, while the analysis of [4] gets a 3.55 approximation. However, we need to use the 2-approximation algorithm of the MST, as there is a cross-monotone cost-sharing function ξ_{MST} for the MST problem.

5 Analysis of the Limited-Independence Marking

Let ϵ be an arbitrary positive constant lying in $(0, 1)$. We now prove that if the demands are marked in t-wise independent fashion where $t = a \log(1/\epsilon)$ for a suitably large constant a, then the expected approximation ratio is at most $(1+\epsilon)$ times what it is with independent marking. For simplicity, we do not consider the algorithm of Section 6; we only consider the version of Section 4 where all demands in D_i get routed to the closest $j \geq i$. The claims here hold for the algorithm of Appendix A also, as will be shown in the full version of this paper. (In the latter algorithm, we randomly break ties if there are two D_i equidistant from a particular group D_l, that can be used to accommodate the demands in D_l. We will replace this by the following. Choose just one global bit b at random. If $b = 0$, then every D_l that needs to break a tie, chooses the D_i of lower index; else if $b = 1$, then every D_l that needs to break a tie, chooses the D_i of higher index. This only multiplies the size of our small sample space $\mathbb{F}^t$ by a factor of 2.)

Recall that the total connection cost is a random variable that is the sum of three quantities: (i) the deterministic value $\sum_l C'_l$, which represents the cost of all demands in each D_l first getting routed to f_l; (ii) the value that corresponds to unsuccessfully traveling through some D_l, and (iii) the total cost paid in traveling from f_i to the closest marked demand in D_i, once D_i is identified as the closest cluster. We will only show that the expected total cost of (iii) gets multiplied by at most $(1+\epsilon)$ due to our $t = a\log(1/\epsilon)$-wise independent marking; the argument for the term (ii) is analogous, and, in fact, simpler. Specifically, we will show the following. Let i be the index of some arbitrary but fixed D_i. We show that the expected value of the random variable

$$\phi \doteq \sum_{j=1}^{i} M X_j X_{j+1} \cdots X_{i-1} \cdot (1 - X_i) A_i$$

gets multiplied by at most $(1+\epsilon)$. (Since i is fixed, we have not subscripted ϕ as ϕ_i; this remark also holds for many other definitions below.) We next present some notation.

- For each group D_j, let $Z_{j,1}, Z_{j,2}, \ldots, Z_{j,M}$ be the respective indicator random variables for the demands in D_j getting marked by our algorithm, when these demands are considered in *nondecreasing order of distance from* f_j. (This ordering will be important only in the case where $j=i$.)
- Let A be any set of ordered pairs $\{(j,k)\}$. Then, $N(A)$ is the indicator random variable for the event "for all $(j,k) \in A$, $Z_{j,k}=0$". Also, $S(A)$ denotes $\sum_{(j,k)\in A} Z_{j,k}$. ("$N$" stands for none, and "$S$" for sum.)

The following lemma distills some results known from [1, 2, 11], to give two types of upper bounds on $\mathrm{E}\left[N(A)\right]$.

Lemma 5.1 *Let t_1 and t_2 be any* ***even*** *positive integers such that $t_1, t_2 \le t$ (recall that the marking is done in a t-wise independent manner). The following hold for any set A of ordered pairs $\{(j,k)\}$.*

(i) *Let $IE(s,A)$ be the random variable denoting the inclusion-exclusion expansion of $N(A)$ truncated at the sth level; i.e.,*

$$IE(s,A) = \sum_{r=0}^{s}(-1)^r \sum_{A'\subseteq A:\ |A'|=r}\ \prod_{(j,k)\in A'} Z_{j,k}.$$

Then, the inequality $N(A) \le IE(t_1,A)$ holds always; also,

$$\mathrm{E}\left[IE(t_1,A)\right] \le (1-1/M)^{|A|} + \left(\frac{e|A|}{Mt_1}\right)^{t_1}. \tag{5.9}$$

(That is, $\mathrm{E}\left[IE(t_1,A)\right]$ is at most $\left(\frac{e|A|}{Mt_1}\right)^{t_1}$ more than what it would be if the marking was done completely independently.)

(ii) *Let $NCM(t_2,A)$ denote the "normalized central moment" $\frac{(S(A)-|A|/M)^{t_2}}{(|A|/M)^{t_2}}$. Then, the inequality $N(A) \le NCM(t_2,A)$ holds always; also,*

$$\text{if } t_2 \le |A|/M, \text{ then } \mathrm{E}\left[NCM(t_2,A)\right] \le 8\cdot(2t_2)^{t_2/2}\cdot(|A|/M)^{-t_2/2}. \tag{5.10}$$

Proof: (Sketch) The upper bounds that are claimed to hold always on $N(A)$ in (i) and (ii), easily follow from the fact that t_1 and t_2 are even. Bound (5.9) follows from [2]; bound (5.10) follows from [1] (see also [11] for similar bounds). ■

We now show how to use Lemma 5.1 to upper-bound $\mathrm{E}\left[\phi\right]$. Let $a_1, a_2, \ldots, a_M$ be the distances of the demands in D_i from f_i, written in *nondecreasing* order. Then, expanding the "$(1-X_i)A_i$" part of ϕ, we see that

$$\phi = \sum_{j=1}^{i} M X_j X_{j+1}\cdots X_{i-1}\cdot\left[\sum_{k=1}^{M} a_k Z_{i,k}\cdot\prod_{\ell=1}^{k-1}(1-Z_{i,\ell})\right].$$

Fix k arbitrarily, and let $z_k \doteq Z_{i,k}\cdot\prod_{\ell=1}^{k-1}(1-Z_{i,\ell})]\cdot\sum_{j=1}^{i} X_j X_{j+1}\cdots X_{i-1}$. We aim to show that $\mathrm{E}\left[z_k\right]$ is multiplied by at most $(1+\epsilon)$ in our t-wise independent marking, as compared to the fully-independent marking. For $j=0,1,\ldots,i-1$, define $A_j = \{(r,s) :\ (i-j)\le r\le (i-1),\ 1\le s\le M\}$. Thus we get

$$z_k = Z_{i,k}\cdot N(\{(i,\ell):\ 1\le \ell\le k-1\})\cdot\sum_{j=0}^{i-1} N(A_j).$$

Now let a_1 be a sufficiently large constant. Define a_2 to be the smallest even integer that is at least $2e\log(1/\epsilon)$, t_1 to be the smallest even integer that is at least $2ea_1\log(1/\epsilon)$, and t_2 to be the largest even integer that is at most $a_1\log(1/\epsilon)/4$. Rewrite z_k as

$$Z_{i,k} \cdot N(\{(i,\ell) : 1 \le \ell \le k-1\}) \cdot \sum_{j \le a_1 \log(1/\epsilon)} N(A_j) + Z_{i,k} \cdot N(\{(i,\ell) : 1 \le \ell \le k-1\}) \cdot \sum_{j > a_1 \log(1/\epsilon)} N(A_j).$$

Thus, by Lemma 5.1, z_k is always bounded by the sum of the following two random variables:

$$Z_{i,k} \cdot IE(a_2, \{(i,\ell) : 1 \le \ell \le k-1\}) \cdot \sum_{j \le a_1 \log(1/\epsilon)} IE(t_1, A_j), \text{ and}$$

$$Z_{i,k} \cdot IE(a_2, \{(i,\ell) : 1 \le \ell \le k-1\}) \cdot \sum_{j > a_1 \log(1/\epsilon)} NCM(t_2, A_j).$$

If we expand the expectations of these two random variables using the linearity of expectation, we get terms each of which is a product of some of the random variables $Z_{\cdot,\cdot}$; importantly, the number of factors in each such term is at most $1 + a_2 + t_1$ and $1 + a_2 + t_2$ respectively. Thus, if we choose $t = 1 + a_2 + t_1$ (recall that $t_1 \ge t_2$), then the expectations of these two random variables become

$$\mathrm{E}\left[Z_{i,k}\right] \cdot \mathrm{E}\left[IE(a_2, \{(i,\ell) : 1 \le \ell \le k-1\})\right] \cdot \sum_{0 \le j \le a_1 \log(1/\epsilon)} \mathrm{E}\left[IE(t_1, A_j)\right] \text{ and} \tag{5.11}$$

$$\mathrm{E}\left[Z_{i,k}\right] \cdot \mathrm{E}\left[IE(a_2, \{(i,\ell) : 1 \le \ell \le k-1\})\right] \cdot \sum_{j > a_1 \log(1/\epsilon)} \mathrm{E}\left[NCM(t_2, A_j)\right] \tag{5.12}$$

respectively. We next use (5.9) and (5.10) to bound these values; we will see that choosing the constant a_1 large enough results in $\mathrm{E}[z_k]$ being at most $(1+\epsilon)$ times what it is with independent marking.

The expression (5.11) is at most

$$\frac{1}{M} \cdot \left((1 - 1/M)^{k-1} + \left(\frac{e(k-1)}{Ma_2} \right)^{a_2} \right) \cdot \sum_{0 \le j \le a_1 \log(1/\epsilon)} \left((1 - 1/M)^{Mj} + (ej/t_1)^{t_1} \right);$$

i.e., at most $\frac{1}{M} \cdot ((1-1/M)^{k-1} + (e/a_2)^{a_2}) \cdot \sum_{0 \le j \le a_1 \log(1/\epsilon)} \left((1 - 1/M)^{Mj} + (ea_1 \log(1/\epsilon)/t_1)^{t_1} \right)$. Similarly, the expression (5.12) is at most $\frac{1}{M} \cdot ((1 - 1/M)^{k-1} + (e/a_2)^{a_2}) \cdot \sum_{j > a_1 \log(1/\epsilon)} (8 \cdot (2t_2/j)^{t_2/2})$. Thus, if the demands are marked in t-wise independent fashion, then $\mathrm{E}[z_k]$ is at most $\frac{1}{M} \cdot ((1 - 1/M)^{k-1} + (e/a_2)^{a_2})$ times

$$\sum_{0 \le j \le a_1 \log(1/\epsilon)} \left((1 - 1/M)^{Mj} + (ea_1 \log(1/\epsilon)/t_1)^{t_1} \right) + \sum_{j > a_1 \log(1/\epsilon)} (8 \cdot (2t_2/j)^{t_2/2}).$$

On the other hand, under fully-independent marking, $\mathrm{E}[z_k] = \frac{1}{M} \cdot (1 - 1/M)^{k-1} \cdot \sum_{j \ge 0} (1 - 1/M)^{Mj}$. Recalling the definitions of a_1, a_2, t_1 and t_2, it is easy to verify that if a_1 is chosen as a sufficiently large constant, then the former value is at most $(1 + \epsilon)$ times the latter.

6 Further Extensions

For the model of connected facility location where there are facility opening costs as well, we can get a simple algorithm costing at most

$$\rho_{FL}(2+\rho_{ST})\,F^* + (1+\rho_{FL})(2+\rho_{ST})\,C^* + (\rho_{ST} + 2e/(e-1))\,S^*,$$

by simply marking demands with probability $1/M$, running a ρ_{FL}-approximate facility location algorithm, connecting the facilities built using a ρ_{ST}-approximate Steiner tree algorithm, and connecting each demand to its nearest open facility. (The proof is very similar to that of Theorem 2.) Using $\rho_{FL} = 1.52$ and $\rho_{ST} = 1.55$, we get a 9.01-approximate algorithm. (A better performance can be obtained by choosing α carefully, and by a more careful analysis.) However, if we use the $\rho_{FL} = 3$ approximate Facility Location algorithm of Pál and Tardos [9] and its accompanying cross-monotone cost sharing scheme, the MST heuristic with $\rho_{ST} = 2$ for Steiner tree (and use its cross-monotone cost-sharing scheme as well), and make each demand pay

$$\xi(D,j) = \frac{1}{\beta'}\,\mathrm{E}[\,M\,\xi_{MST}(D'\cup r, j) + M\,\xi_{FL}(D', j) + \ell(j, D'\cup r)\,]\,, \tag{6.13}$$

for a suitably chosen $\beta' = O(1)$, we can get a β'-approximate cross-monotone cost sharing scheme for CFL with facility costs as well. Further details of this extension will be explained in the full version of the paper.

References

[1] Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *Proc. IEEE Symposium on Foundations of Computer Science*, 276–287, 1994.

[2] Guy Even, Oded Goldreich, Michael Luby, Noam Nisan, and Boban Veličković. Approximations of general independent distributions. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 10–16, 1992.

[3] Anupam Gupta, Amit Kumar, Jon Kleinberg, Rajeev Rastogi, and Bülent Yener. Provisioning a Virtual Private Network: A network design problem for multicommodity flow. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 389–398, 2001.

[4] Anupam Gupta, Amit Kumar, and Tim Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 365–372, 2003.

[5] Kathryn J. Kent and Darko Skorin-Kapov. Population monotonic cost allocations on MSTs. In *Proceedings of the 6th International Conference on Operational Research (Rovinj, 1996)*, pages 43–48. Croatian Oper. Res. Soc., Zagreb, 1996.

[6] H. Moulin, S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory* 18, pp. 511-533, 2001.

[7] Kamal Jain and Vijay Vazirani. Applications of approximation algorithms to cooperative games. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 364–372, 2001.

[8] David R. Karger and Maria Minkoff. Building Steiner trees with incomplete global knowledge. In *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science*, pages 613–623, 2000.

[9] Martin Pál and Éva Tardos. Group Strategyproof Mechanisms via Primal-Dual Algorithms. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003, pp. 584-593.

[10] Gabriel Robins and Alexander Zelikovsky. Improved Steiner tree approximation in graphs. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2000, pp. 770-779.

[11] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. *Chernoff-Hoeffding bounds for applications with limited independence*, SIAM J. Discrete Math., 8 (1995), pp. 223–250.

[12] H. P. Young. Cost allocation. In R. J. Aumann and S. Hart, editors, *Handbook of Game Theory*, volume 2, chapter 34, pages 1193–1235. North-Holland, 1994.

Appendix A: A More Refined Analysis of the Algorithm

Let us modify the experiment of Section 4 to assign the demands to facilities in F: when $D_l \cap F = \emptyset$ and we go outside D_l, instead of always going around the cycle in a clockwise order, consider the smallest t such that either D_{l+t} or D_{l-t} has an non-empty intersection with F; let D_s be the corresponding set with $D_s \cap F \neq \emptyset$. (If both D_{l+t} and D_{l-t} intersect F, we set D_s to be a randomly chosen one of these.) Note that $r \in D_0$, and so $t \leq \lceil k/2 \rceil$. We now send all the demands in D_l to the facility in D_s closest to f_s.

Let the random variables X_i and A_i be as in Section 4. We will not go over the analysis of this experiment in the same level of detail, but let us point out some of the salient ideas: for some facility in D_i to be used by $\geq k$ groups of larger index (say), we must have: (i) $X_i = 0$, and (ii) either all $2k$ groups $D_{i+1}, \ldots, D_{i+2k}$ do not intersect F, or only the last one does and we choose D_i with probability half. Like in (4.6), the expected total cost will again be a non-negative linear combination of the values $\{C'_i\}$ and $\{c(P'_i)\}$; using the idea outlined in this paragraph and (4.4), we can verify that the coefficient of C'_i (in this linear combination) will be

$$1 + (1 - \mathbf{q}) + 2(1 - \mathbf{q}) \sum_{k \geq 1} (\mathbf{q}^{2k} + \frac{1}{2}\mathbf{q}^{2k-1}(1 - \mathbf{q})) \leq 1 + (1 - \mathbf{q}) + \mathbf{q} = 2. \tag{6.14}$$

Analyzing the coefficient of $c(P'_i)$ needs more work. Assume without loss of generality that $\lfloor k/2 \rfloor \leq i \leq k$; the complementary case is similar. Given a set T of indices, let $U(T)$ denote the event that $X_j = 1$ for all $j \in T$; informally, the "U" denotes "unmarked". We consider when the demands in group D_l will use the path P'_i; we will do so by considering the cases $l \leq i$ and $l > i$.

Case I: $l \leq i$. Let $\mathcal{A}_l$ be the event that the demands in group D_l use the path P'_i. For $\mathcal{A}_l$ to hold, the following event $\mathcal{B}_l$ should also happen: $\mathcal{B}_l \equiv U(\{j : |j - l| \leq i - l\})$. So,

$$\Pr[\mathcal{A}_l] = \Pr[\mathcal{B}_l] \times \Pr\left[\mathcal{A}_l \mid \mathcal{B}_l\right] = \mathbf{q}^{2(i-l)+1} \times \Pr\left[\mathcal{A}_l \mid \mathcal{B}_l\right].$$

Now, if $k + 1 - l \geq l$, it is easy to see by symmetry that $\Pr\left[\mathcal{A}_l \mid \mathcal{B}_l\right] \leq 1/2$; thus we have

$$\text{if } k + 1 - l \geq l, \text{ then } \Pr[\mathcal{A}_l] \leq \mathbf{q}^{2(i-l)+1}/2. \tag{6.15}$$

Next, suppose $k + 1 < 2l$. Define an event $\mathcal{C}_l$: $\mathcal{C}_l \equiv U(\{j : i + 1 - l \leq |j - l| \leq k - l\})$. We have $\Pr\left[\mathcal{A}_l \mid (\mathcal{B}_l \wedge \overline{\mathcal{C}_l})\right] = 1/2$, by symmetry. Also, a moment's thought reveals that

$$\Pr\left[\mathcal{A}_l \mid (\mathcal{B}_l \wedge \mathcal{C}_l)\right] = 1 - (1 - \mathbf{q})/2 = (1 + \mathbf{q})/2.$$

Thus we have

$$\begin{aligned}
\Pr[\mathcal{A}_l] &= \mathbf{q}^{2(i-l)+1} \cdot (\Pr\left[\overline{\mathcal{C}_l} \mid \mathcal{B}_l\right] \cdot (1/2) + \Pr\left[\mathcal{C}_l \mid \mathcal{B}_l\right] \cdot (1+\mathbf{q})/2) \\
&= \mathbf{q}^{2(i-l)+1} \cdot ((1-\mathbf{q}^{2(k-i)}) \cdot (1/2) + \mathbf{q}^{2(k-i)} \cdot (1+\mathbf{q})/2) \\
&= \mathbf{q}^{2(i-l)+1} \cdot (1/2) \cdot (1+\mathbf{q}^{2(k-i)+1}).
\end{aligned}$$

Comparing with (6.15), we see that if $l \le i$, then $\Pr[\mathcal{A}_l] \le \mathbf{q}^{2(i-l)+1} \cdot (1/2) \cdot (1+\mathbf{q}^{2(k-i)+1})$. Summing over all $l \le i$, we get that the total contribution of all D_l, $l \le i$, to the coefficient of $c(P'_i)$ in the expected connection cost, is at most

$$\frac{\mathbf{q}}{2(1-\mathbf{q}^2)} \cdot (1+\mathbf{q}^{2(k-i)+1}). \tag{6.16}$$

We now move to the next case.

Case II: $i < l \le k$**.** Let $\mathcal{A}_l$ be as in Case I. Now define $\mathcal{B}'_l \equiv U(\{j : |j-l| \le l-(i+1)\})$. Once again, $\mathcal{B}'_l$ is a necessary condition for $\mathcal{A}_l$. Also note that $\Pr[\mathcal{B}'_l] = 0$ if $k+1-l \le l-(i+1)$; thus, we can restrict attention to those $l > i$ such that $l \le l_0 \doteq \lfloor (k+i+1)/2 \rfloor$. Now, since $l > i$, it is easy to see that $\Pr\left[\mathcal{A}_l \mid \mathcal{B}'_l\right] \le 1/2$; also, $\Pr[\mathcal{B}'_l] = \mathbf{q}^{2(l-i-1)+1}$. Summing over all $l \in [i+1, l_0]$, we get a total contribution of

$$(1/2) \cdot \sum_{j=1}^{l_0-i} \mathbf{q}^{2j-1} \le \frac{\mathbf{q}}{2(1-\mathbf{q}^2)} \cdot (1-\mathbf{q}^{k-i+1}). \tag{6.17}$$

Adding with (6.17), the coefficient of $c(P'_i)$ in the expected connection cost is seen to be at most

$$\frac{\mathbf{q}}{2(1-\mathbf{q}^2)} \cdot (2+\mathbf{q}^{2(k-i)+1} - \mathbf{q}^{k-i+1}) \le \frac{\mathbf{q}}{1-\mathbf{q}^2} \le \frac{e^{-\alpha}}{1-e^{-2\alpha}} = \frac{e^{\alpha}}{e^{2\alpha}-1}.$$

So, the total expected connection cost is at most $2C' + \frac{e^{\alpha}}{e^{2\alpha}-1} \cdot S'$. Using Lemma 4.2 to replace C' by $C^* + S^*$ and S' by $2S^*$, the expected connection cost is at most

$$2C^* + \frac{2(e^{2\alpha}+e^{\alpha}-1)}{e^{2\alpha}-1} \cdot S^*.$$

Adding with (4.2), the total expected cost is at most

$$2(1+\alpha)C^* + \frac{2(2e^{2\alpha}+e^{\alpha}-2)}{e^{2\alpha}-1} \cdot S^*.$$

Since the optimal cost is $Z^* = C^* + S^*$, we choosing $\alpha = 1.296$ to minimize the approximation ratio, to get the following result.

Theorem 6.1 *The algorithm* SimpleCFL *is a* $\beta = 4.6$*-approximation algorithm for* CFL.

Appendix B: Details of Some Proofs

Proof of 3.4: Consider the cost-shares $\beta\,\xi$. For a fixed random sample ω, let F_ω denote the set of facilities; i.e., $F_\omega = \{r\} \cup \{\, j \in D \mid Y_j(\omega) = 1 \,\}$. From our definition of $\xi(D,i)$, we get the following:

$$\textstyle\sum_{i\in D} \beta\,\xi(D,i) = \mathrm{E}_\omega[\, M \displaystyle\sum_{i\in D} \xi_{MST}(F_\omega, i) + \sum_{i\in D} \ell(i, F_\omega)\,] = \mathrm{E}_\omega[\, M\,\mathsf{MST}(F_\omega) + \sum_{i\in D} \ell(i, F_\omega)\,],$$

using the facts that ξ_{MST} is a cost-share function and hence $\xi_{MST}(F_\omega, i) = 0$ for $i \notin F_\omega$, and that ξ_{MST} is budget-balanced and thus $\sum_{i \in F_\omega} \xi_{MST}(F_\omega, i) = c(\mathsf{MST}(F_\omega))$.

Now it is easy to see that the cost-shares $\beta\,\xi$ recover the cost of the solution constructed, and hence are at least $\mathsf{Alg}(D)$. This is true, as for each element of the sample space ω, $M \times \mathsf{MST}(F_\omega) + \sum_i \ell(i, F_\omega)$ is the cost of the solution which opens facilities at F_ω.

To see the other direction, note that the final expression is the expected cost of running SimpleCFL on the set of demands D, and hence Theorem 3.3 implies that the expectation is no more than $\beta\,\mathsf{OPT}(D)$, and thus $\sum_{i \in D} \xi(D, i) \leq \mathsf{OPT}(D)$. ∎

Proof of 4.1: Consider the optimal solution OPT: it connects each $j \in D$ to a facility $i^*(j) \in F^*$ via a shortest path, and hence pays a connection cost of $C^* = \sum_{j \in D} \ell(j, i^*(j))$. Let F be as defined in Step (S2). Define a graph connecting F thus: buy all the edges in T^*, and then for each $j \in F$, buy the edges in the shortest path from j to $i^*(j)$ used by OPT. Note that we pay $M\,c_e$ for each edge e in Step (S2), and hence pay $M\,c(T^*) = S^*$ for edges of T^*. For the random paths we add, the expected cost of the path between j and $i^*(j)$ is $\frac{\alpha}{M} \times M\,\ell(j, i^*(j))$; summing over all $j \in D$ gives the expected total cost of these paths to be αC^*. Since we buy a minimum spanning tree in Step (S2), the expected buying cost of our tree T is at most $2\,(S^* + \alpha C^*)$. ∎

Proof of 4.2: Given the tree T^*, we process this in a two-pass fashion. The first pass goes bottom-up, starting at the leaves which must all be in F^*. Consider a vertex all of whose descendants satisfy the mod-M property, and let x be the demand currently assigned to it. If $x = aM + b$, where $b < M$, we send b units of demand from x to the its parent. The number of demands assigned to each is now a multiple of M. Furthermore, the increase in connection cost is $b < M$ times the distance between x and its parent. this ensures that every node (except possibly r) has demands that are a multiple of M.

Of course, we may now have demands gathering at nodes of OPT that are not in F^*; handling this is fairly simple in a top-down fashion. We will send some of the demand back down the tree from parents to children; however, we maintain the invariant that at most M unit of demand will be send down an edge. Let a vertex $v \notin F^*$ have aM demand (and let us assume we have handled all its ancestors). Since at least $(a-1)M$ of this demand must have been collected in the bottom-up pass, and each child could have only sent $< M$ demand upwards, there must be $\geq a$ children of v sending demands to it. We choose any a of these children and send back M units of demand along those edges.

To complete the argument, note that the net demand sent along an edge of T^* was either b units upwards, or $M-b$ units downwards, and hence the cost of this movement can be changed to S^*, thus obtaining that $C' \leq C^* + S^*$. Furthermore, since we finally do not use any new facilities, and can even close the facilities with no demands assigned to it, $F' \subseteq F^*$. Finally, the cycle property is obtained by taking an Eulerian tour of T^* and replacing it by a cycle by short-cutting repeated vertices using the metric property. This at most doubles the Steiner cost, giving $S' \leq 2S^*$. ∎