2014 IEEE Annual Symposium on Foundations of Computer Science

LP-Based Algorithms for Capacitated Facility Location

Hyung-Chan An*, Mohit Singh† and Ola Svensson*
*School of Computer and Communication Sciences, EPFL.
Email: {hyung-chan.an, ola.svensson}@epfl.ch
†Microsoft Research, Redmond.
Email: mohits@microsoft.com

***Abstract*—Linear programming has played a key role in the study of algorithms for combinatorial optimization problems. In the field of approximation algorithms, this is well illustrated by the uncapacitated facility location problem. A variety of algorithmic methodologies, such as LP-rounding and primal-dual method, have been applied to and evolved from algorithms for this problem. Unfortunately, this collection of powerful algorithmic techniques had not yet been applicable to the more general capacitated facility location problem. In fact, all of the known algorithms with good performance guarantees were based on a single technique, local search, and no linear programming relaxation was known to efficiently approximate the problem.**

In this paper, we present a linear programming relaxation with constant integrality gap for capacitated facility location. We demonstrate that the fundamental theories of multi-commodity flows and matchings provide key insights that lead to the strong relaxation. Our algorithmic proof of integrality gap is obtained by finally accessing the rich toolbox of LP-based methodologies: we present a constant factor approximation algorithm based on LP-rounding.

***Keywords*-approximation algorithms; facility location; linear programming;**

I. Introduction

We consider the metric capacitated facility location (CFL) problem which together with the metric uncapacitated facility location (UFL) problem is the most classical and widely studied variant of facility location. In CFL, we are given a single metric on the set of *facilities* and *clients*, and every facility has an associated *opening cost* and *capacity*. The problem asks us to choose a subset of facilities to open and assign every client to one of these open facilities, while ensuring that no facility is assigned more clients than its capacity. Our aim is then to find a set of open facilities and an assignment that minimize the cost, where the cost is defined as the sum of opening costs of each open facility and the distance between each client and the facility it is assigned to. UFL is the special case of CFL obtained by dropping the capacity constraints, or equivalently setting each capacity to ∞.

In spite of the similarities in the problem definitions of UFL and CFL, current techniques give a considerably better understanding of the uncapacitated version. One prominent reason for this discrepancy is that a standard linear programming (LP) relaxation gives close-to-tight bounds for UFL, whereas no good relaxation was known in the presence of capacities. For UFL, on the one hand, the standard LP formulation has been used in combination with most LP-based techniques, such as filtering [1], randomized rounding [2], [3], primal-dual framework [4], and dual fitting [5], [6], to obtain a fine-grained understanding of the problem resulting in a nearly tight approximation ratio [7].

For CFL, on the other hand, it has remained a major open problem to find a relaxation based algorithms with *any* constant performance guarantee, also highlighted as Open Problem 5 in the list of ten open problems selected by the recent textbook on approximation algorithms of Williamson and Shmoys [8]. This question is especially intriguing as there exist constant factor approximation algorithms for CFL based on the local search paradigm (see e.g. [9]–[14] and Section I-B for further discussion of this approach). Compared to such methods, there are several advantages of algorithms based on relaxations. First, as alluded to above, there is a large toolbox of LP-based techniques that one can tap into once a strong relaxation is known for a problem. Second, LP-based algorithms give a stronger *per instance* guarantee: that is, the rounded solution is compared to the found LP solution and the gap is often smaller than the worst case. This is in contrast to local search heuristics that only guarantees that the cost is no worse than the proven *a priori* performance guarantee assures. Finally, LP-based techniques are often flexible and allow for generalizations to related problems. This has indeed been the case for the uncapacitated versions where algorithms for UFL are used in the design of approximation algorithms for other related problems, see for example [4], [15]–[17].

In this pursuit of LP-based approximation algorithms for capacitated facility location problems, the central question lies in devising a strong LP relaxation that is *algorithmically amenable*. In fact, any combinatorial problem has a relaxation with constant integrality gap: the exact formulation, which is the convex hull of integral solutions, has indeed an integrality gap of 1. However, such formulations for NP-hard optimization problems usually have insufficient structure to give enough insights for designing efficient approximation algorithms. The challenge, therefore, is to instead devise an LP-relaxation that is sufficiently strong while featuring enough structure so as to guide the development of efficient

0272-5428/14 $31.00 © 2014 IEEE
DOI 10.1109/FOCS.2014.35

approximation algorithms using LP-based techniques, such as LP-rounding or primal-dual. However, formulating one for the capacitated facility location problem has turned out to be non-trivial. Aardal et al. [18] made a comprehensive study of valid inequalities for capacitated facility location problem and proposed further generalizations; the strength of the obtained formulations was left as an open problem. Many of these formulations were, however, recently proven to be insufficient for obtaining a constant integrality gap by Kolliopoulos and Moysoglou [19]. In the same paper it is also shown that applying the Sherali-Adams hierarchy to the standard LP formulation will not close the integrality gap.

A. Our Contributions

Our main contribution is a strong linear programming relaxation which has a constant integrality gap for the capacitated facility location problem. We prove its constant integrality gap by presenting a polynomial time approximation algorithm which rounds the LP solution.

Theorem I.1. *There is a linear programming relaxation (*MFN-LP *given in Figure 3) for the capacitated facility location problem that has a constant integrality gap. Moreover, there exists a polynomial-time algorithm that finds a solution to the capacitated facility location problem whose cost is no more than a constant factor times the LP optimum.*

This result resolves Open Problem 5 in the list of ten open problems selected by the textbook of Williamson and Shmoys [8].

Our relaxation is formulated based on multi-commodity flows. We will discuss in this section why the multi-commodity flow is a natural tool of choice in designing strong LP relaxations for our problem, and also how it plays a key role, together with the matching theory, in achieving a constant factor LP-rounding algorithm.

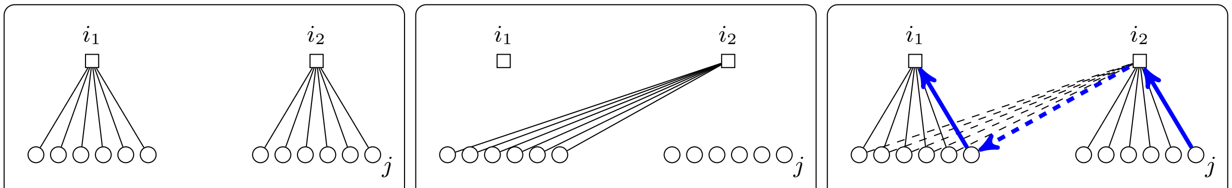
One natural question that arises is characterizing the exact integrality gap of our relaxation. While we prioritized ease of reading over a better ratio in the choice of parameters for this presentation of our algorithm, it appears that the current analysis is not likely to give any approximation ratio better than 5, the best ratio given by the local search algorithms [10]. On the other hand, the best lower bound known on the integrality gap of our relaxation is 2, and the question remains open whether we can obtain an approximation algorithm with a ratio smaller than 5 based on our relaxation.

Open Question. Determine the integrality gap of the LP relaxation MFN-LP.

High-level description of MFN-LP*:* The minimum knapsack problem is a special case of capacitated facility location: given a target value and a set of items with different values and costs, the problem is to find a minimum-cost subset of items whose total value is no less than the given target. Carr et al. [20] showed that flow-cover inequalities [20], [21] yield an LP with a constant integrality gap for this problem; in fact, another aspect of our relaxation shares a similar spirit as these inequalities. The flow-cover inequalities for the minimum knapsack problem say that, when any subset of items is given for "free" to be part of the solution, the LP solution should be feasible to the residual problem. In this residual problem, the target value is decreased by the total target value of the free items; hence, constraints of the residual problem can be strengthened by updating the values of all items to be at most the new target value.

In order to have a similar notion of residual sub-problems in the facility location problem, it is tempting to formulate a sub-problem for each subset of facilities which are open for free. Indeed the knapsack problem suggests exactly this sub-problem, since in the reduction from the knapsack problem to the facility location problem, items correspond to facilities. However, we take a different approach. Observe that there are two types of decisions to be made in the facility location problem: which facilities to open, and how to assign the clients to these open facilities; we focus on the latter. We contemplate an assignment of a subset of clients to some facilities, and insist that this assignment should be a part of the solution. We formulate the residual problem on unassigned clients, update the capacity of each facility and reduce it by the number of clients assigned for free to this facility. We now require that any feasible solution to the problem must contain a feasible solution to the residual problem. We call the assignment of clients for free a *partial assignment*, as they assign only a subset of clients.

While the residual instance would be again an instance of the capacitated facility location, with fewer clients and facilities with reduced capacities, it is not clear whether restricting a feasible solution of the original problem forms a feasible solution to the residual problem. In fact, it does not. The partial assignment reduces capacities at facilities which the feasible solution might have used for clients remaining in the residual instance. To be concrete, consider a feasible integral solution depicted in Figure 1a and a partial assignment in Figure 1b. Note that client j was not assigned by the partial assignment, but in the residual instance, it cannot be assigned to facility i_2 as the original solution indicates. The partial assignment has already assigned enough clients to reduce the capacity of facility i_2 to zero in the residual instance. But observe that the fact that client j could not claim its original place means that some other client has taken its place; therefore, that client must have left behind its space somewhere else (at facility i_1 in this example). Thus we would want to assign client j to facility i_1 in the example. But how can we enable such an assignment in general? Our relaxation allows additional edges to be used for assignments in the residual instance. In particular, we make edges corresponding to the partial assignment available



(a) A feasible integral solution. Both facilities are open, and lines show the assignment.

(b) A partial assignment. Client j is unassigned.

(c) Original solution, augmented with partial assignment edges marked as dashed lines. Thick lines represent alternating path for j.

Figure 1. Example of partial assignment. Squares represent facilities of capacity 6; circles clients.

to be used to "undo" the partial assignment; observe that what we are now looking for is not a direct assignment of clients to facilities but *alternating paths* starting at each client in the residual instance to a facility with spare reduced capacity. We model this problem as a *multi-commodity flow* problem where every unassigned client demands a unit flow to be routed to a facility with residual capacity. In fact, it is crucial for obtaining a strong LP to use multi-commodity flows to model these assignments, as we will see in Section II.

LP-rounding algorithm: In Section III, we give an algorithmic proof of constant integrality gap by presenting a polynomial-time LP-rounding algorithm. An interesting feature of this algorithm is that it does not solve the LP to optimality. Instead, we will give a rounding procedure that either rounds a given fractional solution within a constant factor, or identifies a violated inequality. This approach has been previously used, see for example, Carr et al. [20] and Levi et al. [22].

As is the case for flow-cover inequalities, we do not know whether our relaxation can be separated in polynomial time. However, our rounding algorithm establishes that it suffices to separate it over a given partial assignment in order to obtain a constant approximation algorithm: in a sense, such limited separation is already enough to extract the power of our strong relaxation within a constant factor. That said, it remains an interesting open question whether our relaxation can be separated in polynomial time. Another interesting open question would be whether there exists a different LP relaxation that can be solved in polynomial time and used to design a constant approximation algorithm.

Given a fractional solution consisting of *opening variables* and *assignment variables*, the first step our rounding algorithm takes is very natural: we decide to open all the facilities whose opening variables are large, say, at least $\frac{1}{2}$. The cost of opening these facilities is no more than twice the cost paid by the fractional solution. Now, we find an assignment of *maximum* number of clients to these facilities while maintaining that the assignment cost does not exceed twice the cost of fractional solution. If we manage to assign all clients to the integrally opened facilities, we are done since both the connection cost and facility opening cost can be bounded within a constant factor of the linear programming solution. Else, we obtain a partial assignment of clients to the opened facilities. We use this partial assignment to formulate the multi-commodity flow problem described earlier. Recall, in the multi-commodity flow problem, each unassigned client has a flow commodity which it needs to sink at the facilities using alternating paths. Assume for simplicity, that in the partial assignment all facilities that we opened in the first step are saturated. Now, in the multi-commodity flow problem, a client can only sink flow at facilities with small fractional value because the facilities with large fractional value have zero capacity since they are saturated by the partial assignment. Thus, the flow solution naturally gives us a fractional assignment of remaining unassigned clients to facilities which are open to a small fractional value. In the last step of the algorithm, we round this fractional solution obtained via the flow problem. But why is this problem any easier than the one we started with? Since each facility opening variable is at most $\frac{1}{2}$, the fractional solution can use at most half the capacity of any facility in the residual instance. Thus the capacity constraints are not stringent and we can appeal to known soft-capacitated approximation algorithms which approximate cost while violating capacity to a small factor (two suffices for us). Indeed, such algorithms can be obtained by rounding the standard linear program and we use the result of Abrams et al. [23]. This also implies that an immediate improvement to the approximation ratio of our algorithm would be possible by providing an improved algorithm for the soft-capacitated problem.

In summary, we have used techniques from the theory of multi-commodity flows and matchings to formulate the first linear programming relaxation for the capacitated facility location problem that efficiently approximates the optimum value within a constant. Our proposed LP-rounding algorithm exploits the properties of the multi-commodity flows obtained by solving the linear program and we give a constant factor approximation algorithm for the problem. Our results further open up the possibility to approach the capacitated facility location problem and other related problems using the large family of known powerful LP-based techniques.

B. Further Related Work

Uncapacitated facility location: Since the first constant factor approximation algorithm for UFL was given by Shmoys, Tardos and Aardal [3], several techniques have been developed around this problem. Currently, the best approximation guarantee of 1.488 is due to Li [7]; see also [6], [24]. On the hardness side, Guha and Khuller [25] shows that it is hard to approximate UFL within a factor of 1.463.

Local search heuristics for capacitated facility location: All previously known constant factor approximation algorithms for CFL are based on the local search paradigm. The first constant factor approximation algorithm was obtained in the special case of uniform capacities (all capacities being equal) by Korupolu et al. [12] who analyzed a simple local search heuristic proposed by Kuehn and Hamburger [13]. Their analysis was then improved by Chudak and Williamson [11] and the current best 3-approximation algorithm for this special case is a local search by Aggarwal et al. [9]. For the general problem (CFL), Pál et al [14] gave the first constant factor approximation algorithm. Since then more and more sophisticated local search heuristics have been proposed, the current best being a recent local search by Bansal et al. [10] which yields a 5-approximation algorithm.

Relaxed notions of capacity constraints: Several special cases or relaxations of the capacitated facility location problem have been studied. One popular relaxation is the soft-capacitated problems where the capacity constraints are relaxed in various ways. The standard linear program still gives a good bound for many of these relaxed problems. Shmoys et al. [3] gives the first constant factor approximation algorithm where a facility is allowed to be open multiple times, later improved by Jain and Vazirani [4]. Mahdian et al. [26] gives the current best approximation ratio of 2, which is tight with respect to the standard LP. Abrams et al. [23] studies a variant where a facility can be open at most once, but the capacity can be violated by a constant factor. We also mention that in our approximation algorithm, we use this variant of relaxed capacities as a subproblem. Finally, another special case for which the standard LP has been amenable to is the case of uniform opening costs, i.e., when all facilities have the same opening cost. For that case, Levi et al. [27] gives a 5-approximation algorithm.

We also mention that LP-based approximation algorithms which do not solve the linear program to optimality have been used in the works of Carr et. al [20] and Levi et. al. [22]. In a similar spirit, many primal-dual algorithms do not solve linear programs to optimality (see e.g. [28], [29]), while finding approximate solutions whose guarantee is given by comparison to a feasible dual solution.

Finally, we note that Chakrabarti, Chuzhoy and Khanna [30] used a collection of flow problems to obtain improved approximation algorithms for the max-min allocation problem.

II. Multi-commodity Flow Relaxation

We present our new relaxation for the capacitated facility location problem in this section. Let us first define some notation to be used in the rest of this paper. Let $\mathcal{F}$ be the set of facilities and $\mathcal{D}$ be the set of clients. Each facility $i \in \mathcal{F}$ has opening cost o_i, and cannot be assigned more number of clients than its capacity U_i. We are also given a metric cost c on $\mathcal{F} \cup \mathcal{D}$ as a part of the input: c_{ij} denotes the distance between $i \in \mathcal{F}$ and $j \in \mathcal{D}$.

The variables of our relaxation is the pair $(\boldsymbol{x}, \boldsymbol{y})$ where we refer to $\boldsymbol{x} \in [0,1]^{\mathcal{F}\times\mathcal{D}}$ as the *assignment variables* and to $\boldsymbol{y} \in [0,1]^{\mathcal{F}}$ as the *opening variables*. These variables naturally encode the decisions to which facility a client is connected and which facilities that are opened. Indeed, the intended integral solution is that $x_{ij} = 1$ if client j is connected to facility i and $x_{ij} = 0$ otherwise; $y_i = 1$ if facility i is opened and $y_i = 0$ otherwise. The idea of our relaxation is that every partial assignment of clients to facilities should be extendable to a complete assignment while only using the assignments of $\boldsymbol{x}$ and openings of $\boldsymbol{y}$. To this end let us first describe the partial assignments that we shall consider. We then define the constraints of our linear program which will be feasibility constraints of multi-commodity flows.

A partial fractional assignment $\boldsymbol{g} = \{g_{ij}\}_{i\in\mathcal{F}, j\in\mathcal{D}}$ of clients to facilities is *valid* if

$$\begin{aligned} \forall j \in \mathcal{D}: \quad & \sum_{i\in\mathcal{F}} g_{ij} \leq 1, \\ \forall i \in \mathcal{F}: \quad & \sum_{j\in\mathcal{D}} g_{ij} \leq U_i \qquad \text{and} \\ \forall i \in \mathcal{F}, j \in \mathcal{D}: \quad & g_{ij} \geq 0. \end{aligned}$$

The first condition says that each client should be fractionally assigned at most once and the second condition says that no facility should receive more clients than its capacity. We emphasize that we allow clients to be fractionally assigned, i.e., $\boldsymbol{g}$ is not assumed to be integral. As we shall see later (see Lemma II.3), this does not change the strength of our relaxation but it will be convenient in the analysis of our rounding algorithm in Section III. We also remark that the above inequalities are exactly the b-matching polytope of the complete bipartite graph consisting of the clients on the one side and the facilities on the other side; each client can be matched to at most one facility and each facility i can be matched to at most U_i clients.

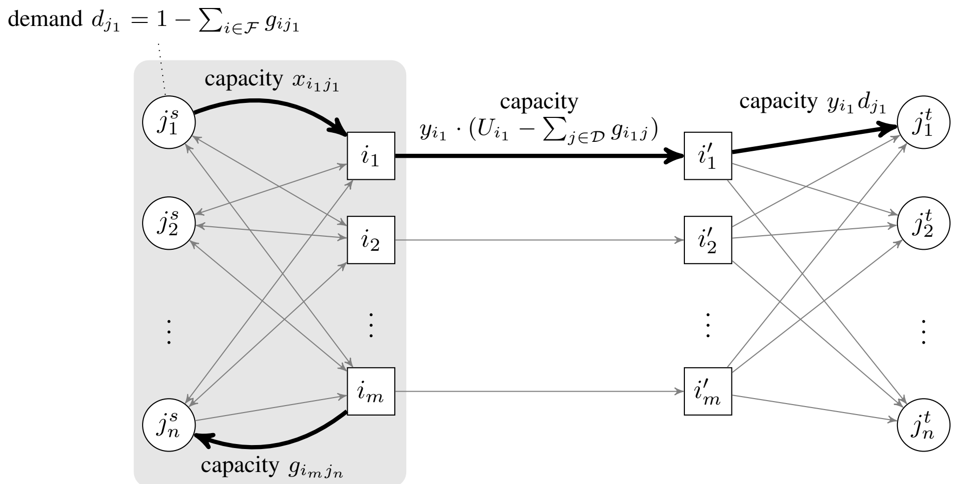


Figure 2. A depiction of the multi-commodity flow network $\mathrm{MFN}(\boldsymbol{g}, \boldsymbol{x}, \boldsymbol{y})$.

The constraints of our relaxation will be that, no matter how we partially assign the clients according to a valid $\boldsymbol{g}$, the solution $(\boldsymbol{x}, \boldsymbol{y})$ should support a multi-commodity flow where each client j becomes the source of its own commodity j, and the demand of this commodity is equal to the amount by which j is "not assigned" by $\boldsymbol{g}$, $1-\sum_{i\in\mathcal{F}} g_{ij}$. The flow network, whose arc capacities are given as a function of $\boldsymbol{g}$ and the solution $(\boldsymbol{x}, \boldsymbol{y})$, is defined as follows (see also Figure 2):

Definition II.1 (Multi-commodity flow network)**.** *For a valid partial assignment $\boldsymbol{g}$, assignment variables $\boldsymbol{x} = \{x_{ij}\}_{i\in\mathcal{F}, j\in\mathcal{D}}$, and opening variables $\boldsymbol{y} = \{y_i\}_{i\in\mathcal{F}}$, let $\mathrm{MFN}(\boldsymbol{g}, \boldsymbol{x}, \boldsymbol{y})$ be a multi-commodity flow network with $|\mathcal{D}|$ commodities, defined as follows. Note that some arcs may have zero capacities.*

(a) Each client $j \in \mathcal{D}$ is associated with commodity j of demand $d_j := 1-\sum_{i\in\mathcal{F}} g_{ij}$, and its source-sink pair is (j^s, j^t).
(b) Each facility $i \in \mathcal{F}$ has two nodes i and i' with an arc (i, i') of capacity $y_i \cdot (U_i - \sum_{j\in\mathcal{D}} g_{ij})$.
(c) For each client j and facility i, there is an arc (j^s, i) of capacity x_{ij}, an arc (i, j^s) of capacity g_{ij}, and an arc (i', j^t) of capacity $y_i d_j$.

Remark II.2. *Intuitively, the bipartite subgraph induced by $\{j^s\}_{j\in\mathcal{D}} \cup \{i\}_{i\in\mathcal{F}}$, marked with a shaded box in Figure 2, is the interesting part of the flow network. $\{i'\}_{i\in\mathcal{F}}$ and $\{j^t\}_{j\in\mathcal{D}}$ are added to this bipartite graph purely in order to state that i is a sink with "double" capacities: a commodity-oblivious capacity $y_i \cdot (U_i - \sum_{j\in\mathcal{D}} g_{ij})$ and a commodity-specific capacity $y_i d_j$ for each client $j \in \mathcal{D}$.*

Let us give some intuition on the definition of $\mathrm{MFN}(\boldsymbol{g}, \boldsymbol{x}, \boldsymbol{y})$. As already noted, the demand $d_j = 1 - \sum_{i\in\mathcal{F}} g_{ij}$ of a client j equals the amount by which j is not assigned by the partial assignment $\boldsymbol{g}$. This demand should only be assigned to opened facilities. Therefore, facility i can accept at most $y_i d_j$ of j's demand which is either d_j or 0 in an integral solution. Observe that such a constraint, for each client and facility, cannot be imposed by a single-commodity flow problem. Multi-commodity flow problems, on the other hand, allows us to express this constraint as a commodity-specific capacity of $y_i d_j$, as denoted by arc (i', j^t) in Figure 2.

Now consider the *commodity-oblivious* capacities of the facilities. The *total* demand an opened facility i can accept is its capacity minus the amount of clients assigned to it in the partial assignment $\boldsymbol{g}$; and a closed facility can accept no demand. Therefore, the total demand a facility i can accept is at most $y_i(U_i - \sum_{j\in\mathcal{D}} g_{ij})$. The arc capacity x_{ij} of an arc (j, i) says that client j should be connected to facility i only if $x_{ij} = 1$. The reason for having arcs of the form (i, j) of capacity g_{ij} is discussed in Section I-A: these allow the alternating paths for routing the remaining demand and are necessary for the formulation to be a relaxation.

We are now ready to formally state our relaxation MFN-LP of the capacitated facility location problem in Figure 3. Note that the only variables of our relaxation are the assignment variables $\boldsymbol{x}$ and the opening variables $\boldsymbol{y}$. While it is natural to formulate each of the multi-commodity flow problem using auxiliary variables denoting the flow, our algorithm will utilize the equivalent formulation obtained via projecting out the flow variables. This projected formulation is a relaxation where the only variables are assignment variables $\boldsymbol{x}$ and the opening variables $\boldsymbol{y}$.

In Lemma II.3 we show that the constraints of MFN-LP can equivalently be formulated over the subset of valid partial assignments that are integral. MFN-LP can therefore be seen as the intersection of the feasible regions of finitely

$$\begin{array}{ll} \text{minimize} & c(\boldsymbol{x},\boldsymbol{y}) := \sum_{i\in\mathcal{F}} o_i \cdot y_i + \sum_{i\in\mathcal{F}, j\in\mathcal{D}} c_{ij}\cdot x_{ij}, \\ \text{subject to} & \text{MFN}(\boldsymbol{g},\boldsymbol{x},\boldsymbol{y}) \text{ is feasible } \quad \forall \boldsymbol{g} \text{ valid;} \\ & \boldsymbol{x}\in[0,1]^{\mathcal{F}\times\mathcal{D}}, \boldsymbol{y}\in[0,1]^{\mathcal{F}}. \end{array}$$

Figure 3. Our relaxation of CFL.

many multi-commodity flow linear programs and is therefore itself a linear program. At first sight, however, it may not be clear that MFN-LP is a relaxation, or how we can separate it. We will answer these questions in the rest of this section.

A. Integral Partial Assignments and Separation

We first present a useful lemma that allows us to consider only the valid assignments $\boldsymbol{g}$ that are integral, i.e., $\{0,1\}$-matrices. This lemma follows from the integrality of the b-matching polytope.

Lemma II.3. *For any* $(\boldsymbol{x},\boldsymbol{y})$, $\text{MFN}(\boldsymbol{g},\boldsymbol{x},\boldsymbol{y})$ *is feasible for all valid* $\boldsymbol{g}$ *if and only if* $\text{MFN}(\hat{\boldsymbol{g}},\boldsymbol{x},\boldsymbol{y})$ *is feasible for all valid* $\hat{\boldsymbol{g}}$ *that are integral.*

Proof: It is clear that if the flow network is feasible for all valid $\boldsymbol{g}$ then it is also feasible for the subset that are integral. We show the harder side. Suppose $\text{MFN}(\hat{\boldsymbol{g}},\boldsymbol{x},\boldsymbol{y})$ is feasible for all valid $\hat{\boldsymbol{g}}$ that are integral and consider an arbitrary valid assignment $\boldsymbol{g}$ that may be fractional. We will show that $\text{MFN}(\boldsymbol{g},\boldsymbol{x},\boldsymbol{y})$ is feasible.

Construct a complete bipartite graph with vertices $\mathcal{F}\cup\mathcal{D}$ and interpret $\boldsymbol{g}$ as the weights on the edges of this complete bipartite graph. As $\boldsymbol{g}$ is valid, we have $\sum_{j\in\mathcal{D}} g_{ij} \le U_i$ for each $i\in\mathcal{F}$ and $\sum_{i\in\mathcal{F}} g_{ij}\le 1$ for each $j\in\mathcal{D}$. In other words, $\boldsymbol{g}$ is a fractional solution to the b-matching polytope. By the integrality of the b-matching polytope (see e.g. [31]), we can write $\boldsymbol{g}$ as a convex combination of valid integral assignments $\hat{\boldsymbol{g}}^1, \hat{\boldsymbol{g}}^2, \ldots, \hat{\boldsymbol{g}}^r$, i.e., there exist $\lambda_1,\lambda_2,\ldots,\lambda_r \ge 0$ such that $\sum_{k=1}^r \lambda_k = 1$ and $\boldsymbol{g} = \sum_{k=1}^r \lambda_k \hat{\boldsymbol{g}}^k$.

Now, let $\boldsymbol{f}^k$ denote the feasible flow for $\text{MFN}(\hat{\boldsymbol{g}}^k,\boldsymbol{x},\boldsymbol{y})$, and choose $\boldsymbol{f} = \sum_k \lambda_k \boldsymbol{f}^k$. Observe that $\boldsymbol{f}$ is a feasible solution to $\text{MFN}(\boldsymbol{g},\boldsymbol{x},\boldsymbol{y})$, since all the capacities and demands of $\text{MFN}(\cdot,\boldsymbol{x},\boldsymbol{y})$ are given as linear functions of $\boldsymbol{g}$. ■

A natural question is whether MFN-LP can be separated in polynomial time. While we currently do not know if this is the case, we will establish in this paper that the feasibility constraint of $\text{MFN}(\boldsymbol{g},\boldsymbol{x},\boldsymbol{y})$ can be separated for any fixed $\boldsymbol{g}$, and that this is sufficient to find a fractional solution whose cost is within a constant factor from the optimum: in a sense, this oracle enables us to extract the power of our strong relaxation within a constant factor. The following lemma states the oracle. It follows from known characterizations using LP-duality of multi-commodity flows and its proof can be found in the full version [32] of this extended abstract.

Lemma II.4. *Given* $\boldsymbol{g}^\star$ *in addition to* $(\boldsymbol{x}^\star,\boldsymbol{y}^\star)$ *such that* $\text{MFN}(\boldsymbol{g}^\star,\boldsymbol{x}^\star,\boldsymbol{y}^\star)$ *is infeasible, we can find in polynomial time a violated inequality, i.e., an inequality that is valid for* MFN-LP *but violated by* $(\boldsymbol{x}^\star,\boldsymbol{y}^\star)$*. Moreover, the number of bits needed to represent each coefficient of this inequality is polynomial in* $|\mathcal{F}|$, $|\mathcal{D}|$, *and* $\log U$, *where* $U := \max_{i\in\mathcal{F}} U_i$.

B. MFN-LP *is a Relaxation of the Capacitated Facility Location Problem*

We show in this subsection that MFN-LP is indeed a relaxation.

Lemma II.5. MFN-LP *is a relaxation of the capacitated facility location problem.*

Proof: Consider an arbitrary integral solution $(\boldsymbol{x}^\star,\boldsymbol{y}^\star)$ to the facility location problem. By Lemma II.3 we only need to verify that $\text{MFN}(\boldsymbol{g},\boldsymbol{x}^\star,\boldsymbol{y}^\star)$ is feasible for each valid integral assignment $\boldsymbol{g}$. Let $\hat{\boldsymbol{g}}$ be an arbitrary valid integral assignment.

Now we consider a directed bipartite graph $G=(V,A)$, of which one side of the vertex set is $\mathcal{D}$, and on the other side, each facility $i\in\mathcal{F}$ appears in $y_i^\star\cdot U_i$ duplicate copies. Consider the following two matchings M_1 and M_2 on these vertices.

- For each client j, M_1 has an edge between j and (a copy of) i for which $x_{ij}^\star = 1$. There will always be a copy of i since $y_i^\star \ge x_{ij}^\star = 1$. We will also ensure that a single copy of a facility does not have more than one incident edge: this is possible due to the capacity constraints on $(x^\star, y^\star)$.
- For each (i,j) such that $\hat{g}_{ij}=1$ and $y_i^\star = 1$, M_2 has an edge between a copy of facility i and client j. Note that no client will have more than one incident edge since $\sum_{i\in\mathcal{F}}\hat{g}_{ij}\le 1$. We will also ensure that a single copy of a facility does not have more than one incident edge. This is possible since $\sum_{j\in\mathcal{D}}\hat{g}_{ij}\le U_i$.

Now we orient every edge in M_1 from clients to facilities; edges in M_2 are oriented in the opposite direction. A is defined as the union of these two directed matchings. Since both M_1 and M_2 are matchings, every vertex in G has indegree of at most one and outdegree of at most one. Hence, we can decompose A into a set of maximal paths and cycles. Moreover, since M_1 matches every client, none of these maximal paths will end at a client. Reinterpret these paths as paths on $\mathcal{D}$ and $\mathcal{F}$, instead of on the duplicate copies of facilities. Let $\mathcal{P}$ denote the set of these (nonempty) paths.

We will now construct a feasible multi-commodity flow on $\text{MFN}(\hat{\boldsymbol{g}},\boldsymbol{x}^\star,\boldsymbol{y}^\star)$. We consider each $P\in\mathcal{P}$. If P starts from a facility, ignore it; otherwise let j be the starting point of P and i the ending point: $P=(j,i_1,j_2,i_2,\ldots,j_k,i)$. If $d_j = 0$, we ignore P; otherwise, we push one unit of flow of commodity j along P, staying within the shaded area of Figure 2: i.e., the flow is pushed along $(j^s, i_1, j_2^s, i_2, \ldots, j_k^s, i)$. When we arrive at i, further push this flow along (i, i', j^t), draining the flow at j^t: this is legal since the flow is of commodity j. We repeat this until we have considered all paths in $\mathcal{P}$. We claim that this procedure yields a feasible multi-commodity flow.

First, note that each arc in A maps to an edge of capacity 1 in $\mathrm{MFN}(\hat{\boldsymbol{g}}, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$. Since $\mathcal{P}$ is a decomposition of (a subset of) A, capacity constraints on (j^s, i) and (i, j^s) are satisfied from the construction. Now consider the capacity of (i, i'). Each time we encounter a path $P \in \mathcal{P}$ that starts at some client and ends at i, one unit of additional flow is sent over this arc. If $y_i^\star = 0$, there will be no such path in $\mathcal{P}$. If $y_i^\star = 1$, there are at most $U_i - \sum_{j \in \mathcal{D}} \hat{g}_{ij}$ paths in $\mathcal{P}$ ending at i, since M_2 matches exactly $\sum_{j \in \mathcal{D}} \hat{g}_{ij}$ copies of i out of U_i in total. This verifies that the capacity constraint on (i, i') is also satisfied. Finally, arc (i', j^t) is used only when we process $P \in \mathcal{P}$ that starts from j and ends at i. This is true for at most one path in $\mathcal{P}$ since there is at most one path starting from each client (note that there are no duplicate copies of clients in G); moreover, P can end at i only if $y_i^\star = 1$ (otherwise, there are no copies of i in G). The capacity constraint on (i', j^t) is therefore also satisfied.

Demand constraints are also satisfied: suppose $d_j = 1$ for some $j \in \mathcal{D}$. This means $\hat{\boldsymbol{g}}$ does not assign j to any facility, and therefore M_2 does not match j. Hence j has indegree of zero and outdegree of one in G, and thus $\mathcal{P}$ contains exactly one path that starts from j. ■

Intuitively, the above proof can also be interpreted as follows: given an arbitrary partial assignment and integral solution, consider the shaded area of Figure 2. By saturating every arc in this area, we obtain a feasible single-commodity flow where every client generates a unit flow either at its original position or at the facility it is assigned to by $\boldsymbol{g}$. While this flow satisifies every commodity-oblivious capacity, it may not be immediately clear why it also satisfies the commodity-specific capacities; here we can appeal to the integrality of $\boldsymbol{y}^\star$, because in this case every facility with nonzero commodity-oblivious capacity will automatically have the full commodity-specific capacity of 1. Such an argument, however, would not extend to a fractional solution (to the standard LP for example), which illustrates the strength of our relaxation.

III. Approximation Algorithm

In this section, we describe our approximation algorithm and prove Theorem I.1: [1]

[1]The cost function includes two components, facility opening costs and connection costs. Optimizing the parameters to obtain the same worst case performance for both components will lead to significant improvements in the constant obtained above. But such methods will not lead to improvement over 5-approximation due to local search [10].

Theorem I.1 (restated)**.** *There exists a 288-approximation algorithm for the capacitated facility location problem. The cost of its output is no more than 288 times the optimal cost of* MFN-LP.

The algorithm is based on rounding a given fractional "solution" to MFN-LP. However, as we do not know how to solve MFN-LP exactly, we give a *relaxed* separation oracle that either outputs a violated inequality or returns an integral solution obtained from the fractional solution by increasing the cost only by a constant factor. A similar approach has previously been used by Carr et al. [20] and later by Levi et al. [22].

Algorithm overview: Our algorithm first guesses the cost of the optimal solution to MFN-LP using a binary search[2]. For each guess, say γ, we run an ellipsoid algorithm. At each step of the ellipsoid algorithm, we obtain a fractional solution $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$, possibly infeasible. We then first verify the boundary constraints $\mathbf{0} \leq \boldsymbol{x}^\star, \boldsymbol{y}^\star \leq \mathbf{1}$ and the objective constraint $c(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \leq \gamma$. If $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ violates one of these inequalities, we output it and continue to the next iteration of the ellipsoid algorithm. Otherwise, we either construct a so-called semi-integral solution (defined below) or output a violated inequality showing infeasibility of the flow network $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$ for some $\boldsymbol{g}^\star$. In the final step, our algorithm rounds this semi-integral solution into an integral solution by increasing the cost by a constant factor.

We remark that the main step of our algorithm exploiting the strength of MFN-LP is the step for finding a semi-integral solution or outputting a violated inequality (summarized in Theorem III.3). An interesting detail is that our rounding algorithm only needs that the multi-commodity flow network is feasible for a *single* $\boldsymbol{g}^\star$ in order to output a semi-integral solution. Once we have a semi-integral solution, the rounding is fairly straightforward using previous algorithms for soft-capacitated versions. We now first define semi-integral solutions and describe the rounding to integral solutions in Section III-A. We then continue with the proof of Theorem III.3 which is the main technical contribution of this section.

A. Semi-Integral Solutions: Definition and Rounding

The idea of semi-integral solutions is that they partition the facilities into two sets: the set I of integrally opened facilities and the set S of facilities of small opening. Clients may be fractionally assigned to both facilities in I and S. However, there is an important constraint regarding the assignment to facilities in S (condition (iii) in the definition below). For each client j, it says that at most a y_i fraction of j's total assignment to facilities in S can be assigned to

[2]We remark that the relaxed separation oracle can also simply be used with the standard optimization version of the ellipsoid method, which would not involve a binary search.

$i \in S$. This will allow us to round semi-integral solutions by using techniques developed for the standard LP relaxation.

Definition III.1. *A solution* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ *is* semi-integral *if it satisfies the following conditions.*

(i) $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ *satisfies the assignment constraints, i.e., for each* $j \in \mathcal{D}$, $\sum_{i \in \mathcal{F}} \hat{x}_{ij} = 1$ *and for each* $i \in \mathcal{F}$, $\sum_{j \in \mathcal{D}} \hat{x}_{ij} \le \hat{y}_i U_i$.

(ii) For each $i \in \mathcal{F}$, $\hat{y}_i = 1$ *or* $\hat{y}_i \le \frac{1}{2}$. *Let* $I = \{i : \hat{y}_i = 1\}$ *and* $S = \mathcal{F} \setminus I$.

(iii) For each $j \in \mathcal{D}$, *let* $\hat{d}_j = \sum_{i \in S} \hat{x}_{ij}$. *Then we have* $\hat{x}_{ij} \le \hat{y}_i \hat{d}_j$ *for each* $i \in S$ *and* $j \in \mathcal{D}$.

We now describe the procedure for rounding the semi-integral solution to an integral solution. All facilities in I, whose opening variables are set to one in the semi-integral instance, are opened. Consider the residual instance where each client has a residual demand $\hat{d}_j$, amount to which it is not assigned to facilities in I. This residual demand is satisfied by facilities in S, each of which is open to a fraction of at most $\frac{1}{2}$ by the semi-integral solution. Conditions (i) and (iii) of the semi-integral solution imply that the residual solution is a feasible solution to the standard LP for the residual instance. Since the opening variables are set to a small fraction in the residual instance, we can use an approximation algorithm for the soft-capacitated facility location problem which rounds the standard LP. An (α, β)-approximation algorithm for the soft-capacitated facility location problem returns a solution whose cost is no more than α times the cost of the optimal fractional solution and violates the capacity of any open facility by a factor of at most β. We give the algorithm our residual instance as input where we scale down the capacities by a factor of β but scale up the opening variables by the same factor. Observe that as long as each $\hat{y}_i \le \frac{1}{\beta}$ for each facility $i \in S$, we obtain a feasible solution to the standard LP. Here we use the $(18, 2)$-bicriteria approximation algorithm due to Abrams et al. [23] to complete our rounding to an integral solution.

Lemma III.2. *Given a semi-integral solution* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$, *we can in polynomial time find an integral solution* $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ *whose cost is at most* $36c(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$.

We give the formal proof of Lemma III.2 in the full version of this paper [32].

B. Finding a Semi-Integral Solution or a Violated Inequality

We are now ready to describe and prove the main ingredient of our rounding algorithm.

Theorem III.3. *There is a polynomial time algorithm that, given* $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$, *either*

- *shows that* $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ *is infeasible for* MFN-LP *and returns a violating inequality, or*
- *returns a solution* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ *such that* $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ *is semi-integral and* $c(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \le 8c(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$.

Note that the above theorem together with Lemma III.2 implies Theorem I.1 with the claimed approximation guarantee $8 \cdot 36 = 288$.

We prove the theorem by describing the algorithm together with its properties. The algorithm consists of several steps. First, we round up the large opening variables of $\boldsymbol{y}^\star$ to obtain modified opening variables $\boldsymbol{y}'$. We define

$$y'_i := \begin{cases} 1, & \text{if } y_i^\star \ge \frac{1}{4}; \\ y_i^\star, & \text{otherwise.} \end{cases}$$

Let I be the set of facilities that are fully open by $\boldsymbol{y}'$: $I := \{i \in \mathcal{F} : y'_i = 1\}$. S denotes the set of facilities that are open by a small fraction: $S := \mathcal{F} \setminus I$.

Given that our algorithm is going to open all the facilities in I, we will try to find a partial assignment $\boldsymbol{g}^\star$ that assigns as many clients to these facilities as possible, while at the same time ensuring $\boldsymbol{g}^\star$ does not become too costly compared to $\boldsymbol{x}^*$. To this end, we will derive $\boldsymbol{g}^\star$ from a maximum b-matching in a bipartite graph on $\mathcal{F}$ and $\mathcal{D}$ whose edges are capacitated by $2\boldsymbol{x}^*$. Let $G = (\mathcal{D}, I, E)$ be the complete bipartite graph whose bipartition is given by the clients $\mathcal{D}$ and the opened facilities I. An arc (j, i) where $j \in \mathcal{D}$ and $i \in I$ is given a capacity of $2x_{ij}^\star$. This is to ensure that the cost of the matching is within a factor of 2 compared to the original assignment cost. Every client j has a capacity of one and each facility $i \in I$ is given a capacity of U_i. Let $\boldsymbol{z}$ denote a maximum fractional b-matching of G. Note that the matching may not be integral because of the capacities on the edges. As $\boldsymbol{z}$ is a maximum fractional matching, its residual network H with arc set $\{(j, i) : z_{ij} < 2x_{ij}^\star\} \cup \{(i, j) : z_{ij} > 0\}$ has useful properties that we describe below. In particular, if we consider an *unsaturated* client j, i.e., $\sum_{i \in I} z_{ij} < 1$, then j has no path in H to a facility i with remaining capacity, as that would contradict that $\boldsymbol{z}$ is a maximum matching.

We shall now formalize these properties. Let us call a client $j \in \mathcal{D}$ *saturated* if $\sum_{i \in \mathcal{F}} z_{ij} = 1$, and *unsaturated* otherwise; define $I_H := \{i \in I : i \text{ is reachable in } H \text{ from some client } k \text{ that was unsaturated}\}$ and $D_H := \{j \in \mathcal{D} : j \text{ is reachable in } H \text{ from some client } k \text{ that was unsaturated}\}$. Similar to clients, a facility $i \in I$ is called *saturated* if $\sum_{j \in \mathcal{D}} z_{ij} = U_i$ and *unsaturated* otherwise. The following lemma summarizes three useful observations on $\boldsymbol{z}$ and H.

Lemma III.4. *The following must hold.*

(a) Any facility $i \in I_H$ *is saturated, i.e.,* $\sum_{j \in \mathcal{D}} z_{ij} = U_i$.

(b) If $i \in I \setminus I_H$ *and* $j \in D_H$, $z_{ij} = 2x_{ij}^\star$.

(c) If $i \in I_H$ *and* $j \in \mathcal{D} \setminus D_H$, $z_{ij} = 0$.

Proof: We first prove (a). Suppose toward contradiction that there exists a facility $i \in I_H$ that is not saturated. By the definition of I_H there exists a client k that is unsaturated and i is reachable from k in H. Therefore there exists an alternating path from k to i which contradicts that the chosen fractional matching $\boldsymbol{z}$ was maximum.

We now prove (b). By the definition of D_H, there exists an unsaturated client k such that j is reachable from k in H. Therefore, any facility i such that $z_{ij} < 2x^\star_{ij}$ is also reachable from k and therefore part of I_H. The proof of (c) follows from the fact that $(i,j) \notin H$ since i is reachable from an unsaturated client and j is not. Therefore, $z_{ij} = 0$. ■

Now the valid partial assignment $\boldsymbol{g}^\star$ is constructed as follows:

$$g^\star_{ij} = \begin{cases} z_{ij} & \text{if } i \in I_H \\ z_{ij} & \text{if } i \in I \setminus I_H, j \in \mathcal{D} \setminus D_H \\ 0 & \text{if } i \in I \setminus I_H, j \in D_H \\ 0 & \text{if } i \in S. \end{cases} \quad (1)$$

Note that $\boldsymbol{g}^\star$ is defined in terms of $\boldsymbol{z}$. This will allow us to analyze the flow network using the properties of $\boldsymbol{z}$ described in Lemma III.4.

Once we have this partial assignment, the algorithm verifies if $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is feasible. If not, we invoke Lemma II.4 to find a violated inequality and Theorem III.3 holds. Otherwise, the algorithm proceeds to construct a semi-integral solution using this partial assignment. For the rest of this section, we will assume that $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is feasible. Note that the feasibility of $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$ guarantees the feasibility of $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}')$ since $\boldsymbol{y}' \geq \boldsymbol{y}^\star$.

Claim III.5. *If $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is feasible and $\boldsymbol{y}' \geq \boldsymbol{y}^\star$, $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}')$ is feasible.*

Proof: Consider a feasible flow for $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}^\star)$. Observe that it is feasible for $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}')$ as well, since the arc capacities of $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y})$ is nondecreasing in $\boldsymbol{y}$ while the demands remain the same since they depend on $\boldsymbol{g}^\star$. ■

We have now made our choice of $\boldsymbol{g}^\star$ that satisfies the following three key properties which help us round $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$:

1) $\boldsymbol{g}^\star \leq \boldsymbol{z} \leq 2\boldsymbol{x}^\star$ and therefore $c(\boldsymbol{g}^\star) \leq 2c(\boldsymbol{x}^\star)$;
2) $\boldsymbol{g}^\star$ assigns clients only to the fully open facilities, i.e., facilities in I;
3) $\boldsymbol{g}^\star$ satisfies the property formalized by Lemma III.6. (Note that Lemma III.6 is proven for our carefully constructed partial assignment. It does not hold in general for arbitrary partial assignments.)

Let $\boldsymbol{f}$ denote the flow certifying the feasibility of $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}')$. We decompose $\boldsymbol{f}$ into flow paths where we let $\mathcal{P}_{ij}$ denote the set of flow paths carrying non-zero flow from j^s to j^t that use the arc (i, i'). That is, these are the paths which take flow from j and sink it at i. Let $f(P)$ denote the flow on a path $P \in \mathcal{P}_{ij}$. For each $i \in \mathcal{F}$ and $j \in \mathcal{D}$, we let $h(i,j) = \sum_{P \in \mathcal{P}_{ij}} f(P)$ denote the amount of flow that client j sinks at facility i. For any subset $X \subseteq \mathcal{F}$ and $j \in \mathcal{D}$, let also $h(X,j) := \sum_{i \in X} h(i,j)$, i.e., the total amount of flow that client j sinks at facilities in X.

Lemma III.6. *There exists a feasible flow to the multi-commodity flow problem $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}')$ such that each client $j \in \mathcal{D}$ sends at least half its demand to facilities in S, i.e., $h(S,j) \geq \frac{d_j}{2} = \frac{1}{2}(1 - \sum_{i \in \mathcal{F}} g^\star_{ij})$.*

The proof of the above lemma can be found in the full version of this paper [32].

Observe that the flow satisfying the conditions in Lemma III.6 can be obtained in polynomial time by adding additional linear constraints to the multi-commodity flow linear program for $\mathrm{MFN}(\boldsymbol{g}^\star, \boldsymbol{x}^\star, \boldsymbol{y}')$. Let $\boldsymbol{f}$ denote such a flow. The algorithm now proceeds by using this flow to define a semi-integral solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$. Lemma III.6 guarantees $h(S,j) \geq d_j/2$; hence we define the semi-integral solution by scaling up this assignment by a factor of at most 2. This ensures that each client assigns all its demand d_j to S and that it is a semi-integral solution. Formally, we construct the semi-integral solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ as follows:

$$\hat{y}_i = \begin{cases} 1, & \text{if } i \in I; \\ 2y^\star_i, & \text{if } i \in S; \end{cases}$$

$$\hat{x}_{ij} = \begin{cases} g^\star_{ij}, & \text{if } i \in I, j \in \mathcal{D}; \\ d_j \frac{h(i,j)}{h(S,j)}, & \text{if } i \in S, j \in \mathcal{D}; \end{cases}$$

where we define $\frac{h(i,j)}{h(S,j)}$ to be 0 if $h(S,j) = 0$. For $i \in S$ and $j \in \mathcal{D}$, we have $\hat{x}_{ij} = h(i,j) \cdot \frac{d_j}{h(S,j)} \leq 2h(i,j)$ from Lemma III.6. This allows us to bound the total cost of solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$.

Lemma III.7. *The solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is semi-integral and $c(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \leq 8c(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$.*

The above lemma finishes the proof of Theorem III.3. Its proof can be found in the full version of this paper [32].

ACKNOWLEDGMENT

We thank Tim Carnes and David Shmoys for inspiring discussions during the early stages of this work. We also thank the anonymous reviewers for their helpful comments.

The first author and the last author are supported by ERC Starting Grant 335288-OptApprox.

REFERENCES

[1] J. Lin and J. S. Vitter, "Approximation algorithms for geometric median problems," *Inf. Process. Lett.*, vol. 44, pp. 245–249, 1992.

[2] F. A. Chudak and D. B. Shmoys, "Improved approximation algorithms for the uncapacitated facility location problem," *SIAM J. Comput.*, vol. 33, no. 1, pp. 1–25, 2004.

[3] D. B. Shmoys, E. Tardos, and K. Aardal, "Approximation algorithms for facility location problems (extended abstract)," in *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 1997, pp. 265–274.

[4] K. Jain and V. V. Vazirani, "Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation," *J. ACM*, vol. 48, no. 2, pp. 274–296, 2001.

[5] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani, "Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP," *J. ACM*, vol. 50, pp. 795–824, November 2003.

[6] K. Jain, M. Mahdian, and A. Saberi, "A new greedy approach for facility location problems," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, ser. STOC '02, 2002, pp. 731–740.

[7] S. Li, "A 1.488 approximation algorithm for the uncapacitated facility location problem," in *Automata, Languages and Programming - 38th International Colloquium (ICALP)*, 2011, pp. 77–88.

[8] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*. Cambridge University Press, 2010.

[9] A. Aggarwal, A. Louis, M. Bansal, N. Garg, N. Gupta, S. Gupta, and S. Jain, "A 3-approximation algorithm for the facility location problem with uniform capacities," *Math. Program.*, vol. 141, no. 1-2, pp. 527–547, 2013.

[10] M. Bansal, N. Garg, and N. Gupta, "A 5-approximation for capacitated facility location," in *ESA*, 2012, pp. 133–144.

[11] F. A. Chudak and D. P. Williamson, "Improved approximation algorithms for capacitated facility location problems," *Math. Program.*, vol. 102, no. 2, pp. 207–222, 2005.

[12] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, "Analysis of a local search heuristic for facility location problems," *J. Algorithms*, vol. 37, no. 1, pp. 146–188, 2000.

[13] A. A. Kuehn and M. J. Hamburger, "A heuristic program for locating warehouses," *Management science*, vol. 9, no. 4, pp. 643–666, 1963.

[14] M. Pál, É. Tardos, and T. Wexler, "Facility location with nonuniform hard capacities," in *FOCS*, 2001, pp. 329–338.

[15] J. Byrka, A. Srinivasan, and C. Swamy, "Fault-tolerant facility location: a randomized dependent LP-rounding algorithm," in *Integer Programming and Combinatorial Optimization*. Springer, 2010, pp. 244–257.

[16] R. Krishnaswamy, A. Kumar, V. Nagarajan, Y. Sabharwal, and B. Saha, "The matroid median problem," in *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2011, pp. 1117–1130.

[17] S. Li and O. Svensson, "Approximating k-median via pseudo-approximation," in *STOC*, 2013, pp. 901–910.

[18] K. Aardal, Y. Pochet, and L. A. Wolsey, "Capacitated facility location: valid inequalities and facets," *Mathematics of Operations Research*, vol. 20, no. 3, pp. 562–582, 1995.

[19] S. G. Kolliopoulos and Y. Moysoglou, "Sherali-Adams gaps, flow-cover inequalities and generalized configurations for capacity-constrained facility location," *CoRR*, vol. abs/1312.0722, 2013.

[20] R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips, "Strengthening integrality gaps for capacitated network design and covering problems," in *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '00, 2000, pp. 106–115.

[21] L. A. Wolsey, "Faces for a linear inequality in 0-1 variables," *Mathematical Programming*, vol. 8, no. 1, pp. 165–178, 1975. [Online]. Available: http://dx.doi.org/10.1007/BF01580441

[22] R. Levi, A. Lodi, and M. Sviridenko, "Approximation algorithms for the multi-item capacitated lot-sizing problem via flow-cover inequalities," in *IPCO*, 2007, pp. 454–468.

[23] Z. Abrams, A. Meyerson, K. Munagala, and S. Plotkin, "On the integrality gap of capacitated facility location," Carnegie Mellon University, Tech. Rep. CMU-CS-02-199, 2002, see http://theory.stanford.edu/ za/CapFL/CapFL.pdf.

[24] J. Byrka and K. Aardal, "An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem," *SIAM J. Comput.*, vol. 39, no. 6, pp. 2212–2231, 2010.

[25] S. Guha and S. Khuller, "Greedy strikes back: Improved facility location algorithms," *J. Algorithms*, vol. 31, no. 1, pp. 228–248, 1999.

[26] M. Mahdian, Y. Ye, and J. Zhang, "A 2-approximation algorithm for the soft-capacitated facility location problem," in *RANDOM-APPROX*, 2003, pp. 129–140.

[27] R. Levi, D. B. Shmoys, and C. Swamy, "LP-based approximation algorithms for capacitated facility location," *Math. Program.*, vol. 131, no. 1-2, pp. 365–379, 2012.

[28] A. Agrawal, P. Klein, and R. Ravi, "When trees collide: An approximation algorithm for the generalized steiner problem on networks," *SIAM Journal on Computing*, vol. 24, no. 3, pp. 440–456, 1995.

[29] M. X. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," in *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '92, 1992, pp. 307–316.

[30] D. Chakrabarty, J. Chuzhoy, and S. Khanna, "On allocating goods to maximize fairness," in *FOCS*, 2009, pp. 107–116.

[31] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[32] H.-C. An, M. Singh, and O. Svensson, "LP-based algorithms for capacitated facility location," *CoRR*, vol. abs/1407.3263, 2014, http://arxiv.org/abs/1407.3263.