Approximation Algorithms for Soft-Capacitated Connected Facility Location Problems

Assaf Rappaport

Approximation Algorithms for Soft-Capacitated Connected Facility Location Problems

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Assaf Rappaport

Submitted to the Senate of
the Technion — Israel Institute of Technology
Sivan 5772 Haifa June 2012

The research thesis was done under the supervision of Prof. Danny Raz in the Computer Science Department.

I would like to thank my advisor Danny Raz for taking a chance and giving me an opportunity to study under his direction. I am grateful for his devoted guidance, encouragement, and patience throughout the last couple of years. I would like to thank Yardena Kolet for her relentless support.

Contents

Abstract **1**

Abbreviations and Notations **2**

1 Introduction **3**

2 Related Work **6**

3 Problem Model **10**
- 3.1 Problem Definition 10
- 3.2 Preliminaries and Notations 11
- 3.3 The r-Gathering Problem 12

4 A Constant Approximation Algorithm **13**
- 4.1 Soft-ConFL Algorithm 13

5 Experimental Results **24**
- 5.1 Google Data Centers 24
- 5.2 Alternative Algorithms for Comparison 25

6 Conclusion and Future Work **31**

List of Figures

5.1 Google data centers world wide 25
5.2 Total Cost as function of Facility Opening Cost 26
5.3 Total Cost as function of Update Rate 27
5.4 Soft-ConFL algorithm vs. Greedy algorithm and Soft-FLP (facility cost of 5,000) 28
5.5 Soft-ConFL algorithm vs. Greedy algorithm and Soft-FLP (facility cost of 3,000) 29
5.6 Soft-ConFL algorithm vs. Greedy algorithm (facility cost of 5,000 and 10,000) 29
5.7 Soft-ConFL results for different update rates (facility cost of 1,000) 30

Abstract

In recent years, companies such as eBay, Facebook, Google, Microsoft, and Yahoo! have made large investments in massive data centers supporting cloud services. These data centers are becoming the hosting platform for a wide spectrum of composite applications with an increasing trend towards more communication intensive applications. As a result, the bandwidth usage within and between data centers is rapidly growing.

Data centers placement is a challenging set of optimization problems where the goal is to optimally place the applications and their related data over the available infrastructure. Unlike traditional facility location problems, in our case data is continuously updated, and the cost associated with this update increases with the number of data replica and the network distance between them.

We model this problem as a soft-capacitated connected facility location problem, which is NP-Hard in the general case. We present the first deterministic constant approximation algorithm for this problem and show, using extensive simulations and realistic data center and network topology, that our algorithm provides practically good placement decisions.

Abbreviations and Notations

CDN	—	Content Distribution Network
$CFLP$	—	Capacitated Facility Location Problem
$ConFL$	—	Connected Facility Location Problem
FLP	—	Facility Location Problem
$PCConFL$	—	Prize Collecting Connected Facility Location Problem
RoB	—	Rent or Buy
$SaaS$	—	Software as a Service
$Soft-ConFL$	—	Soft-Capacitated Connected Facility Location Problem
SPP	—	SaaS Placement Problem
$UFLP$	—	Uncapacitated Facility Location Problem
VNS	—	Variable Neighborhood Search

Chapter 1

Introduction

Data centers and cloud services are continuing to grow rapidly, with ever more functionality and ever more users around the globe [1]. Because of this growth, major cloud service providers now use tens of geographically dispersed data centers, and they keep on building more. These data centers are becoming the hosting platform for a wide spectrum of composite applications. Companies such as eBay, Facebook, Google, Microsoft, and Yahoo! are reported to have made large investments in building massive data centers supporting cloud services. In particular, there is an increasing trend towards more communication intensive applications in data centers, and thus bandwidth usage is rapidly growing both inside and between data centers [2].

Replica placement across data centers is a very common approach for improving performance and availability of content services. Content replication algorithms deploy a set of servers, distributed throughout the data center's network, and replicate the relevant data across these servers. Both the time required to access the data and network traffic is reduced by redirecting the application's requests to a nearby replica. The deployment of multiple replica servers also achieves content servers' redundancy and improves the availability of the system. In the past, replica placement algorithms were mainly considered in the context of web pages [35], but the increasing popularity of cloud services and media streaming [36] introduce a new domain that can benefit from this concept. Content replication has been a topic of extensive research, with a recent focus on CDNs (Content Distribution Networks) [3].

One can divide the research problems associated with replica placement across data centers into the How, What and Where aspects. How to direct the client's requests to the proper replica server having the desired content (the request routing system), what content should be distributed to the replica servers across the network (the content selection aspect) and where to place the replica servers throughout the network while keeping them up to date.

In this work, we consider the problem of placing replicas of an object at multiple locations in the network. Consider an email application which depends on an authentication service. We focus on the problem of placing replicas of an object (e.g., the authentication service) at multiple locations. Replica placement deals with the actual number and network location of the replicas. Clearly, we would like to minimize the network distance between an email application and the closest replica containing the desired content (in this example the authentication server) and thus having more replicas helps. On the other hand, having more replicas is more expensive so we need to model the cost and the benefit in a way that can allow to make the appropriate decisions regarding the number and the network locations of the replicas. This problem is strongly related to a family of optimization problems generally referred to as *facility location* problems [25].

Most of the existing algorithms neglects the cost of keeping the replicas across the network up to date, and in cases where this cost is non-neglectable this may lead to suboptimal realistic solutions. A replica must be synchronized with the original content server in order to supply reliable and precise service to the client requests. The amount of synchronization traffic across the network depends on the number of replicas deployed in the network, the topology of the distributed update and the rate of updates in the content of the server. It is also important to include in the model the fact that the capacity in terms of the number of concurrent customers for each replica is bounded. Thus, we extend our model by adding capacity constraint for each replica. We consider a soft version of the capacity constraint, where capacity u_i is associated with each replica i, but we are allowed to place several copies in the same location, which means that if we want this replica to serve x units of client demand, we have to open $\lceil \frac{x}{u_i} \rceil$ copies.

Our work considers a network design problem that combines a facility location and connectivity problem. In a vigorous content world of dynamic and interactive services, the replica's content must be synchronized and up to date, and the update process of the replica hosting servers may be a significant factor of the network traffic load. We assume that the hosting servers are updated simultaneously using a minimal Steiner tree. The problem is to choose the best locations for the replicas (or hosting servers) among the potential sites. We model the scenario above as a Soft-Capacitated Connected Facility Location Problem which is NP-Hard in the general case. We assume that each client uses a single replica (of course, multiple clients can use the same replica). In other words, a client gets all of its content from the same replica. In this work, we present a deterministic constant approximation algorithm to the soft capacitated connected facility location problem. The algorithm uses deterministic constant approximation algorithms for the uncapacitated facility location problem and for the minimum Steiner tree problem. Our algorithm constructs a related facility location problem by modifying both opening cost of facilities and the distance function. Then it runs the standard algorithms (which neglects the update costs and the capacity constraints) on the modified problem to receive a solution to the original problem. We prove that this method yields a deterministic constant approximation to the problem. To the best of our knowledge, this work is the first to present a deterministic constant approximation algorithm for the soft capacitated connected facility location problem.

We evaluate the performance of our proposed algorithm through extensive simulation experiments using a realistic data center network topology. We compare our scheme against existing alternatives (a greedy algorithm and a local-search facility location algorithm), on publicly available data regarding the Google data center network. The results indicate that our new algorithm preforms better than the currently available algorithms, and this improvement can be significant at certain settings.

Chapter 2

Related Work

Cloud computing is an emerging computing paradigm in which applications, data, and IT resources are provided as a service to users over the Internet. One kind of services that can be offered through the Cloud is Software as a Service or SaaS [16] [17]. Although SaaS can be delivered without using the Cloud computing infrastructure, with increasing demands for SaaS each year [18], SaaS vendors have to find a solution to cope with these increasing requests. A SaaS deployed in a Cloud is usually composed of several components, where each of the components represents a business function of the SaaS that is being delivered [19]. Some of these components may depend on other components, and some of the components may need to access data files that are located in Cloud storage servers. As we know, Cloud providers have many clusters of servers located across the globe. The problem of placing the components of a SaaS and their related data in the Cloud is referred to as SaaS Placement Problem (SPP). Kwok and Mohindra [20] presented a placement that is concerned with SaaS. The authors consider a placement problem for SaaS components in a multi-tenant architecture. The placement of the components is made within a set of available servers, and the main objective is to optimize the resource usage in each server. The principal rule used by the placement approach is rather straightforward, that is, a new instance of a component should be deployed in a server with the smallest residual resource left after having the instance. This is to reserve servers that have larger residual resources for instances that have a higher resource demand. That work also proposes a resource computation model to calculate the resource demand for an upcoming tenant, including the storage requirement. Although concerned with SaaS placement, this work is more focused on the multi-tenant resource model and does not take into consideration the SaaS's data placement in the network. Yusoh and Maolin [21] investigate the placement of applications and their related data within cloud environments. A penalty based genetic algorithm is proposed as a solution to the placement problem. The solution aims at placing the processing algorithm on a compute node that has a better bandwidth value with respect to the storage node. The results of experiments show that it is both a feasible and scalable solution, since the size of the cloud and the number of components increases the computation time increases in a near linear manner.

As explained in the previous chapter, in this work we consider a network design problem that combines both facility location and connectivity problems. The Connected Facility Location Problem has a wide range of applications and has recently received considerable attention both in the theoretical computer science literature and in the operations research literature. However, compared to some closely related problem classes, there is just a small number of papers on the topic [14]. The majority of publication about ConFL comes from the computer science community which presents approximation algorithms of different kinds and qualities. The operations research community has developed a small number of heuristic methods.

Karger and Minkoff [5] introduced the so-called maybecast problem which is a probabilistic version of the Steiner tree problem. They consider the distribution of single data items from a root to a set of clients. Each client is activated and request data independently with a known probability. When a client is activated, all edges on its path to the root become active. The goal is to find a Steiner tree that minimizes the expected cost of the active edges. The problem of finding a tree with minimal expected cost can be modeled as a connected facility location problem. The authors described a constant factor approximation for their problem. The name connected facility location has been introduced by Gupta et al. [8] in their work on virtual private networks. They prooved ConFL is NP-hard and improved the previous result by introducing a 10.66 factor approximation algorithm. Their algorithm was based on a linear programming (LP) rounding technique. Since then several authors proposed approximation algorithms for diverse variants of ConFL. Swamy and Kumar [9] presented the first primal-dual algorithm with an approximation ratio of 9 and later the same authors [10] improved the approximation factor to 8.55. They also developed a primal-dual approximation algorithms for the so-called rent-or-buy problem (RoB), a variant of ConFL, in which all nodes are potential facilities with no opening costs and for the k-ConFL, a variant of ConFL, which comprises the additional restriction that in an optimum solution at most k facilities can be opened. The authors gave approximation ratios of 4.55 and 15.55 for RoB and k-ConFL, respectively. The approximation factors have been successively improved in Jung et al [11]. and Williamson and Zuylen [12]. Recently, Eisenbrand et al. [13] combined approximation algorithms for the basic facility location problem and the connectivity problem of the opened facilities by running a randomized approximation algorithm with an expected approximation ratio of 4 for ConFL, 2.92 for RoB and 6.85 for k-ConFL. The ratios can be derandomized with a resulting approximation factors of 4.23, 3.28 and 6.98 respectively.

The capacitated facility location problem and variations of it have been well-studied in the literature (see, for example,the book of Mirchandani and Francis [23]) and arise in practice (see, for example, the paper of Barahona and Jensen [22] for an instance of a parts warehousing problem from IBM). There are two main variants of the problem, depending on whether a facility can be opened at most once (hard-capacitated) or may be opened k times at a cost of $k \cdot f_i$ (soft-capacitated). We consider the soft-capacitated facility location variant which is also known as facility location problem with integer decision variables in the operations research literature [24]. Shmoys, Tardos and Aardal [25] gave the first approximation algorithms with constant performance guarantee for a number of metric facility location problems. For the soft-capacitated facility location problem they gave a 5.69 approximation algorithm. Chudak and Shmoys [26] gave a 3 approximation algorithm for a variant of the problem with uniform capacities using LP rounding. For non-uniform capacities, Jain and Vazirani [27] showed how to reduce this problem to the uncapacitated facility location problem, and presented a primal-dual algorithm with approximation ratio of 4. A local search algorithm proposed by Arya et al [28] had an approximation ratio of 3.72. This result was further improved by Mahdian, Ye, and Zhang [29] who gave a 2.89-approximation algorithm, and later the same authors [30] developed the current best approximation ratio, a 2-approximation algorithm.

Leitner and Raidl [31] introduced the first variant of capacitated connected facility location problem. The authors considered a connected facility location problem where only clients which are reasonable from an economic point of view need to be served (prize collecting variant) with capacity constraints on possible facilities (PCConFL). They presented two Variable Neighborhood Search (VNS) approaches for a variant of PCConFL without assignment and opening costs. Subsequently, the same authors [32] [33] [34] proposed a Lagrangian relaxation based approach which has been hybridized with local search and very large scale neighborhood search as well as two mixed integer programming models based on multi-commodity flows.

Recently, Eisenbrand et al. [15] presented the connected soft-capacitated facility location problem (soft-ConFL). Using the same techniques as in [13], the authors gave a randomized approximation algorithm with an expected approximation ratio of 6.27. To the best of our knowledge, this work is the first to present a deterministic constant approximation algorithm for the soft capacitated connected facility location problem.

Chapter 3

Problem Model

In this chapter we describe the *Soft-Capacitated Connected Facility Location Problem*, introduce several notations and definitions that are useful for the analysis of our problem and define the r-gathering problem which will be used to prove our main results.

3.1 Problem Definition

We are given an undirected graph $G = (V, E)$ with non-negative symmetric costs $c_{i,j}$ $(i, j \in V)$ on the edges. Let $F \subseteq V$ be a set of locations where facilities may be placed and $C \subseteq V$ be a set of demand nodes or clients that must be assigned to an open facility. Client $j \in C$ requires a non negative d_j units of demand, and facility $i \in F$ has a non negative opening cost f_i and can serve at most u_i units of demand. We focus on the soft capacitated problem in which multiple facilities can be built at a single location. The goal is to find a set $S \subseteq F$ of open facilities, a feasible assignment $\sigma : C \to S$ of clients in C to open facilities in S and a connected subgraph $R = (V_R, E_R)$ spanning S, so as to minimize the total cost:

$$Cost\left(S, \sigma, R\right) = M \cdot ST\left(R\right) + T\left(\sigma\right) + C_f\left(S\right)$$

where:

1. M is a rate update parameter,

2. $ST(R) = \sum_{i,j \in E_R} c_{i,j}$ is the edge cost of the graph R[1],

3. $T(\sigma) = \sum_{j \in C} d_j \cdot c_{j,\sigma(j)}$ is the cost of serving each client j from its assigned open facility $i \in S$, and

4. $C_f(S) = \sum_{i \in S} y_i \cdot f_i$ is the opening cost of the facilities, where y_i the number of facilities opened at location $i \in S$.

The capacity constraint implies that $\sum_{j \in C|\sigma(j)=i} d_j \leq u_i \cdot y_i$ for each i. Note that in the text we abuse notations and use the term facility f_i to describe the set of facilities opened in location i.

3.2 Preliminaries and Notations

We start with introducing several notations and definitions that are useful for the analysis of the soft capacitated connected facility location problem. Given a set $S \subseteq F$ of open facilities and an assignment σ:

- $L_i(\sigma)$ is the set of clients assigned to facility $i \in S$ under assignment σ:

$$L_i(\sigma) = \{j \in C | \sigma(j) = i\}.$$

- $D_i(\sigma)$ is the total amount of demand assigned to facility $i \in S$ under assignment σ:

$$D_i(\sigma) = \sum_{j \in L_i(\sigma)} d_j.$$

- D is the total demand of clients in the network:

$$D = \sum_{j \in C} d_j.$$

We say that σ is λ-loaded if it satisfies:

$$D_i(\sigma) \geq \lambda \text{ for all } i \in S.$$

That is, every location with at least one open facility serves at least λ units of demand under assignment σ.

[1]The minimum cost connected subgraph spanning S is a minimum Steiner tree.

3.3 The r-Gathering Problem

One of the interesting recent variants of Metric Facility Location is the r-gathering problem, introduced simultaneously by Karger and Minkoff [5] and by Guha et al. [4] (where it is called load-balanced facility-location). The basic additional requirement in the r-gathering problem is that each facility will be assigned at least r customers (customers are not necessarily assigned to the nearest open facility in the capacitated facility location problems). This variant captures the idea that opening a facility is economically justified only when it serves at least a certain amount of demand (and this constraint may even be more natural than facility costs in some settings).

Both papers ([5] and [4]) considered a generalization of r-gathering in which customers have different demands, the connection-costs are the product of the demand and distance, and each facility must serve customers having a total of at least r demand. They both presented a $(\frac{1+\alpha}{1-\alpha}\beta, \alpha)$ bicriteria approximation, for any $\alpha < 1$, where β is the approximation-ratio of the metric facility location problem (currently 1.5 for the classic problem [6], and 1.582 for the generalization in which customers may have different demands [7]). Namely, their algorithms guarantee that each open facility in the solution will serve at least αr demand, and the cost will be at most $(\frac{1+\alpha}{1-\alpha}\beta)$ times the optimal cost of the r-gathering problem.

In our theoretical analysis, we introduce the λ-loaded facility location problem. The problem is a special type of the demand-weighted facility location problems in which we require that each open facility has at least λ units of demand assigned to it. Throughout the analysis of our problem, we use Karger and Minkoff [5] and Guha et al. [4] bicriteria approximation. Note that we present a deterministic constant approximation to the problem despite using this bicriteria approximation algorithm.

Chapter 4

A Constant Approximation Algorithm

In this chapter we present our deterministic constant approximation algorithm for the soft capacitated connected facility location problem. To the best of our knowledge, this is the first deterministic constant approximation algorithm for this problem.

Throughout this chapter we assume that the total demand of clients is larger than the update rate parameter:

$$M \leq D.$$

If this is not the case, the optimal solution is trivial: Open only one facility (with minimum cost for satisfying total clients demands). It is straight forward to show that this yields a 2 approximation algorithm.

4.1 Soft-ConFL Algorithm

In this section, we present the Soft-ConFL algorithm. Given any ρ-approximation algorithm for the uncapacitated facility location problem and any φ-approximation algorithm for the minimum Steiner tree problem, Soft-ConFL algorithm yields a $(24\rho+2\varphi+48\rho\varphi)$-approximation algorithm for the soft capacitated connected facility location problem.

Given an original soft capacitated connected facility location problem, we construct a related facility location problem by modifying both the opening cost of facilities and the distance function. First, we add a cost λ_i to each facility $i \in F$. This cost is defined as twice the minimum cost of stisfying M units of demand from facility i. More formally, let $j_1, j_2, \cdots j_n$ be the clients, ordered in increasing distance from facility i. Let k be the minimum number such that

$$d_{j_1} + d_{j_2} + \cdots + d_{j_k} \geq M.$$

For simplicity let us assume this sum is exactly equal to M (we can always split client k into two smaller demands). The opening cost of facility i is set to $f'_i = f_i + \lambda_i$, where λ_i is defined to be

$$\lambda_i = 2(c_{ij_1} d_{j_1} + c_{ij_2} d_{j_2} + \cdots + c_{ij_k} d_{j_k}).$$

Second, We modify the distance function to be:

$$c'_{i,j} = c_{i,j} + max(\frac{f'_i}{u_i}, \frac{f'_j}{u_j}),$$

where for any $i \notin F$, we set $\frac{f_i}{u_i}$ to be 0.

Now we run any given ρ-approximation algorithm for the uncapacitated facility location problem on the modified problem. We modify the output of this approximation algorithm by closing any facility with $D_i(\sigma) \leq \frac{M}{2}$ and assigning every client to the nearest open facility (this may require opening additional copies in this site). We run any given φ-approximation algorithm for the minimum Steiner tree problem on the set of opened facilities S, and this is the output of the Soft-ConFL algorithm.

Our main result is the following theorem that states that *Soft-ConFL Algorithm* indeed yields a constant approximation solutions.

Theorem 1 *Given a ρ-approximation algorithm for the uncapacitated facility location problem and a φ-approximation algorithm for the minimum Steiner tree problem, one can find a solution to the soft capacitated connected facility location problem, with a cost of at most $(24\rho + 2\varphi + 48\rho\varphi)$ times the optimum cost.*

In order to prove the theorem we prove the following four lemmas. The first lemma is a technical lemma stating that any solution can be converted into an M-loaded solution without increasing the cost too much.

Lemma 1 *For any given feasible solution to the soft capacitated connected facility location problem with update parameter $M \geq D$ $SOL = (S, \sigma, R)$, one can find a feasible $M - loaded$ solution, $SOL' = (S', \sigma', R')$, such that*

$$Cost(SOL') \leq 4 \cdot Cost(SOL).$$

Proof

We are given a feasible solution to the soft capacitated connected facility location problem $SOL = (S, \sigma, R)$ with update parameter $M \geq D$. We can assume that $R = \{V_R, E_R\}$ is a tree, otherwise one can find a lower cost sub-tree (with no cycles) of R which spans V_R. The following procedure iteratively converts the solution SOL into an $M - loaded$ solution SOL'.

Throughout the execution of the procedure, we differentiate between two types of facilities: *M-loaded facilities* which are open and have at least M units of demand assigned to them, and *Unloaded facilities* which are open but have less than M units of demand assigned to them. $S_{M-loaded}$ will denote the set of opened facilities which are M-loaded for the current solution; that is,

$$S_{M-loaded} = \{i \in S; D_i(\sigma) \geq M\}.$$

We also differentiate between two types of clients: *Co-op clients* which are assigned to *M-loaded facilities* and *Free clients* which are assigned to *Unloaded facilities*. We define $TotalFree(n, \sigma)$ to be the total demand assigned to unloaded facilities under assignment σ in sub-tree n; that is,

$$TotalFree(n, \sigma) = \sum_{i \in n \cap S_{M-loaded}} D_i.$$

Claim: Given a tree rooted at n with $M \leq TotalFree(n, \sigma)$, one can find a sub-tree n' of n with $M \leq TotalFree(n', \sigma) \leq 3M$, such that n' has no subtree with M or more free demand.

Proof: The proof is obtained by induction on the tree height (h). Note that a single node has less than M free demand (if it has more than M units of demand assigned then it is considered as M-loaded facility with 0 free demand). For the base case, $h = 1$, there is only one complete binary tree of height one. It has one node that is the root and 2 leafs. Since any single node has less than M free demand the total free demand of the tree is $\leq 3M$. Thus for $h = 1$, $M \leq TotalFree(n, \sigma) \Rightarrow M \leq TotalFree(n, \sigma) \leq 3M$. For the induction step, assume that the claim is true for all binary trees of height $\leq h$. We will prove that the claim is true for all binary trees of height $h+1$. Consider a complete binary tree of height $h+1>1$. This tree, T, has a root r that is a non-leaf and hence has 2 children r_L and r_R that are roots of binary trees of height h or less. We consider two options: at least one sub-tree (without loss of generality r_L) has $M \leq TotalFree(r_L, \sigma)$ or both sub-trees have $TotalFree(r_L, \sigma) < M$. In the first case, by the inductive hypothesis, one can find sub-tree r' with $M \leq TotalFree(r', \sigma) \leq 3M$. In the second case we have a root r that by definition has less than M free demand and two sub-trees with less than M free demand assigned to each and thus $TotalFree(r, \sigma) \leq 3M$.

The main procedure, $FindMLoadedSolution(S, \sigma, R)$ is described below. It gets as an input a feasible solution $SOL = (S, \sigma, R)$ to the soft capacitated connected facility and returns an M-loaded solution. As mentioned before, we assume $M \leq D$, otherwise there exist no feasible M-loaded solution. The procedure first convert R into a binary tree rooted at r (the general tree is converted into a binary tree by introducing dummy nodes). We set S' to hold the set of opened M-loaded facilities. Throughout the execution we add new M-loaded facilities to S'. In each iteration of the algorithm we look for a sub-tree with $M \leq TotalFree(n', \sigma) \leq 3M$, according to our claim.

Algorithm 4.1.1: FINDMLOADEDSOLUTION(S, σ, R)

comment: Returns an M-loaded solution

$T \leftarrow Binary\ spanning\ tree\ of\ R\ rooted\ at\ r$
$S' \leftarrow S_{M-loaded}$
$\sigma \leftarrow \sigma'$
while FINDMINIMALFREETREE$(r, \sigma') \neq NULL$
do $\begin{cases} n \leftarrow \text{FINDMINIMALFREETREE}(r, \sigma') \\ l \leftarrow the\ minimal\ cost\ facility\ in\ n\ which\ is\ not\ M_loaded \\ S' = S' \cup l \\ assign\ free\ demands\ to\ l \\ update\ \sigma'\ with\ the\ new\ assignment \end{cases}$
$assign\ remaining\ free\ demands\ to\ the\ minimal\ cost\ facility\ in\ S'$
return (S', σ', R')

The algorithm iteratively converts any given solution into an *M-loaded* solution without increasing the total cost by more than a factor of four. Throughout the execution of the algorithm, it maintains a feasible solution where the demand is assigned to open facilities. The following properties hold:

(P1) $ST(R') \leq ST(R)$

(P2) $C_f(S') \leq C_f(S)$

(P3) $T(\sigma') \leq T(\sigma) + 3M \cdot ST(R)$

These properties certainly hold when the algorithm starts. Furthermore, the fact that they hold when the algorithm stops, proves Lemma 1. Property (P1) clearly holds since $S' \subseteq S$. Property (P2) is maintained by the algorithm in each iteration, since we assign clients' demand to the minimal cost facility in each subtree. To show that property (P3) is maintained, first observe that in each iterations any *M-loaded* subtree (with no free demands) has no change in its facilities' assignments. Each iteration reassign free demand units to an open facility which is not *M-loaded* or part of an *M-loaded* subtree. Thus, the procedure will modify any subtree at most one time. Second, in each iteration (including the final iteration) we reassign maximum $3M$ demand units in subtree n therefore, we increase the cost of assigning each client by at most $3M$ times the cost of subtree n. Since we modify each subtree only once, we increase the total assignment cost by at most $3M \cdot ST(R)$. □

The second lemma is used to show that the *Soft-ConFL Algorithm* yields a constant approximation $\frac{M}{2}$-loaded solution using any ρ-approximation algorithm for the uncapacitated facility location problem.

Lemma 2 *Given a ρ-approximation algorithm for the uncapacitated facility location problem and and a parameter λ, one can find a $\frac{\lambda}{2} -$ loaded solution of cost at most 3ρ times the optimum $\lambda -$ loaded facility location cost.*

Proof

We start by defining a modified facility location problem and show that its solution meets our requirements. To define this new facility location problem, we add a cost λ_i to each facility $i \in F$. This cost is defined as twice the minimum cost of moving λ units of demand from the clients to facility i. Let $j_1, j_2, \cdots j_n$ be the clients, ordered in increasing distance from facility i. Let k be the minimum number such that

$$d_{j_1} + d_{j_2} + \cdots + d_{j_k} \geq \lambda.$$

For simplicity let us assume this sum is exactly equal to λ (we can always split client k into two smaller demands). The opening cost of facility i is $f_i + \lambda_i$, where λ_i is defined to be

$$\lambda_i = 2(c_{ij_1}d_{j_1} + c_{ij_2}d_{j_2} + \cdots + c_{ij_k}d_{j_k}).$$

Note that if $SOL = (S, \sigma, R)$ is a $\lambda - loaded$ solution, we get

$$\sum_{j \in L_i(\sigma)} c_{ij}d_j \geq \tfrac{\lambda_i}{2}.$$

First, we show that any solution SOL of the $\lambda - loaded$ facility location problem is a feasible solution to the modified problem defined above with cost that is at most a factor of three. The traffic cost ($T(\sigma)$) and the basic cost of opening facilities ($C_f = \sum_{i \in S} f_i$) are the same for both problems, thus we only need to consider the added cost of opening facilities ($\sum_{i \in S} \lambda_i$). Since SOL is $\lambda - loaded$, we get:

$$\sum_{i \in S} \lambda_i \leq 2 \cdot \sum_{i \in S} \sum_{j \in L_i(\sigma^S)} c_{ij}d_j = 2 \cdot T(\sigma).$$

Second, we show that any solution of the modified facility location problem can be converted into a $\frac{\lambda}{2} - loaded$ solution with equal or smaller cost. Showing that closing a facility i, with $D_i(\sigma) \leq \frac{\lambda}{2}$, only reduces the total cost.

As a first step, assign every client to the nearest open facility. Clearly this step can only reduce total cost. Assume that there exists an open facility i with less than $\frac{\lambda}{2}$ assigned demand, and let $j_1, j_2, \cdots j_k$ be the clients that are used to define λ_i. Define

$$c = \tfrac{1}{\lambda}(c_{ij_1}d_{j_1} + c_{ij_2}d_{j_2} + \cdots + c_{ij_k}d_{j_k}) = \tfrac{\lambda_i}{2\lambda}.$$

We claim that there exists a client j', not assigned to i in the given solution at distance at most $2c$ from i, since otherwise:

$$c = \frac{1}{\lambda}(c_{ij_1}d_{j_1} + c_{ij_2}d_{j_2} + \cdots + c_{ij_k}d_{j_k}) = \frac{1}{\lambda}(\sum_{j_l \in L_i(\sigma)} c_{ij_l}d_{j_l} + \sum_{j_l \notin L_i(\sigma)} c_{ij_l}d_{j_l}) \geq \frac{1}{\lambda}(\sum_{j_l \notin L_i(\sigma)} c_{ij_l}d_{j_l})$$

Since we assumed that $\forall j \notin L_i(\sigma) \Rightarrow c_{ij} > 2c$, we get:

$$\frac{1}{\lambda}(\sum_{j_l \notin L_i(\sigma)} c_{ij_l}d_{j_l}) > \frac{1}{\lambda}(\sum_{j_l \notin L_i(\sigma)} 2cd_{j_l}).$$

By the assumption, less than $\frac{\lambda}{2}$ of the nearby demand is assigned to i, thus

$$\frac{1}{\lambda}(\sum_{j_l \notin L_i(\sigma)} 2cd_{j_l}) > \frac{1}{\lambda}(2c \cdot \frac{\lambda}{2}) = c$$

$$\Downarrow$$

$$c < c \text{ contradiction}$$

Since each client is assigned to the nearest open facility, this j' must be assigned to a facility i' at distance at most $2c$. By the triangle inequality, this facility is at distance less than $4c$ from i. Suppose we close facility i and assign its clients to i'. This step adds at most $4c$ to the assignment distance of i's clients, thus the total increase in traffic cost is the amount of demand (less than $\frac{\lambda}{2}$) times the added distance (at most $4c$), a total of less than $2\lambda c$. Since $c = \frac{\lambda_i}{2\lambda}$, the change in assignment cost is less than λ_i.

Now, given the facility location problem and the parameter λ, define the modified facility location problem with the additional λ_i cost, and find a $\rho - approximation$ solution for it. We proved that the cost of this solution is at most 3ρ times the optimal $\lambda - loaded$ solution, and it can be converted with no additional cost into a $\dfrac{\lambda}{2} - loaded$ solution. □

The next lemma is used to show that the cost of connecting a set of facilities which are M-loaded is bounded.

Lemma 3 *Let $SOL' = (S', \sigma', R')$ be any solution for the soft capacitated connected facility location problem and let $SOL = (S, \sigma, R)$ be any $\lambda - loaded$ solution, then*

$$ST(R) \leq \frac{1}{\lambda}(T(\sigma) + T(\sigma')) + ST(R')$$

Proof

The proof of this lemma is quite simple. We show that for each facility i in S we can find a path of cost p_i to a facility in S', such that:

$$\sum_{i \in S} p_i \leq \frac{1}{\lambda}(T(\sigma) + T(\sigma')).$$

We choose p_i to be

$$p_i = min_{j \in L_i(\sigma)}(c_{i,j} + c_{j,\sigma'(j)}).$$

By definition:

$$T(\sigma) = \sum_{i \in S} \sum_{j \in L_i(\sigma)} d_j \cdot c_{i,j}$$

and

$$T(\sigma') = \sum_{j \in C} d_j \cdot c_{j,\sigma'(j)} = \sum_{i \in S} \sum_{j \in L_i(\sigma)} d_j \cdot c_{j,\sigma'(j)}$$

Therefore

$$T(\sigma) + T(\sigma') = \sum_{i \in S} \sum_{j \in L_i(\sigma)} d_j \cdot (c_{i,j} + c_{j,\sigma'(j)}) \geq \sum_{i \in S} \sum_{j \in L_i(\sigma)} d_j \cdot p_i$$

since SOL is $\lambda - loaded$, we have

$$T(\sigma) + T(\sigma') \geq \sum_{i \in S} \lambda \cdot p_i$$
$$\Downarrow$$
$$\sum_{i \in S} p_i \leq \frac{1}{\lambda}(T(\sigma) + T(\sigma'))$$

The union of the Steiner tree which spans S' and all paths between S to S' spans S. Thus:

$$ST(R') + \sum_{i \in S} p_i \geq ST(R)$$
$$\Downarrow$$
$$ST(R') + \frac{1}{\lambda}(T(\sigma) + T(\sigma')) \geq ST(R)$$

$\square$

The last lemma is used to show that *Soft-ConFL Algorithm* yields a soft-capacitated solution with constant approximation.

Lemma 4 *Given a ρ-approximation algorithm for the $\lambda - loaded$ uncapacitated facility location problem, one can get a 2ρ-approximation for the $\lambda - loaded$ soft capacitated facility location version.*

Proof

Given an instance I of the $\lambda - loaded$ soft capacitated facility location, we construct an instance I' of the $\lambda - loaded$ uncapacitated facility location with the following modified distance function

$$c'_{i,j} = c_{i,j} + max(\frac{f_i}{u_i}, \frac{f_j}{u_j}),$$

where for any $i \notin F$, we set $\frac{f_i}{u_i}$ to be 0. It can be easily verified that the modified distance function satisfy the metric conditions. The facility costs and client demands in the new instance are the same as the facility costs f_i and client demands d_j in the soft capacitated instance.

First we show that given a solution $SOL = (S, \sigma)$ to I with assignment cost $T(\sigma)$ and facility cost $C_f(S)$, there exist a solution for I' with an assignment cost of at most $T(\sigma) + C_f(S)$ and a facility cost of at most $C_f(S)$. Let y_i be the number of facilities built at i in SOL. Let $x_{i,j}$ be an indicator variable that is 1 iff client j is assigned to facility i under assignment σ. $T(\sigma) = \sum_{i,j} x_{i,j} \cdot c_{i,j} \cdot d_j$ and $C_f(S) = \sum_i y_i \cdot f_i$. The capacity constraint implies that $\sum_j x_{i,j} \cdot d_j \leq u_i \cdot y_i$. We construct a feasible solution $SOL' = (S', \sigma')$ of I' as follows: The assignments of nodes to facilities is the same as in σ (which assures that it would satisfy the $\lambda - loaded$ condition). A facility is built at i iff at least one facility is built at i in S. Clearly the opening cost of the facilities $C_f(S')$ is at most $C_f(S)$ and the assignment cost $T(S')$

$$\begin{aligned}
\sum_{i,j} x_{i,j} d_j c'_{i,j} &= \sum_{i,j} x_{i,j} d_j (c_{i,j} + \frac{f_i}{u_i}) = \\
&= \sum_{i,j} x_{i,j} d_j c_{i,j} + \sum_i f_i \frac{\sum_j x_{i,j} d_j}{u_i} \leq \\
&\leq \sum_{i,j} x_{i,j} d_j c_{i,j} + \sum_i f_i y_i \\
&= T(\sigma) + C_f(S).
\end{aligned}$$

Note that this implies that $OPT(I') \leq 2 \cdot OPT(I)$.

Second we show that given a feasible solution $SOL' = (S', \sigma')$ to I', there exists a feasible solution $SOL = (S, \sigma)$ to I such that $cost(SOL) \leq cost(SOL')$. Let Y_i be an indicator variable that is 1 iff a facility is built at i

in SOL'. Let $x_{i,j}$ be an indicator variable that is 1 iff client j is assigned to facility i in SOL'. The solution SOL for instance I is obtained as follows: The assignments of nodes to facilities is the same as in σ' (which assures that it would satisfy the $\lambda - loaded$ condition). The number of facilities built at i is

$$y_i = \lceil \frac{\sum_j x_{i,j} \cdot d_j}{u_i} \rceil.$$

Thus, $y_i \leq \frac{\sum_j x_{i,j} \cdot d_j}{u_i} + Y_i$. The cost of SOL' is

$$\begin{aligned}\sum_{i,j} x_{i,j} d_j c'_{i,j} + \sum_i f_i Y_i &= \sum_{i,j} x_{i,j} d_j (c_{i,j} + \frac{f_i}{u_i}) + \sum_i f_i Y_i \\ &= \sum_{i,j} x_{i,j} d_j c_{i,j} + \sum_i f_i (\frac{\sum_j x_{i,j} \cdot d_j}{u_i} + Y_i) \\ &\geq \sum_{i,j} x_{i,j} d_j c_{i,j} + \sum_i f_i y_i = cost(SOL).\end{aligned}$$

Since we have a ρ approximation algorithm for the $\lambda - loaded$ uncapacitated facility location problem, we can obtain a solution for I' with a cost of at most $\rho \cdot OPT(I')$, and since we proved $OPT(I') \leq 2 \cdot OPT(I)$ we get $\rho OPT(I') \leq 2\rho OPT(I)$. Now, we can obtain a solution to I of cost at most $2\rho OPT(I)$, yielding a 2ρ approximation for the $\lambda - loaded$ soft capacitated facility location problem. □

Proof of Theorem 1

Let $SOL^* = (S^*, \sigma^*, R^*)$ be the optimal solution of the soft capacitated connected facility location problem. Lemma 1 implies that there exists a M-loaded solution $SOL' = (S', \sigma', R')$ such that:

$$Cost(SOL') \leq 4 \cdot Cost(SOL^*).$$

Let $\widehat{SOL} = (\widehat{S}, \widehat{\sigma}, \widehat{R})$ be the optimal solution of the equivalent $M - loaded$ soft capacitated facility location problem which minimizes the total assignment cost and opening cost of facilities. Since $\widehat{SOL}$ is optimal solution of all feasible $M - loaded$ solution (including SOL') we get:

$$T(\widehat{\sigma}) + C_f(\widehat{S}) \leq T(\sigma') + C_f(S') \leq Cost(SOL') \leq 4 \cdot Cost(SOL^*).$$

By Lemma 2 and Lemma 4 we get that given a $\rho-approximation$ algorithm for the uncapacitated facility location problem we can find a $\frac{M}{2}-loaded$ solution $SOL = (S, \sigma, R)$ that satisfies:

$$T(\sigma) + C_f(S) \leq 6\rho(T(\widehat{\sigma}) + C_f(\widehat{S})) \leq 24\rho(Cost(SOL^*)).$$

By Lemma 3 and since SOL is a $\frac{M}{2}-loaded$ solution, we get:

$$ST(R) \leq \frac{2}{M}(T(\sigma) + T(sigma^*)) + ST(R^*)$$
$$\Downarrow$$
$$M \cdot ST(R) \leq 2(T(\sigma) + T(\sigma^*)) + M \cdot ST(R^*) \leq 2 \cdot Cost(SOL^*) + 2 \cdot T(\sigma) \leq (2 + 48\rho)Cost(SOL^*)$$

Since the Minimum Steiner Tree Problem is APX-complete (Bern and Plassmann [3]), we use a $\varphi - approximation$ algorithm (Robins and Zelikovsky [4]), thus:

$$M \cdot ST(R) \leq (2 + 48\rho)\varphi \cdot Cost(S^*)$$

Combining the above, we obtain the following inequality:

$$Cost(SOL) = T(\sigma) + C_f(S) + M \cdot ST(R) \leq (24\rho + 2\varphi + 48\rho\varphi)Cost(SOL^*),$$

which proves Theorem 1. $\square$

Chapter 5

Experimental Results

In this chapter we evaluate the performance of our algorithm through extensive simulation experiments on realistic scenarios. We use available data about Google's 36-node data centers network, in order to compare our algorithm against existing alternatives. Unless stated otherwise, we assume unified demand for each node and unified cost for each data center. In our simulations, we assume that the network distances between the data centers are relative to the geographic distance between them.

5.1 Google Data Centers

Google's data infrastructure is massive and spread across the world, according to Data Center Knowledge (DCK) in 2008, there were 36 data centers all together (see Figure 5.1[1] on Page 25) - 19 in the U.S., 12 in Europe, 3 in Asia, one in Russia, and one in South America[2]. Not all of the locations are dedicated Google data centers, since it is claimed that they sometimes lease space in other companies' data centers[3]. In our simulations we used the map of locations which was constructed by Pingdom and Data Center Knowledge.

[1] http://techcrunch.com/2008/04/11/where-are-all-the-google-data-centers/
[2] http://www.datacenterknowledge.com/archives/2008/03/27/google-data-center-faq/
[3] http://royal.pingdom.com/2008/04/11/map-of-all-google-data-center-locations/

Figure 5.1: Google data centers world wide

5.2 Alternative Algorithms for Comparison

In order to evaluate the performance of the *Soft-ConFL Algorithm* we compared our results to two alternative algorithms:

- **Greedy Algorithm:** Iteratively the algorithm picks one facility f_i which minimizes the total cost and add it to set S. During the iterations, the algorithm holds the set with the minimum cost and returns it.

- **Soft Capacitated Facility Location Algorithm (Soft-FLP):** A local search iterative heuristic which starts with any feasible solution, and improves the quality of the solution iteratively. At each step, it considers only local operations to improve the cost of the solution. A solution is called a local minima if there is no local operations which improves the cost. The local search procedure that we consider permits *adding*, *dropping* and *swapping* a facility. Arya et al. [28] showed that such a procedure gives a 3 approximation algorithm in the case where update costs are not considered.

We evaluated the performance of the *Soft-ConFL Algorithm* in two different scenarios. In the first scenario, we evaluated the performance of the algorithm with different facility opening costs, investigating the algorithm

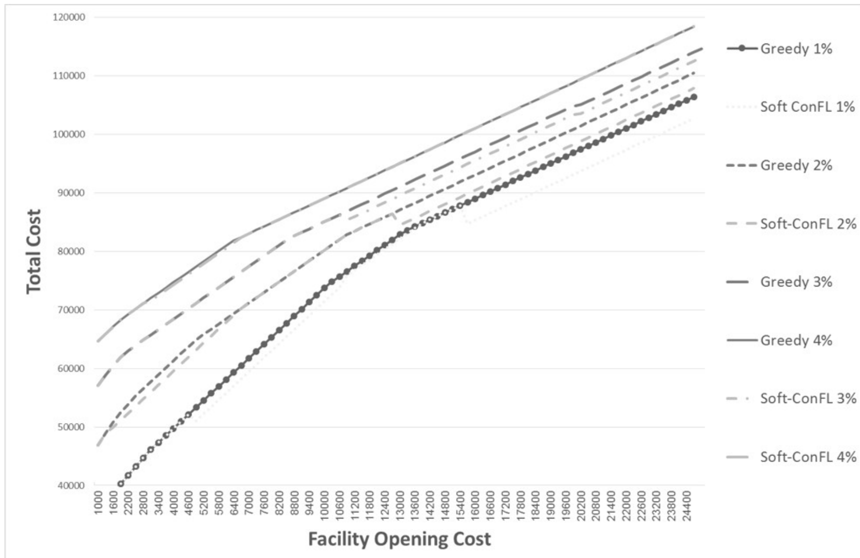

Figure 5.2: Total Cost as a function of Facility Opening Cost

sensitivity to different facility cost (1,000-25,000 units of cost per facility). Figure 5.2 depicts the total cost of the Soft-ConFL algorithm and the Greedy algorithm as a function of facility opening cost. We compared the algorithm performance at different update rate variables which are presented as percentage of the total demand ($1\% - 4\%$). The greedy algorithm yields inferior results which cost up to 7% more than the Soft-ConFL Algorithm.

In the second scenario, we evaluated the performance of the algorithm with different inputs of update rate variables. We've investigated the algorithm sensitivity to different update rates ($0\% - 5\%$ as percentage of the total demand). Figure 5.3 depicts the total cost of the Soft-ConFL algorithm and the Greedy algorithm as a function of update rate. We set different unified costs for opening a facility ($1,000 - 16,000$). The greedy algorithm yields inferior results which cost up to 7% more than the Soft-ConFL Algorithm.

For a proper comparison of our algorithm performance, we have used the same local search algorithm from [28] as the uncapacitated facility location algorithm which is used in Soft-ConFL.

Figure 5.4 depicts the results of the Soft-ConFL algorithm, the Greedy algorithm and the Soft-FLP. We set a unified cost of 5,000 units for each

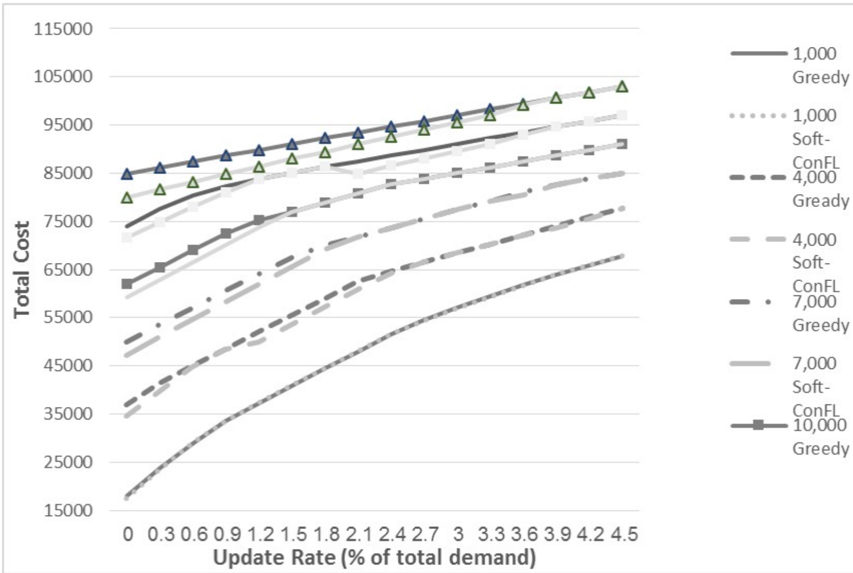

Figure 5.3: Total Cost as a function of Update Rate

location. Note that our cost function combines facility and network cost and the total cost of the minimum spanning tree of the data centers network is 22,000 units. We compared the algorithm performance at different update rates. The update rate variable is presented as percentage of the total demand[4]. One can observe that the higher the update rate, the performance of the Soft-FLP algorithm is worse, because it ignores the update cost of the facilities. The greedy algorithm yields inferior results for low update rates ($< 6\%$) because it opens a large set of facilities. However, for larger update rates ($> 7\%$) where it opens a small set of facilities it yields the same results as Soft-ConFL. Note that update rate of 6.4% is a turning point where the three algorithm yield similar results, the greedy and Soft-ConFL unify, and Soft-FLP starts to yields significant inferior results.

Figure 5.5 is similar to Figure 5.4 but the unified facility cost was set to a lower value of 3,000. As in the previous case, the local search algorithm (dashed line), which neglects the cost of keeping the replicas across the network up to date, yields sub-optimal results. We can see that the algorithms

[4]Note that the demand in our problem is in fact demand per time unit, and the ratio is dimensionless quantity.

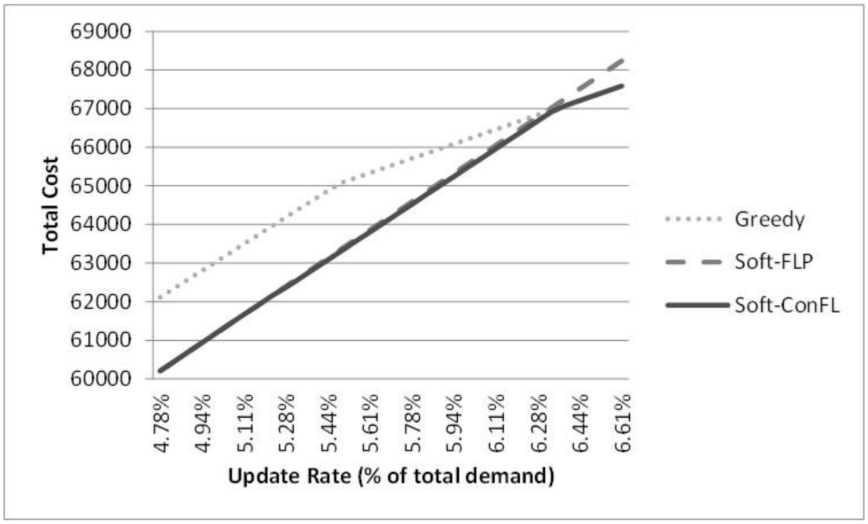

Figure 5.4: Soft-ConFL algorithm vs. Greedy algorithm and Soft-FLP (facility cost of 5,000)

that considers this factor (the greedy algorithm and Soft-ConFL) can lead to better solutions. Note that for low update rates ($< 3.5\%$) both the greedy algorithm and Soft-ConFL yield the same results, for a certain range of update rates ($3.5\% < M < 7.5\%$) the Soft-ConFL yields better results and for higher update rates the graphs of both algorithm reunites.

The Soft-FLP algorithm yields suboptimal results for high update rates, thus in Figure 5.6 we focus on comparing the greedy algorithm and Soft-ConFL. We present two experiments with different facility costs (5,000 and 10,000). Note that in both cases we get almost identical pattern where the Soft-ConFL yields better results for lower update rates and for higher update rates they yield identical result. Interestingly Soft-ConFL yields better results for $9\% < M < 12\%$ and facility cost of 5,000.

In Figure 5.7, we study in more details the results of Soft-ConFL for different update rates. As expected, the higher the cost to update the facilities the less facilities have been chosen (Starting with 10 facilities for 2.8% and ending with 3 facilities for 13.9%). Note that the decrease in the set of opened facilities was not done by picking a subset of an existing set, but the algorithm introduced new facilities and closed existing facilities. Some

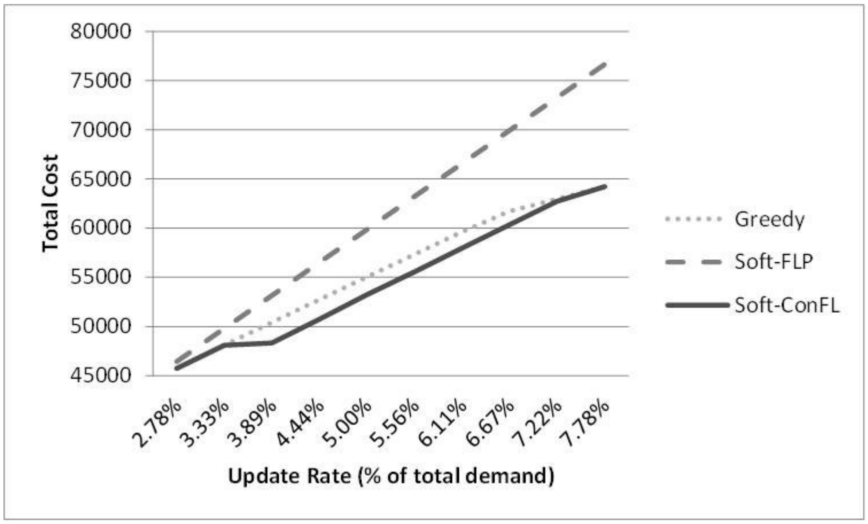

Figure 5.5: Soft-ConFL algorithm vs. Greedy algorithm and Soft-FLP (facility cost of 3,000)

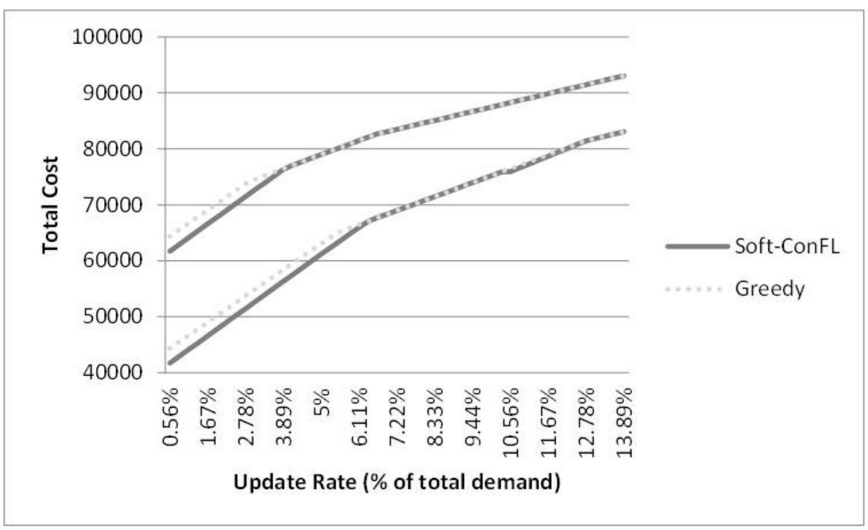

Figure 5.6: Soft-ConFL algorithm vs. Greedy algorithm (facility cost of 5,000 and 10,000)

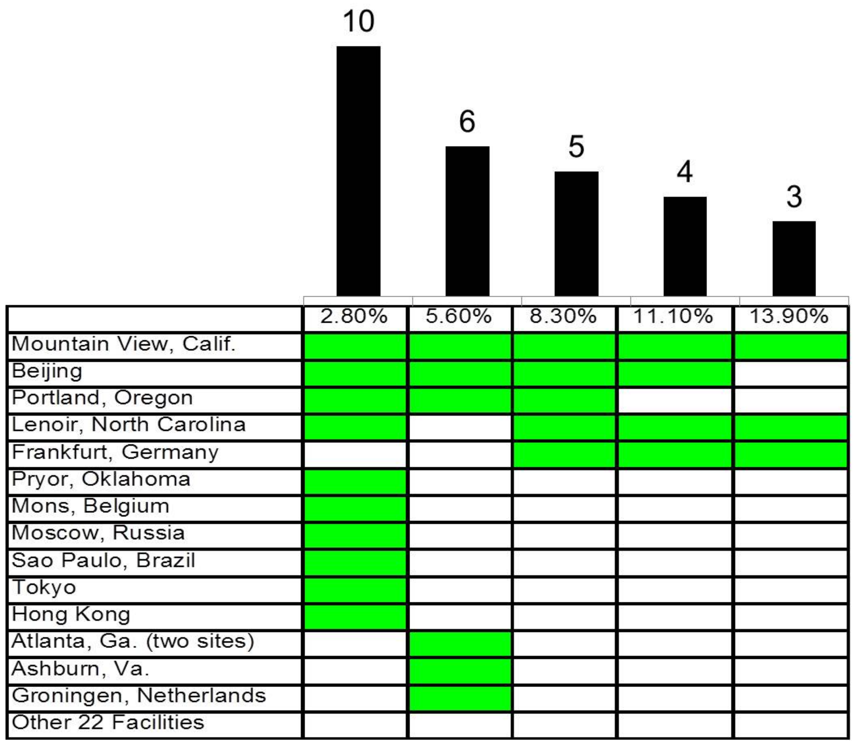

Figure 5.7: Soft-ConFL results for different update rates (facility cost of 1,000)

interesting facilities are: the Mountain View facility that was chosen in any iteration, the Lenoir facility that was chosen at the first iteration (2.8%), was not chosen in the second iteration but was re-chosen again in the other iterations and the Frankfurt facility that was added to the open facility sets only from the third (8.3%) iteration.

Chapter 6

Conclusion and Future Work

In this work, we presented the Soft-ConFL algorithm which is the first deterministic constant approximation algorithm for the soft capacitated connected facility location problem. We proved that given any deterministic constant approximation algorithms to the uncapacitated facility location problem and to the minimal Steiner tree problem, we can construct a constant approximation algorithm to the soft capacitated connected facility location problem by an appropriate modification of the problem parameters.

We evaluated the performance of our proposed algorithm through detailed simulation experiments using a real data center network topology. We compared our scheme against existing alternatives (the greedy algorithm and local-search facility location algorithm) to assess its technical soundness, and we showed that our algorithm provides practically good placement decisions.

Future research in this field may consider hard capacity constraints where a facility can be opened at most once in each location. This variant might be relevant for cases where the bottleneck for serving clients' demand is the network (either than processing power). Additionally, It is interesting to evaluate the performance of Soft-ConFL using other uncapacitated facility location algorithms that are based on a linear programming or Lagrangian relaxation approaches.

Bibliography

[1] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, *Volley: Automated Data Placement for Geo-Distributed Cloud Sevices*, In Proceedings of the 7th USENIX conference on Networked systems design and implementation (NSDI), pages 2-2, Berkeley, CA, USA, 2010.

[2] X. Meng, Y. Pappas, and L. Zhang, *Improving the scalability of data center networks with traffic-aware virtual machine placement*, In Proceedings of the 29th conference on Information communications (INFOCOM), pages 1154-1162, 2010.

[3] S. Borst, V. Gupta, and A. Walid, *Distributed Caching Algorithms for Content Distribution Networks*, In Proceedings of the 29th conference on Information communications (INFOCOM), pages 1478-1486, 2010.

[4] S. Guha, A. Meyerson, K. Munagala, *Hierarchical placement and network design problems*, In Proceeding of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 603-612, 2000.

[5] D. R. Karger and M. Minkoff, *Building Steiner trees with incomplete global knowledge*, In Proceeding of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 613-623, 2000.

[6] J. Byrka, *An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem*, In Proceeding of the 10th International Workshop on Approximation Algorithms for

Combinatorial Optimization Problems (APPROX), pages 29-43, 2007.

[7] M. Sviridenko, *An improved approximation algorithm for the metric uncapacitated facility location problem*, In Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization, pages 240-257, 2002.

[8] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi and B. Yener, *Provisioning a virtual private networks: A network design problem for multicommodity flow*, In Proceeding of the 33rd Annual ACM Symposium on Theory of Computing (STOC), pages 389-398, 2001.

[9] C. Swamy and A. Kumar, *Primal-dual algorithms for connected facility location problems*, In Proceeding of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), pages 256-269, 2002.

[10] C. Swamy and A. Kumar, *Primal-dual algorithms for connected facility location problems*, Algorithmica, vol. 40, no. 4, pages 245-269, 2004.

[11] H. Jung, M. Hasan and K. Y. Chwa, *Improved primal-dual approximation algorithm for the connected facility location problem*, Combinatorial Optimization and Applications, pages 265-277, 2008.

[12] D. P. Williamson and A. van Zuylen, *A simpler and better derandomization of an approximation algorithm for single source rent-or-buy*, Operations Research Letters, 35(6):707-712, 2007.

[13] F. Eisenbrand, F. Grandoni, T. RothvoB and G. Schafer, *Approximating connected facility location problems via random facility sampling and core detouring*, In Proceeding of the nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1174-1183, 2008.

[14] S. Gollowitzer and I. Ljubic , *MIP models for connected facility location: A theoretical and computational study*, Technical Report Nr. [2009-07], Department of Statistics and Decision Support Systems, University of Vienna, Austria, 2009.

[15] F. Eisenbrand, F. Grandoni, T. Rothvoβ and G. Schafer, *Connected facility location via random facility sampling and core detouring*, Journal of Computer and System Sciences 76: 709-726, 2010.

[16] F.M. Aymerich, G. Fenu, and S. Surcis, *An approach to a Cloud Computing network*, in Proceedings of IEEE Applications of Digital Information and Web Technologies. IEEE: Ostrava, Czech Republic, pages 113-118, 2008.

[17] L.M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, *A Break in the Clouds: Towards a Cloud Definition*, SIGCOMM Computer Communication Review, 39(1): pages 50-55, 2009.

[18] K.S. Candan, W.S. Li, T. Phan and M. Zhou, *Frontiers in information and Software as Services*, in Proceeding of the IEEE 25th International Conference on Data Engineering. IEEE: Shanghai, China, pages 1761-1768, 2009.

[19] P.A. Laplante, Z. Jia and J. Voas, *What's in a name? Distinguishing between SaaS and SOA*, IT Professional, 10(3): pages 46-50, 2008.

[20] T. Kwok and A. Mohindra, *Resource calculations with constraints, and placement of tenants and instances for multi-tenant SaaS applications*, in Sixth International Conference on Service-Oriented Computing. Springer: Sydney, Australia, pages 633-648, 2008.

[21] Z.I.M. Yusoh and T. Maolin, *A penalty-based genetic algorithm for the composite SaaS placement problem in the Cloud*, IEEE Congress on Evolutionary Computation (CEC), pages 1-8, 2010.

[22] F. Barahona and D. Jensen, *Plant location with minimum inventory*, Mathematical Programming 83:101-111, 1998.

[23] P. Mirchandani and R. Francis, eds. Discrete Location Theory. John Wiley and Sons, Inc., New York, 1990.

[24] P. Bauer and R. Enders. *A capacitated facility location problem with integer decision variables*, In International Symposium on Mathematical Programming (ISMP), 1997.

[25] D. Shmoys, E. Tardos and K. Aardal, *Approximation algorithms for facility location problems*, In 29th ACM Symposium on Theory of Computing, pages 265-274, 1997.

[26] F.A. Chudak and D. Shmoys, *Improved approximation algorithms for the capacitated facility location problem*, In Proceeding 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 875-876, 1999.

[27] K. Jain and V.V. Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation*, Journal of the ACM, pages 274-296, 2001.

[28] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala and V. Pandit, *Local search heuristics for k-median and facility location problems*, In Proceedings of 33rd ACM Symposium on Theory of Computing, 2001.

[29] M. Mahdian, Y. Ye and J. Zhang, *Improved approximation algorithms for metric facility location problems*, In Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, pages 229-242, 2002.

[30] M. Mahdian, Y. Ye and J. Zhang, *A 2-approximation algorithm for the soft-capacitated facility location problem*, In Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization, pages 129-140, 2003.

[31] M. Leitner, G. R. Raidl, *Variable neighborhood search for a prize collecting capacity constrained connected facility location problem*, In Proceedings of the 2008 International Symposium on Applications and the Internet, IEEE Computer Society, pages 233-236, 2008.

[32] M. Leitner, G. R. Raidl, *A Lagrangian decomposition based heuristic for capacitated connected facility location* In Proceedings of the 8th Metaheuristic International Conference (MIC), Hamburg, Germany, 2009.

[33] M. Leitner, G. R. Raidl, *Combining Lagrangian decomposition with very large scale neighborhoood search for capacitated connected facility location*, In: Post-Conference Book of the Eight Metaheuristics International Conference (MIC), 2010.

[34] M. Leitner, G. R. Raidl, *Branch-and-Cut-and-Price for Capacitated Connected Facility Location*, Technical Report TR 186-1-10-01, Vienna University of Technology, Vienna, Austria, 2010.

[35] G. Peng, *CDN: Content distribution Network*, Technical Report TR-125, Experimental Computer Systems Lab, Department of Computer Science, State University of New York, Stony Brook, 2004.

[36] C. D. Cranor, M. Green, C. Kalmanek, D. Shur, S. Sibal, J. E. Van der Merwe, C. J. Sreenan, *Enhanced Streaming Services in a Content Distribution Network*, IEEE Internet Computing, vol. 5, no. 4, 2001.

[37] James D. Guyton and Michael F. Schwartz, *Locating nearby copies of replicated internet servers*, Technical Report CU-CS-762-95, Department of Computer Science, University of Colorado-Boulder, 1995.

[38] N. Fujita, Y. Ishikawa, A. Iwata, and R. Izmailov, *Coarse-grain Replica Management Strategies for Dynamic Replication of Web Contents*, Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 45, Issue 1, pp. 19-34, May 2004.