National and Kapodistrian University of Athens

Department of Mathematics

Graduate Program in Logic, Algorithms and Computation

Approximation Algorithms on Network Resource Allocation

M.Sc. Thesis

by Anna Koutli

Supervisor: Professor Vassilis Zissimopoulos, UoA

Athens, July 2015

Η παρούσα Διπλωματική Εργασία
εκπονήθηκε στα πλαίσια των σπουδών
για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης
στη
Λογική και Θεωρία Αλγορίθμων και Υπολογισμού
που απονέμει το
Τμήμα Μαθηματικών
του
Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών

Εγκρίθηκε την …………… από Εξεταστική Επιτροπή
αποτελούμενη από τους:

Ονοματεπώνυμο	**Βαθμίδα**	**Υπογραφή**
1. ………………………..	………………………………………………….	……………………
2. ………………………..	………………………………………………….	……………………
3. ………………………..	………………………………………………….	……………………

Abstract

Resource Allocation Problems are of particular importance not only from a theoretical aspect, but also due to their applications in network design. A central place in this area have occupied the so called *Facility Location Problems*, where given a set of facilities with opening costs and a set of clients with service demands and their respective connection costs, the objective is to satisfy the clients' demands while minimizing the total cost. This line of problems dates back to the 60's, however it was not until the mid-90's that a breakthrough was made in terms of approximation. Since then, numerous variants and results have appeared.

In this work, some of the most representative variants are examined followed by a comparison of the techniques employed so far to tackle them, most prominently LP-based techniques and local search. We begin by a thorough presentation of the basic Uncapacitated Facility Location Problem and then continue with the more generic Fault-Tolerant setting. We proceed by analyzing a capacitated version, a variant with outliers and generalizations of the original problem. Finally, we examine the case where the notion of *time* is introduced, leading to Facility Leasing Problems.

Keywords

Resource Allocation Problems, Facility Location Problems, Facility Leasing, Fault-tolerant objectives, Combinatorial Optimization, LP-based techniques, Local Search.

Acknowledgements

Θα ήθϵλα να ϵυχαριστήσω την τριμϵλή ϵπιτροπή για την υποστήριξη, τον κ. Θηλυκό για το ϵνδιαφέρον, τον Ανδρέα, τον Κωστή και τον Κώστα από την Programize για την κατανόηση και τη βοήθϵια μϵ κάθϵ δυνατό τρόπο, το softlab για τη φιλοξϵνία, τον κ.Παπασπύρου ϵπϵιδή ϵίναι πρότυπο ανθρώπου και καθηγητή, την οικογένϵια και τους φίλους μου που δϵν δυσανασχέτησαν μϵ την ϵξαφάνισή μου, και, ιδιαιτέρως, το Δημήτρη που μϵ ηρϵμούσϵ σϵ κάθϵ κρίση και βοήθησϵ έμπρακτα ουκ ολίγες φορές...

Contents

Chapter 1

The Facility Location Problem

1.1 Preliminaries

In Facility Location Problems we are given a set of demand points $\mathcal{C}$ and a set of facilities $\mathcal{F}$. The opening cost of a facility $i \in \mathcal{F}$ is f_i and the cost of connecting a client $j \in \mathcal{C}$ to facility i is equal to the distance c_{ij} between j and i. In the metric uncapacited version, clients and facilities are embedded in a metric space and the goal is to open a subset of facilities such that each is connected to an open facility, with the objective to minimize the total (opening and connection) cost.

There are numerous variants of Facility Location Problems. If at least one client requires to be connected to more than one facility to satisfy the demand, the problems are called Fault-Tolerant. In other cases, we can chose not to connect a client and pay a penalty instead. There is also quite an extensive literature on hierarchical facility location models. Sahin and Sural in their survey [51] classify these problems depending on flow patterns, service availability at each level of hierarchy, and spatial configuration of services in addition to the objectives to locate facilities.

It is easily shown that non-metric FLP is at least as hard as the set cover problem, thus an $\mathcal{O}(\log n)$ approximation is the best attainable. In fact, this can be achieved by formulating the non-metric FLP as a set cover problem and then apply the standard greedy algorithm for the set cover. Furthermore, Guha and Khuller [26] proved in 1999 that there can be no better possible approximation factor for the Metric Uncapacitated Facility Location than 1.463, unless $NP \subseteq DTIME[n^{O(\log\log n)}]$. In 2002, Jain et al. [33] further generalized the result and proved that there exists no (γ_f, γ_c)-bifactor approximation for $\gamma_c < 1 + 2e^{-\gamma_f}$ unless $NP \subseteq DTIME[n^{O(\log\log n)}]$, and in the same year Sviridenko strengthened the inapproximability results showing that there is no ρ-approximation for the UFL with $\rho < 1.463$ unless $P = NP$ [55]. However, there can be versions of the problem solved in polynomial time, such as the structured p-facility location problem on the line, where we select no more than p sites for facilities to serve customers located at n demand points and the cost of serving any customer is a unimodal function of the location of the serving facilities [31]. According to Byrka & Aardal in [14], the approximation gap for the metric facility location problem is not yet closed.

More formally, in the (uncapacitated) facility location (UFL) problem, we have a set $\mathcal{F}$ of n_f facilities and a set $\mathcal{C}$ of n_c clients. For every facility $i \in \mathcal{F}$ there is a non-negative opening cost f_i and for every facility $i \in \mathcal{F}$ and client $j \in \mathcal{C}$, there is a connection cost c_{ij}. The objective is to open a subset of facilities and connect all the clients to these open facilities so that the total cost is minimized. From now on, we shall only refer to the metric version, where connection costs satisfy the triangle inequality. Originally, the problem was studied in the early 60s from more traditional combinatorial optimization perspectives, such as worst case analysis, polyhedral combinatorics and and empirical heuristics (Balinski [10], Kuehn & Hamburger [40], Manne [47], Stollsteimer [54], Cornuejols et al. [21]).

In 1980 Hochbaum [30] presented the first approximation algorithm for this kind of problems. Assuming that the customer set $I = \{1, 2, ..., n\}$ consists of finite discrete locations and that J is the feasible facility location set, where J might be: (i) a discrete set of points, (ii) any point in a finite dimensional Euclidean space with cost the d_{ij} Euclidean distances, or (iii) considering the points of I as vertices on a graph, the set J consists of the vertices and any point along the arcs between them, with an arbitrary cost function associated with the arcs, Hochbaum reformulated each case as a set covering problem and employed the standard greedy heuristic, thus guaranteeing an $O(\log n)$ approximation for the general non-metric facility location problems.

The first constant factor approximation algorithm for the metric UFL was given in 1997 by Shmoys, Tardos and Aardal [53]. Based on the filtering technique of Lin & Vitter [44], which rounded fractional solutions to linear programming relaxations, their algorithm yielded a 3.16-approximation for the UFL. Moreover, they presented algorithms with constant approximation to certain variants of the capacitated version and a 4-approximation algorithm based on filtering and rounding for the $2-level$ uncapacitated facility location problem. One year later, Chudak [18] further improved the approximation factor for the UFL to $1 + 2/\epsilon$. Again, his algorithm was based on

LP-rounding, using properties of optimal solutions to the linear program, randomized rounding, as well as a generalization of the decomposition techniques of Shmoys, Tardos, and Aardal.

1.2 Primal-Dual Schemes

Both of the initial algorithms bearing a constant factor approximation for the UFL were based on LP-rounding and therefore had high running times. The first to employ a primal-dual scheme were Jain & Vazirani [35] in 1999, with a running time of $O(m \log m)$, where $m = n_c \times n_f$ is the total number of edges in the underlying complete bipartite graph on clients and facilities. The novelty of their method lies in extending the primal-dual schema to handle a primal-dual pair of of LP's that are not a covering-packing pair. It is also worth mentioning that their algorithm is suitable for distributed computation.

Following is the most commonly proposed integer program for the UFL, which is also employed by Jain and Vazirani. In this program, y_i denotes whether facility i is open and x_{ij} indicates whether client j is connected to facility i. The first constraint ensures that each client is connected to at least one facility and the second ensures that this facility must be open.

$$minimize \sum_{i \in \mathcal{F}, j \in \mathcal{C}} c_{ij} x_{ij} + \sum_{i \in \mathcal{F}} f_i y_i$$

$$\text{subject to:} \quad \sum_{i \in \mathcal{F}} x_{ij} \geq 1 \qquad \forall j \in \mathcal{C} \tag{1.1}$$

$$y_i - x_{ij} \geq 0 \qquad \forall i \in \mathcal{F},\ j \in \mathcal{C} \tag{1.2}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{F},\ j \in \mathcal{C} \tag{1.3}$$

$$y_i \in \{0, 1\} \qquad \forall i \in \mathcal{F} \tag{1.4}$$

The LP-relaxation of the above program is:

$$minimize \sum_{i \in \mathcal{F}, j \in \mathcal{C}} c_{ij} x_{ij} + \sum_{i \in \mathcal{F}} f_i y_i$$

$$\begin{aligned} \text{subject to:} \quad & \sum_{i \in \mathcal{F}} x_{ij} \geq 1 && \forall j \in \mathcal{C} && (1.5) \\ & y_i - x_{ij} \geq 0 && \forall i \in \mathcal{F},\ j \in \mathcal{C} && (1.6) \\ & x_{ij} \geq 0 && \forall i \in \mathcal{F},\ j \in \mathcal{C} && (1.7) \\ & y_i \geq 0 && \forall i \in \mathcal{F} && (1.8) \end{aligned}$$

The dual program is:

$$maximize \sum_{j \in \mathcal{C}} \alpha_j$$

$$\begin{aligned} \text{subject to:} \quad & \alpha_j - \beta_{ij} \leq c_{ij} && \forall i \in \mathcal{F},\ j \in \mathcal{C} && (1.9) \\ & \sum_{j \in \mathcal{C}} \beta_{ij} \leq f_i && \forall i \in \mathcal{F} && (1.10) \\ & \alpha_j \geq 0 && \forall j \in \mathcal{C} && (1.11) \\ & \beta_{ij} \geq 0 && \forall j \in \mathcal{C},\ i \in \mathcal{F} && (1.12) \end{aligned}$$

Suppose the LP has an optimal solution that is integral, i.e. $I \subseteq \mathcal{F}$ and $\phi : \mathcal{C} \to I$ and for this solution, $y_i = 1$ iff $i \in I$, and $x_{ij} = 1$ iff $i = \phi(j)$. Let also (α, β) the optimal dual solution. In a nutshell, the primal and dual complementary slackness conditions imply the following:

- Each open facility is fully paid for, that is if $i \in I$, then

$$\sum_{j:\phi(j)=i} \beta_{ij} = f_i.$$

- Suppose client j is connected to facility i, that is $\phi(j) = i$. Then, j does not contribute for opening any facility besides i, i.e. $\beta_{i'j} = 0$ if $i' \neq i$. Furthermore, $\alpha_j - \beta_{ij} = c_{ij}$, that is α_j can be viewed as the total price paid by client j, where c_{ij} is the connection cost for edge (i, j) and β_{ij} is the contribution of j towards opening facility i.

The standard approach would be to relax the dual complementary slackness conditions. Instead, Jain & Vazirani relax the primal conditions as follows, while maintaining the dual conditions:

- $\forall j \in \mathcal{C} : (1/3)c_{\phi(j)j} \leq \alpha_j - \beta_{\phi(j)j} \leq c_{\phi(j)j}$,
- $\forall i \in I : (1/3)f_i \leq \sum_{j:\phi(j)=i} \beta_{ij} \leq f_i$.

Then, the cost of the (integral) solution found would be within thrice the dual found, leading to a 3-approximation algorithm. However, in order to use the same algorithm to solve the k-median problem, the primal conditions are relaxed in a way such as to partition clients into the two sets of *directly* and *indirectly* connected. Now, only directly connected clients shall pay for opening facilities. Thus, for an indirectly connected client j, the primal condition is subsequently relaxed:

$$(1/3)c_{\phi(j)j} \leq \alpha_j \leq c_{\phi(j)j}.$$

Consequently, the Jain & Vazirani algorithm consists of two phases. In Phase 1, the algorithm finds a dual feasible solution and also determines a set of tight edges (aka. edges where the dual constraint corresponding to the connection goes tight, i.e. for a connection ij: $\alpha_j - \beta_{ij} = c_{ij}$) and a set of temporarily open facility F_t. In Phase 2, it chooses which subset I of F_t to open and a mapping ϕ from clients to I is determined. More detailed:

I and ϕ define a primal integral solution, where $x_{ij} = 1$ if $\phi(j) = i$ and $y_i = 1$ if $i \in I$. The values of α_j and β_{ij} obtained at the end of Phase 1 form a feasible dual solution. It remains to be shown how the dual α_j's pay for the primal opening and connecting costs. Denote by α_j^f and α_j^e the contribution of client j to these two costs respectively, so that $\alpha_j = \alpha_j^f + \alpha_j^e$. If j is indirectly connected, then $\alpha_j^f = 0$ and $\alpha_j^e = \alpha_j$, else, if j is directly connected, then $\alpha_j = c_{ij} + \beta_{ij}$, where $\imath = \phi(j) = i$. From now on, let $\alpha_j^f = \beta_{ij}$ and $\alpha_j^e = c_{ij}$. It is fairly straightforward to see that for $i \in I$ if (i, j) was tight at the end of Phase 1, then $\phi(j) = i$ and that (for $i \in I$ again) $\sum_{j:\phi(j)=i} \alpha_j^f = f_i$, which also leads to $\sum_{i \in I} f_i = \sum_{j \in \mathcal{C}} \alpha_j^f$. Note that only the directly connected clients pay for the cost of opening facilities. So, it is proved that for an indirectly connected client j, $c_{ij} \leq 3\alpha_j$, where $i = \phi(j)$:

Proof: Since j is indirectly connected to i, there is a tight edge (i', j) such that i is the closing witness for i' and i' the connecting witness for j. That is, there is an edge (i', i) in H and, thus, there must be client j' that has a tight edge to both facilities i and i'. Let t_i and $t_{i'}$ be the time at which i and i' were respectively declared temporarily open. Given that i is the closing witness for i', $t_i \leq t_{i'}$. Since edge (i', j) is tight, $\alpha_j \geq c_{i'j}$, and since i' was the connecting witness for j, $\alpha_j \geq t_{i'}$.

Algorithm Phase 1

1. A notion of time is defined so that each event can be associated with the time it happened. The phase starts at time 0, with zero primal and zero dual solution and all clients are *unconnected*. As time passes, the algorithm raises the dual variable α_j for each *unconnected* client uniformly at rate one. The following events may happen - if several events happen at the same time, choose arbitrarily one of them:

   - When $\alpha_j = c_{ij}$ for some edge (i, j), this edge is considered *tight*. Starting from this point, the dual variable β_{ij} is raised uniformly in order to ensure that the first constraint in the dual LP is not violated. If i is temporarily open, j is declared connected to i and i is considered the connecting witness for j (see below).

   - Considering that β_{ij} goes towards paying for facility i, facility i is considered paid for if $\sum_j \beta_{ij} = f_i$. When a facility i is fully paid for and there is a client j having a *tight* edge to i such that j is still unconnected, this facility is declared *temporarily open* and *all* unconnected cities having tight edges to this facility are declared *connected* and i is considered to be the *connecting witness* for each of these clients. The dual variables α_j of these clients are not raised anymore.

2. Repeat until all clients get connected.

Algorithm Phase 2

1. Let $\mathcal{F}_t$ denote the set of *temporarily* open facilities and T denote the graph with vertex set V consisting of the client set $\mathcal{C}$ and the set of temporarily open facilities $\mathcal{F}_t$ and edge set E consisting of these edges that where tight at the end of Phase 1.

2. Consider a new graph H that has $\mathcal{F}_t$ as the vertex set and edge (i_1, i_2) if both facilities $i_1,\ i_2$ where connected in T to the same client j, that is if both facilities were "tightly" opened by the same client j.

3. Order the facilities in $\mathcal{F}_t$ according to the time they were temporarily opened and pick a *maximal* independent set I of $\mathcal{F}_t$ in H, beginning from the earliest facility.

4. While considering a facility i, if it has a neighbour in the independent set, then this neighbour is called the *closing witness* for i and i remains closed. In the end, all facilities in I are declared *open*.

5. Clients are then connected to the *open* facilities:

   - If for a client j there is a tight edge (i, j) and i is open, then $\phi(j) = i$ and j is considered *directly* connected.
   - Otherwise, consider the tight edge (i, j) with i the connecting witness for j and, since $i \notin I$, its closing witness i' is open, so define $\phi(j) = i'$ and client j becomes *indirectly* connected.

It holds that $t_{i'} \geq c_{ij'}$ and $t_{i'} \geq c_{i'j'}$. Suppose not. Then consider the edge which went tight later and the instant at which it was declared tight. At this instant however, both i and i' had been declared temporarily open and so j' must have already been declared connected, which leads to a contradiction, since an already connected city cannot get additional tight edges. Hence the cost of each of the three edges $(i', j), (i', j')$ and (i, j') is bounded by a_j. By the triangle inequality, we get that $c_{ij} \leq c_{i'j} + c_{i'j'} + c_{ij'} \Rightarrow c_{ij} \leq 3\alpha_j$. □

Taking into consideration all the above mentioned, it now follows that the primal solution constructed by the algorithm is at most thrice the dual solution:

$$\sum_{i \in \mathcal{F}, j \in \mathcal{C}} c_{ij} x_{ij} + 3 \sum_{i \in \mathcal{F}} f_i y_i \leq 3 \sum_{j \in \mathcal{C}} \alpha_j.$$

Also, the algorithm runs in polynomial time, with the sorting step weighing the most: we sort all edges by increasing cost in $O(m \log m)$, where $m = n_f \times n_c$. Then, for each facility i, we maintain its unpaid cost and the current number of clients that are contributing towards its cost, initialising each to f_i and 0 respectively. Thus, each iteration takes $O(n_f)$ time and there are $O(n_c)$ iterations. Therefore, besides the sorting, the rest of the algorithm takes linear time, that is $O(m)$, and thus it is a 3-approximation algorithm for the UFL with a running time of $O(m \log m)$.

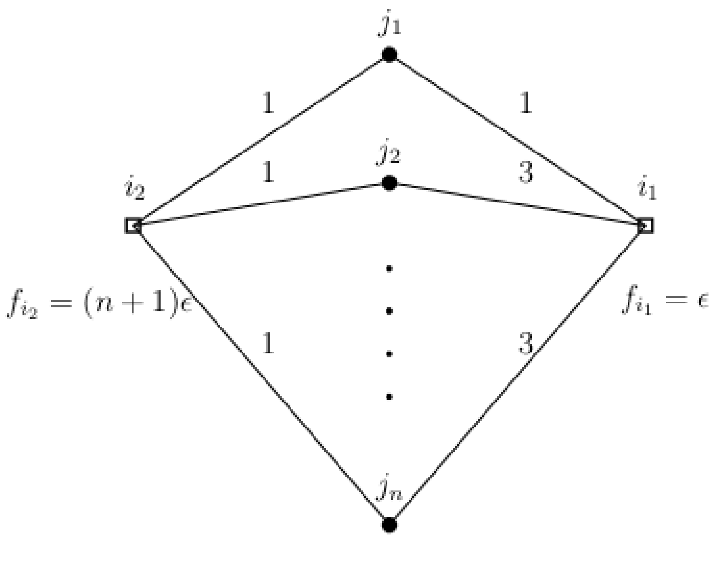

Figure 1.1: tightness example

This approximation is *tight*, as shown by the following instance: Consider a graph with n clients, $j_1, j_2, ..., j_n$ and two facilities i_1 and i_2, with $c_{i_2j} = 1$, $\forall j \in \mathcal{C}$. Also, for facility i_1 we have $c_{i_1j_1} = 1$ and $c_{i_1j_l} = 3$ for $l = 2, ..., n$. The opening costs are $f_{i_1} = \epsilon$ and $f_{i_2} = (n+1)\epsilon$, for a small number ϵ. The optimal solution is to open i_2 and connect all clients to it, at a total cost of $(n+1)\epsilon + n$. However, the algorithm will open facility i_1 and connect all clients to it, at a total cost of $\epsilon + 1 + 3(n-1)$.

In 2004, Charikar & Guha [16] presented a gap example showing that the primal-dual algorithm can construct a dual whose value is $3 - \epsilon$ away from the optimal for arbitrarily small ϵ, introducing however in their example the notion of *demand*:

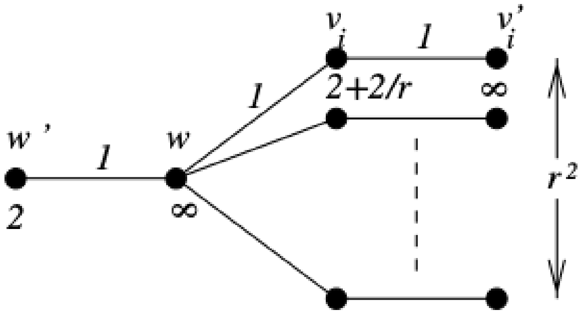

Figure 1.2: gap example - image taken from [16]

Suppose the example instance as tree rooted at w'. w' has a child w with service cost $c_{w'w} = 1$. w also has r^2 children $v_1, ..., v_{r^2}$ with connection cost $c_{wv_i=1}$, for $i = 1, ..., r^2$, where r is a parameter. Also, each v_i has an only child v'_i with $c_{vv'_i} = 1$. To complete the instance graph, all other distances are considered as shortest path instances along the tree. Node w' has facility cost $f_{w'} = 2$ and the nodes v_i have facility cost $f_{v_i} = 2 + 2/r$. All other nodes are considered without facility or, more conveniently, having facility costs of ∞. The node w has demand $2r$ - a part in their proof which strays from the original UFL and leans more towards the fault-tolerant version. Each v'_i node has unit demand and the rest of the nodes have zero demand. From the dual solution returned by the algorithm, $\alpha_w = 2r(1+1/r)$ and $\alpha_{v'_i} = 1+2/r$ for each v'_i. The value of the dual solution is $r^2 + 4r + 2$, while the optimal solution by choosing on the v_i facilities has total cost $3r^2 + 2r + 2/r$. Therefore, for r large enough, the dual solution is $3-\epsilon$ times the optimal. This means that it is not possible to prove an approximation ratio better than $3 - \epsilon$ using the dual constructed as a lower bound and can be considered as the analog of an integrality gap for LPs.

1.3 Combinatorial Approaches

In 2000 Korupolu, Plaxton and Rajaraman [38] proved that a simple local search heuristic achieves a $(5+\epsilon)$ approximation ratio in $O(n^4 \log n/\epsilon)$ running time. Four years later, Charikar & Guha [16] applied the ideas of *cost scaling* and *greedy improvement* on a more complex greedy local search algorithm of their conception, which also served as introduction the fault-tolerant version of the problem, by covering the case where each node j has a demand d_j. We proceed to a brief presentation of Charikar & Guha's local search algorithm and afterwards we shall expand on the aforementioned techniques.

The initial solution is chosen by sorting the facilities in order of increasing facility cost and picking the first i facilities in this order which satisfy the needs while minimizing the total cost $F_i + C_i$. The initial solution is computed in $O(n^2)$ time and it is shown that it is at most $n^2 F_{SOL} + nC_{SOL}$, where F_{SOL} and C_{SOL} are the facility and service costs of an arbitrary solution SOL to the LP.

At each local search step, let $\mathcal{F}$ be the set of facilities in the current solution. The algorithm considers a random facility i which may or may not already belong to $\mathcal{F}$ and examines updating the solution with $i \in \mathcal{F}$. Nodes that are assigned to a facility already in $\mathcal{F}$ but are closer to i are now assigned to i. Also, the algorithm tries to remove facilities already in the current solution. If $i' \in \mathcal{F}$ is removed, then all clients j connected to i' are now connected to i. The authors define as $gain(i)$ the largest possible decrease in F + C by switching clients to i and removing facilities. If the total cost only increase by adding facility i to $\mathcal{F}$, then $gain(i)$ is said to be 0. If $gain(i) > 0$, then i is included to $\mathcal{F}$ and the solution is rearranged accordingly. This step is repeated until there is no i such that $gain(i) > 0$.

It is shown that $gain(i)$ can be computed in $O(n)$ time. Also, $\sum gain(i) \geq C - (F_{SOL} + C_{SOL})$ and $\sum gain(i) \geq F - (F_{SOL} + 2C_{SOL})$, where F_{SOL} and C_{SOL} are the facility and connection costs of an arbitrary *fractional* solution SOL. Thus, the local search algorithm after $O(n \log(n/\epsilon))$ iterations reaches a solution $C + F \leq 2F_{SOL} + 3C_{SOL} + \epsilon(F_{SOL} + C_{SOL})$ with probability at least 1/2, leading to the following Theorems:

Theorem. *The algorithm reaches a solution with* $F \leq (1+\epsilon)(F_{SOL} + 2C_{SOL})$ *and* $C \leq (1+\epsilon)(F_{SOL} + C_{SOL})$ *in* $O(n(\log n + \frac{1}{\epsilon}))$ *steps, that is* $O(n^2(\log n + \frac{1}{\epsilon}))$ *running time, with constant probability.*

Theorem. *The derandomized algorithm finds a solution such that* $F \leq (1+\epsilon)(F_{SOL}+$

$2C_{SOL}$) and $C \leq (1+\epsilon)(F_{SOL}+C_{SOL})$ in $O(n(\log n + \frac{1}{\epsilon}))$ steps with $O(n^3(\log n + \frac{1}{\epsilon}))$ running time, by examining all facilities and choosing the one with the highest gain,instead of picking a random facility i.

The novelty of their method comes to the application of cost scaling in order to exploit the asymetry in the guarantees of the service and facility cost. The idea is to scale the facility costs uniformly by a factor δ and then solve the modified instance by using local search. The solution of the modified instance is then scaled back to determine the cost of the original instance.

Assuming that the facility and connection costs of the optimal solution are F_{OPT} and C_{OPT} respectively, then after scaling there exists a solution to the modified instance of δF_{OPT} facility cost and C_{OPT} connection cost. From the above theorem, there exists a solution to the scaled instance such that:

$$F \leq (1+\epsilon')(\delta F_{OPT} + 2C_{OPT}) \text{ , and } C \leq (1+\epsilon')(\delta F_{OPT} + C_{OPT})$$

By scaling back to a solution of facility cost F/δ and setting $\delta = \sqrt{2}$, it is easily shown that the UFL can be approximated to a $1 + \sqrt{2} + \epsilon$ factor in randomized $O(n^2/\epsilon + n^2 \log n)$ time, where $\epsilon = (1+\sqrt{2})\epsilon'$.

On a more general examination, by setting $\delta = 2C_{SOL}/(\gamma F_{SOL})$, it is shown that the facility cost is at most $(1+\gamma)F_{SOL}$ and the service cost is $(1+2/\gamma)C_{SOL}$ upto factors of $(1+\epsilon)$ for arbitrarily small ϵ. Furthermore, executing the above algorithm for some value of δ withina $1+\epsilon'$ factor of $2C_{SOL}/(\gamma F_{SOL})$ results in the following theorem:

Theorem. *Let SOL be any solution to the UFL, even fractional, with facility cost F_{SOL} and service cost S_{SOL}. For any γ, the above local search heuristic combined with scaling gives a $(1+\gamma, 1+2/\gamma)$-approximation upto multiplicative factors of $(1+\epsilon)$ for arbitrarily small $\epsilon > 0$.*

In the tradeoff problem using the (p, q) notation, p denotes the approximation factor of the facility cost and q the approximation factor of the service cost. Charikar & Guha' result is an improvement on the previously known tradeoffs and very close to the best possible, which is $(1+\gamma, 1+1/\gamma)$ as can be demonstrated in the following example:

Consider an instance $\mathcal{I}$ consisting of two nodes u and v with facility costs $f_u = 1$, $f_v = 0$ and connection cost $c_{uv} = 1$. The demands of the nodes are $d_u = 1$ and $d_v = 0$, that is, node u is also a *client*. Therefore, there are two integer solutions to

$\mathcal{I}$: SOL_1 picks u as a facility with $F_{SOL_1} = 1$ and $C_{SOL_1} = 0$, while SOL_2 picks v as a facility with $F_{SOL_2} = 0$ and $C_{SOL_2} = 1$. For $\gamma > 0$ a fractional solution SOL can be constructed for $\mathcal{I}$ with $F_{SOL} = 1/(1+\gamma)$ and $C_{SOL} = \gamma/(1+\gamma)$. This fractional solution is obtained by taking the linear combination $(1/(1+\gamma))SOL_1 + (\gamma/(1+\gamma))SOL_2$. It follows that there can be no integer solution with facility cost strictly less than $(1+\gamma)F_{SOL}$ and service cost strictly less than $(1+1/\gamma)C_{SOL}$.

It should be noted that Charikar & Guha also achieve an improved approximation for the capacitated version of the facility location problem by applying scaling with $\delta = 2\sqrt{2} - 2$, thus getting a $3 + 2\sqrt{2} + \epsilon$ approximation.

Based on the augmentation technique presented by Guha & Khuller [26], they alter the selection step of the local search algorithm by picking the node u of cost f_u such that if the total service cost decreases from C to C' after opening u, the ratio $\frac{C-C'-f_u}{f_u}$ is maximized. Observing that $C - C' - f_u$ is in fact $gain(u)$, they greedily pick the *gain* with the best ratio. Again, the node with the best ratio can be found in $O(n^2)$ time and since no node can be added twice, the algorithm takes $O(n^3)$. If the initial solution has facility and connection costs F_0 and C_0, respectively, and SOL is a feasible fractional solution, after application of the greedy augmentation the solution cost is at most

$$F_0 + F_{SOL} \max\left[0, \ln(\frac{C_0 - C_{SOL}}{F_{SOL}})\right] + F_{SOL} + C_{SOL} \; .$$

In the final version of their algorithm, Charikar & Guha first scale the facility costs by a δ such that $\ln(3\delta) = 2/(3\delta)$. Then they run the primal-dual algorithm on the scaled instance, scale back the solution and finally apply the greedy augmentation technique. This yields a 1.8526-approximation in $O(n^3)$ time and if combined with Chudak's LP-rounding algorithm in [18],the approximation drops to 1.728 .

1.4 Greedy approach using Dual-Fitting analysis

In 2002 Jain et al. in [33] introduced the notion of factor-revealing LP and one year later in [32] formalized the dual-fitting method for facility location problems and proceeded with its application to factor-revealing LPs.

A typical example of dual fitting analysis is in case of the simple set covering algorithm. In this greedy algorithm, the dual constructed is infeasible. However, the value of the primal *integral* solution returned by the algorithm is bounded by that of the dual. Thus, if the dual is divided by a factor γ, the shrunk dual is feasible in the sense that it fits into the original instance. In this case, the shrunk dual serves as a lower bound on the optimal solution and γ is the approximation factor. In order to find the minimum γ for the problem, one has to find the worst case scenario, that is an instance where the dual solution needs to be shrunk the most to become feasible. In this line, Jain et al. create a factor revealing LP that encodes the problem of finding the worst possible instance with n_c clients, resulting in a family of LP's, one for each value of n_c. The supremum of the optimal solutions to these LP's is factor γ. Because it is complex to compute the supremum, the authors instead prove that the upper bound on the (feasible) solutions of the duals of these LPs is also an upper bound on the optimal γ.

The idea of factor-revealing LPs is similar to using LP bounds in coding theory, where they are used to obtain bounds on the minimum distance of a code with a given rate. Factor revealing LPs essentially serve to prove a bound on the approximation ratio. In the case of UFL, Jain et al. also use it to estimate the approximation ratio of facility costs versus the approximation ratio of connection costs. They also prove that the algorithms presented in [33] and [32] have what they named the Lagrangian multiplier preserving property. This property is present in the classic primal-dual algorithm of Jain & Vazirani and they consider it the key to the versatility of algorithms which preserve that property. Suppose $\mathcal{A}$ and approximation algorithm for the UFL. $\mathcal{A}$ is a Lagrangian Multiplier Preserving α-approximation if for every isntance $\mathcal{I}$ with optimal solution cost OPT, F and C are the facility and connection costs, respectively, of the solution returned by algorithm $\mathcal{A}$ such that $C \leq \alpha(OPT - F)$.

With these two tools, Jain et al. formulated two versions of a greedy algorithm. The first and simpler algorithm follows the logic of the set cover algorithm - iteratively pick the most cost-effective choice at each step. They alter the LP formulation in order to treat facilities as "star-centers". More specifically, a star consists of one facility i as center and the cities connected to it. The cost of the star (i, C'), where i is the facility and $C' \subseteq C$ a subset of clients, is $f_i + \sum_{j \in C'} c_{ij}$. The cost of the star (i, C') is defined as $(f_i + \sum_{j \in C'} c_{ij})/|C'|$.They ensure that the primal solution is fully paid by the dual by setting the rule that when a city connects to an open facility, it withdraws its contribution towards the opening costs of other facilities. More specifically:

The star-formulated IP, where $\mathcal{S}$ is the set of all stars, c_S the cost of star S and

Algorithm Algorithm 1

```
1. Let $U$ the set of unconnected clients. In the beginning, all clients are unconnected ($U := C$) and all facilities are unopened.
2. While $U \neq \emptyset$:
    • Find the most cost-effective star $(i, C')$, open facility $i$, if it is not already open, and connect all cities in the star to $i$.
    • Set $f_i := 0$, $U := U \backslash C'$.
```

x_s the variable indicating whether star S is picked in the solution:

$$minimize \sum_{S \in \mathcal{S}} c_S x_S$$

$$\text{subject to:} \quad \sum_{S : j \in S} x_S \geq 1 \qquad \forall j \in \mathcal{C} \tag{1.13}$$

$$x_S \in \{0, 1\} \qquad \forall S \in \mathcal{S} \tag{1.14}$$

The LP-relaxation of the above program is:

$$minimize \sum_{S \in \mathcal{S}} c_S x_S$$

$$\text{subject to:} \quad \sum_{S : j \in S} x_S \geq 1 \qquad \forall j \in \mathcal{C} \tag{1.15}$$

$$x_S \geq 0 \qquad \forall S \in \mathcal{S} \tag{1.16}$$

The dual program is:

$$maximize \sum_{j \in \mathcal{C}} \alpha_j$$

$$\text{subject to:} \quad \sum_{S : j \in S \cap \mathcal{C}} \alpha_j \leq c_S \qquad \forall S \in \mathcal{S} \tag{1.17}$$

$$x_S \geq 0 \qquad \forall j \in \mathcal{C} \tag{1.18}$$

If (1.17) is re-written as $\sum_{j \in C} \max(0, \alpha_j - c_{ij}) \leq f_i$ for every facility i, the most cost effective star in each iteration can be found by raising the dual variables of all unconnected cities simultaneously until we reach the first star for which $\sum_{j \in C} \max(0, \alpha_j - c_{ij}) = f_i$. Thus, the algorithm can be restated in order to capture the LP formulation, which is similar to that of the set cover problem:

Algorithm Algorithm 1 restated

1. Introduce a notion of time, so as to associate the events with the "time" they occurred. Starting at time 0, each client is considered unconnected ($U := C$), all facilities unopened and $\alpha_j = 0 \; \forall \; j \in C$.

2. While $U \neq \emptyset$, increase the time and simultaneously for every client $j \in U$ increase variable α_j at the same rate, until one of the following events happen (if two events occur simultaneously, they are processed in random order):

   (a) $\alpha_j = c_{ij}$ for an unconnected city j and an open facility i. In this case, connect city j to facility i and remove j from U.

   (b) $\sum_{j \in C} \max(0, \alpha_j - c_{ij}) = f_i$ for an unopened facility i. In this case, open facility i and for every unconnected client j with $\alpha_j \geq c_{ij}$, connect j to i and remove j from U.

It is clear that the contribution α_j of each city j contributes to opening *at most* one facility. That way $\bar{\alpha}$ is not a feasible solution, because by excluding clients and withdrawing their contributions, there can be a facility i such that $\sum_{j \in C} \max(0, \alpha_j - c_{ij}) > f_i$. Consequently, a γ factor needs to be found such that $\bar{\alpha}/\gamma$ is feasible. Without loss of generality, clients can be ordered by increasing contribution $\alpha_1 \leq \alpha_2 \leq ... \leq \alpha_k$ and each time examine the bounds for the first k clients. That is, find the minimum γ for which $\sum_{j=1}^{k} (\alpha/\gamma - c_{ij}) \leq f_i$, or, equivalently, the maximum of the ratio $\sum_{j=1}^{k} \alpha_j / (f + \sum_{j=1}^{k} d_j)$, where $f = f_i$ and $d_j = c_{ij}$, leading to the following LP:

$$z_k = maximize \; \frac{\sum_{j=1}^{k} \alpha_j}{f + \sum_{j=1}^{k} d_j}$$

$$\text{subject to:} \quad \alpha_j \leq \alpha_{j+1} \qquad \forall j \in \{1, ..., k-1\} \tag{1.19}$$

$$\alpha_j \leq \alpha_l + d_j + d_l \qquad \forall j, l \in \{1, ..., k\} \tag{1.20}$$

$$\sum_{l=j}^{k} \max(\alpha_j - d_l, 0) \leq f \qquad \forall j \in \{1, ..., k-1\} \tag{1.21}$$

$$\alpha_j, d_j, f \geq 0 \qquad \forall j \in \{1, ..., k-1\} \tag{1.22}$$

The above z_k is equal to the optimal solution of the following factor-revealing LP:

$$z_k = maximize \sum_{j=1}^{k} \alpha_j$$

$$\text{subject to:} \quad f + \sum_{j=1}^{k} d_j \leq 1 \tag{1.23}$$

$$\alpha_j \leq \alpha_{j+1} \qquad \forall j \in \{1, ..., k-1\} \tag{1.24}$$

$$\alpha_j \leq \alpha_l + d_j + d_l \qquad \forall j, l \in \{1, ..., k\} \tag{1.25}$$

$$x_{jl} \geq \alpha_j - d_l \qquad \forall j, l \in \{1, ..., k\} \tag{1.26}$$

$$\sum_{l=j}^{k} x_{jl} \leq f \qquad \forall j \in \{1, ..., k\} \tag{1.27}$$

$$\alpha_j, d_j, f \geq 0 \qquad \forall j \in \{1, ..., k-1\} \tag{1.28}$$

By defining $\gamma = \sup_{k \geq 1}\{z_k\}$, we get that every facility is at most γ-overpaid. Also, the approximation factor of the algorithm is precisely $\sup_{k \geq 1}\{z_k\}$. It is difficult to calculate the exact upper bound of this supremum. However, Jain et al. tackled it by solving the dual of the factor-revealing LP for smaller values of k and thus made an assumption of a how small this upper bound could be. They proceed proving that, $\forall k \geq 1, \ z_k \leq 1.861$, which is their approximation factor of Algorithm 1. They furthermore present a tight example for $k = 2$, where the approximation factor is 1.5 (see Figure below), and by computer-aided calculations also show that $z_{300} \approx 1.81$, implying that the approximation factor actually lies somewhere between 1.81 and 1.861.

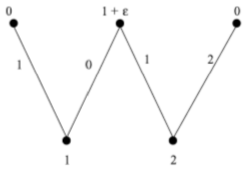

Figure 1.3: tight example for Algorithm 1 for $k = 2$ with approximation ratio 1.5 - image taken from [32]

An improved version of algorithm 1 follows, where clients now do not withdraw their contributions to other facilities once they get connected to an open facility, thus resulting to a better approximation of 1.61:

For the sake of brevity, we omit the analysis, which is similar to that of Algorithm 1.

The authors of [32] also present some results on the tradeoff between facility and connection costs concerning Algorithm 2:

Theorem 1. *Let $\gamma_f \geq 1$ and $\gamma_c := \sup_k\{z_k\}$, where z_k is the solution of the following LP:*

$$z_k = maximize \ \frac{\sum_{j=1}^{k} \alpha_j - \gamma_f f}{\sum_{i=1}^{k} d_i}$$

$$\begin{aligned}
\textit{subject to:} \quad \alpha_i &\leq \alpha_{i+1} & \forall 1 \leq i \leq k \quad & (1.29)\\
r_{j,i} &\geq r_{j,i+1} & \forall 1 \leq j \leq i \leq k \quad & (1.30)\\
\alpha_i &\leq r_{j,i} + d_i + d_l & \forall 1 \leq j \leq i \leq k \quad & (1.31)\\
\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=1}^{k} \max(\alpha_i - d_j, 0) &\leq f & \forall 1 \leq i \leq k \quad & (1.32)\\
\alpha_j, d_j, f, r_{j,i} &\geq 0 & \forall 1 \leq j \leq i \leq k \quad & (1.33)
\end{aligned}$$

Then for every instance $\mathcal{I}$ of UFL, and for every solution SOL for $\mathcal{I}$ with facility cost F_{SOL} and connection cost C_{SOL}, the cost of the solution found by Algorithm 2 is at most $\gamma_f F_{SOL} + \gamma_c C_{SOL}$.

Consequently, based on the proof in [26] for hardness of the UFL, they state the following theorem and compare the tradeoffs in the figure below:

Algorithm Algorithm 2

1. Introduce a notion of time. Starting at time 0, each client is considered unconnected ($U := C$), all facilities unopened and $\alpha_j = 0 \; \forall \; j \in C$. At every moment, each client j offers part of its contribution to each unopened facility i in the following way:

   – If j is unconnected, the offer is equal to $\max(\alpha_j - c_{ij}, 0)$.

   – If j is already connected to some other facility i', then its offer to i is equal to $\max(c_{i'j} - c_{ij}, 0)$.

2. While $U \neq \emptyset$, increase the time and simultaneously for every client $j \in U$ increase variable α_j at the same rate, until one of the following events happen (if two events occur simultaneously, they are processed in random order):

   (a) For an unopened facility i, the total offer it receives from clients is equal to the cost of opening i. In this case, open i and every client j (connected or unconnected) with nonzero contribution to i gets connected to i. Client j is no longer allowed to decrease the amount it offered to i.

   (b) For an unconnected client j and an open facility i, $\alpha_j = c_{ij}$. In this case, connect j to i and remove j from U.

Theorem 2. *Let γ_f and γ_c constants with $\gamma_c \leq 1 + 2e^{-\gamma_f}$. Assume there is an algorithm $\mathcal{A}$ such that for every instance $\mathcal{I}$ of the metric facility location problem, $\mathcal{A}$ finds a solution SOL whose cost is no more that $\gamma_f F_{SOL} + \gamma_c C_{SOL}$. Then $NP \subseteq DTIME\left[n^{O(\log\log n)}\right]$.*

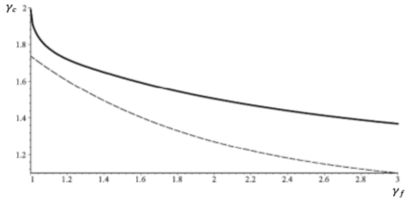

Figure 1.4: the tradeoff between γ_f and γ_c - image taken from [32]

The dashed line corresponds to the general lower bound stated in the Theorem 2, while the thick line represents the tradeoff between γ_f and γ_c, where $\gamma_f \geq 1$ and $\gamma_c := \sup_k\{z_k\}$ as described in Theorem 1, for $k = 100$ and after running tests on various values of γ_f between 1 and 3.

By setting $\gamma_f = 1$, Jain et al. elaborate on this tradeoff to design algorithms for other variants of the facility location problem.

1.5 Randomized Algorithms

Lin & Vitter [44], working on geometric median problems in 1992, used the *filtering technique* to round the fractional solution of the LP of the metric k-median problem to obtain an integer solution $2(1{+}1/\epsilon)$ times the fractional solution while using $(1{+}\epsilon)k$ medians. Five years later, Shmoys, Tardos and Aardal [53] used this technique combined with a rounding algorithm to achieve a 4-approximation.

1.5.1 Byrka's algorithm

The big break-through in the approach of the metric facility location problem came in 2007, when Byrka presented in [13] an algorithm with the optimal bifactor approximation ratio, which as proven in [32] is $(\gamma_f, 1 + 2e^{-\gamma_f})$. This was achieved by modifying the (1+ $2/e$)-approximation algorithm presented in [18] by Chudak, thus obtaining a 1,5-approximation, or else expressed as a bifactor approximation, an

(1.6774,1.3738)-approximation which gives optimal approximation in instances dominated by connection costs.

The key techniques employed by Byrka in this work are the modification of the support graph of the LP before clustering the clients and using the average distances of the fractional solution to bound the cost via a *sparsening* technique, similar to filtering. The sparsening technique is used to measure and control ”*irregular*” instances which are potentially tight for the original (1+ $2/e$)-approximation algorithm. A simplified overview of Byrka's algorithm follows, before we proceed to the analysis of his techniques and the final presentation of the algorithm:

```
Algorithm Overview
Step 1. solve the LP-relaxation with solution $(x^*, y^*)$
Step 2. scale up the fractional solution obtaining $(\bar{x}, \bar{y})$
Step 3. compute a greedy clustering on the scaled up solution $(\bar{x}, \bar{y})$, where cluster centers are those clients minimizing some average diastances
Step 4. for every cluster center $j$ randomly open one of his close facilities with probabilities $\bar{x}_{ij}$
Step 5. for every facility $i$ that is not a close facility of any cluster center, open it randomly with probability $\bar{y}_i$
Step 6. connect each client to its closest open facility.
```

We begin by presenting the clustering method. The support graph of the *LP* solution is a bipartite graph G with vertex set $V = \mathcal{C} \cup \mathcal{F}$, where $\mathcal{C}$ and $\mathcal{F}$ are the client and facility set respectively, and an edge connecting the according vertices of $i \in \mathcal{F}$ and $j \in \mathcal{C}$ if $x_{ij} > 0$ in the *LP* optimal solution. Two clients j, j' adjacent to the same facility in G are called neighbours. *Clustering* is the partitioning of clients into clusters with a leading client for each cluster, called the cluster center. No two cluster centers can be neighbours in the support graph G.

Scaling the facility opening costs in the beginning and applying greedy augmentation helps to balance the analysis of an approximation algorithm -here, Chudak & Shmoys' $(1 + 2/e)$-approximation algorithm. Suppose we have a feasible solution for a metric UFL instance. For greedy augmentation the steps are:

- For each facility i not opened in the solution, compute the impact of opening this i on the total cost of the solution (the aforementioned *gain* of opening i,

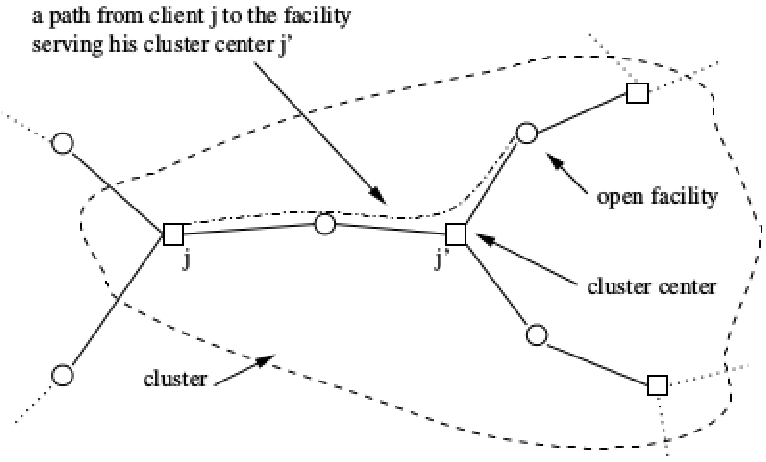
a path from client j to the facility serving his cluster center j'
open facility
j
j'
cluster center
cluster

Figure 1.5: image taken from [13]

now denoted as g_i).

- While $\exists i \in \mathcal{F} s.t.\ g_i > 0$, open a facility i_0 that maximizes $\frac{g_i}{f_i}$.
- Update values of g_i.

The procedure terminates when no facility's opening cost can decrease the total cost.

For a given approximation algorithm A for the metric UFL and a real $\delta \geq 1$, greedy augmentation together with scaling can be implemented resulting in the following algorithm $S_\delta(A)$:

```
Algorithm  $S_\delta(A)$ scaling procedure
1. scale up all facility opening costs by a factor $\delta$
2. run algo $A$ on the modified instance
3. scale back the opening costs
4. run the greedy augmentation
```

Byrka, following the analysis of Mahdian, Ye and Zhang [46] shows that if A is a (λ_f, λ_c)-approximation algorithm for the metric UFL, then $S_\delta(A)$ is a $(\lambda_f + ln(\delta), 1 + \frac{\lambda_c - 1}{\delta})$-approximation algorithm for this problem. Thus, the above method can be applied to balance instance where connection costs are greatly larger than facility costs. He then proceeds to construct a second algorithm via sparsening of the graph of the fractional solution, so as to tackle the opposite imbalance (i.e. when facility costs dominate the connection costs). The combination of the two algorithms achieves the 1,5 approximation.

The sparsening technique is similar to Lin and Vitter's [44] filtering techinque, a way of modifying the fractional solution. Suppose for the given LP of a metric UFL we have optimal primal solution (x^*, y^*) and optimal dual (α^*, β^*), where facility cost $F^* = \sum_{i \in \mathcal{F}} f_i y_i^*$, connection cost $C^* = \sum_{i \in \mathcal{F}, j \in \mathcal{C}} c_{ij} x_{ij}^*$ and each client j has its share a_j of the total cost. This cost can be divided into a client's fractional connection cost $C_j^* = \sum_{i \in \mathcal{F}} c_{ij} x_{ij}^*$ and his fractional facility cost $F_j^* = a_j^* - C_j^*$.

The main idea of the sparsening technique is to make use of some irregularities of an instance if they occur. An instance is considered regular if the facilities that fractionally serve a client j are all at the same distance from j. On such *regular* instances, using algorithm $S_\delta(A)$ with scaling and greedy augmentation the original $F^* + (1 + \frac{2}{e})C^*$ solution is dropped to an *optimal* 1.463 approximation. On *irregular* instances, particular clients are fractionally served by facilities at different distances. The sparsening techinque divides the serving facilities of a client into two groups: the *close* and the *distant* facilities. The links to distant facilities are removed $before$ the clustering step, so as to decrease distances to cluster centers. So, in the case of regular instances, the sparsening technique gives the same results as $S_\delta(A)$, but for irregular instances it takes some advantage of that irregularity.

Now, let $(\bar{x}, \bar{y})$ the obtained complete solution. For a client j the facility i is a *close* facility, if it fractionally serves j in $(\bar{x}, \bar{y})$, else it is a *distant* facility, if $\bar{x}_{ij} = 0$, but $x_{ij}^* > 0$. Byrka then proceeds to define $r_\gamma(j)$, a measure of the irregularity of the instance around client j. It is the average distance to *distant* facility minus the fractional connection cost C_j^* (which can also be perceived as the general average distance to both close and distant facilities), divided by the fractional facility cost of client j:

$$r_\gamma(j) = \begin{cases} \frac{\frac{\gamma}{\gamma-1}\left(\sum_{i \in \{i \in \mathcal{F} | \bar{x}_{ij} = 0\}} c_{ij} x_{ij}^* - C_j^*\right)}{F_j^*}, & \text{for } F_j^* > 0, \\ 0, & \text{for } F_j^* = 0 \end{cases}$$

where $0 \leq r_\gamma(j) \leq 1$. If $r_\gamma(j) = 0$, client j is served in (x^*, y^*) by facilities that are all in the same distance, else if $r_\gamma(j) = 1$, client j is served by facilities that are at different distances and the distant facilities are so far from j he is not willing

Algorithm Sparsening Procedure

Suppose we are given the primal optimal solution (x^*, y^*):

- scale up the y-variables by a constant $\gamma > 1$.
- with the y-variables fixed, the x-variables can be changed to minimize the total cost. Each client j uses his closest facilities in the following way:
  1. order facilities according to their nondecreasing distances to j.
  2. *fully* connect client j ($x_{ij} = y_{ij}$) to the first facilities in the ordering, with the possible exception of the last chosen one, for which $0 < x_{ij} < y_j$.
  3. Facilities can be *split* in order to assume the solution is *complete* (that is, $\nexists i \in \mathcal{F}, j \in \mathcal{C}\ s.t\ 0 < x_{ij} < y_i$)

to contribute to their opening. Let also $r'_\gamma(j) = r_\gamma(j) \cdot (\gamma - 1)$. For client j with $F_j^* > 0$, $r'_\gamma(j) = \frac{C_j^* - \sum_{i \in \mathcal{F}} c_{ij}\bar{x}_{ij}}{F_j^*}$, which is the fractional connection cost minus the average distance to a *close* facility, divided by the fractional facility cost of a client j. Then, for every client j it holds that:

- his average distance to a *close* facility equals $D_{av}^C(j) = C_j^* - r'_\gamma(j) \cdot F_j^*$,
- his average distance to a *distant* facility equals $D_{av}^D(j) = C_j^* + r_\gamma(j) \cdot F_j^*$,e
- his maximal distance to a *close* facility is at most the average distance to a distant facility: $D_{max}^C(j) \leq D_{av}^D(j) = C_j^* + r_\gamma(j) \cdot F_j^*$.

Considering the bipartite graph G obtained from $(\bar{x}, \bar{y})$, where each client is directly connected to his *close* facilities, Byrka greedily clusters this graph in each round, choosing as cluster center an *unclustered* client j with the minimal $D_{av}^C(j) + D_{max}^C(j)$. That way, each cluster center has a minimal value of $D_{av}^C(j) + D_{max}^C(j)$ among clients in his cluster.

Byrka describes an intermediate algorithm $A1(\gamma)$ as follows:

Algorithm $A1(\gamma = 1.67736)$ produces a solution with expected cost $E[cost(SOL)] \leq 1.67736 \cdot F^* + 1.37374 \cdot C^*$.

Finally, Byrka combines his $A1$ algorithm with the Jain et al. [32] 1.61-approximation algorithm for the metric UFL to obtain a 1.5-approx. The analysis is based on the following lemma by Mahdian, Ye & Zhang:

Algorithm $A1(\gamma)$

```
1. solve the $LP$ and obtain optimal primal solution $(x^*, y^*)$.
2. scale up the $y$-variables by a constant $\gamma > 1$
3. change value of $x$-variables so as to use closest possible fractionally open facilities
   - if necessary, split facilities to obtain a complete solution $(\bar{x}, \bar{y})$
4. compute a greedy clustering for $(\bar{x}, \bar{y})$, choosing as said before for cluster centers
   unclustered clients minimizing $D^C_{av}(j) + D^C_{max}(j)$
5. for every cluster center $j$, open one of his close facilities randomly with probabilities $\bar{x}_{ij}$
6. for each facility $i$ that is not a close facility of any cluster center, open it
   independently with probability $\bar{y}_i$
7. connect each client to an open facility that is closest to him.
```

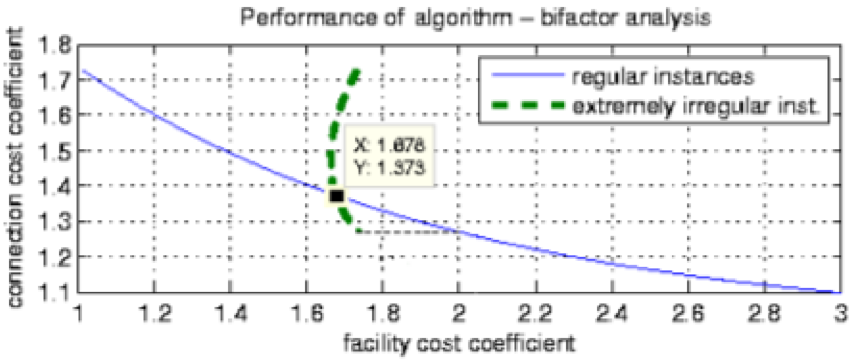

Figure 1.6: image taken from [13]

Lemma. *The cost of a solution return by the JMS algorithm is at most* $1.11 \cdot F^* + 1.7764 \cdot C^*$*, where* F^* *and* C^* *are the optimal solution's facility opening and connection costs, respectively, for the relaxed LP.*

So, considering the solutions obtained with the A1 and JMS algorithms, the cheaper of them is expected to have a cost at most 1.5 times the cost of the optional fractional solution.

There are some interesting remarks concerning the approximation limits. If the A2 algorithm combined the A1 algorithm with Mahdian, Ye & Zhang's 1.52-approximation algorithm for the metric UFL, then A2's approximation would drop to 1.4991. Byrka's algorithm also has improved results in the 3-level and 4-level facility location problem. While Byrka & Aardal constructed instances that are hard for the MYZ algorithm, construction of hard instances for the $A1(\gamma)$ algorithm still remains an open problem. Scaling and greedy augmentation enables to move the bifactor approximation guaranty of an algorithm along the approximability lower bound of Jain, Mahdian & Saberi towards higher facility opening costs. If a technique was developed to move the analysis in the opposite direction, together with the A1 algorithm, it would imply closing the approximation gap for the metric UFL. However, such an approach has the difficulty of analysing an algorithm that closes some of the previously opened facilities.

1.5.2 Shi Li's results

In 2011 Shi Li in [42] presented the best to date approximation algorithm on the UFL problem, achieving a 1.488-approximation. Shi Li based his result on Byrka's algorithm [13] and in fact reached this approximation by improving the analysis of Byrka's own algorithm. Byrka had used a γ parameter in his analysis for the A1 algorithm which he had opted to set to 1.6774. Shi Li elaborated on this and proved that if γ is randomly selected, the approximation ratio drops to 1.488. The novelty of his method lies in that he gave an explicit distribution for γ by introducing a 0-sum game.

Chapter 2

The Fault Tolerant Facility Location Problem

2.1 Preliminaries

So far, we have seen the case in the metric uncapacitated facility location problem where each client has *unit* demand. However, in a combinatorial approach presented in the previous section by Charikar & Guha [16], we had seen that the authors in the analysis of their algorithm had also accommodated the case where each client had a distinct *demand* d_j.

There are various settings from real-world problems where clients might have a demand to be connect to more than one facilities to cover their needs. Imagine for example a network, where facility nodes are servers with cached information and the clients are nodes with data requests. We want nodes to be able to access the data at any moment. In order to cover system failures, or cases where i.e. a server is down for maintenance, the client should have access to more than one servers. Another example would be that of facilities which are electric power supply stations and clients-consumers, such as hospitals for example of factories, which have high electric power consumption and need to reassure that they will be able to function non-stop even when a power supply station goes off, meaning that there is also a call for back-up suppliers.

The common element in all of the above real-life scenarios is the need to build resilience in the network and tackle efficiently provider failures. If each provider may fail with probability p and each client needs a guarantee to be served of at least q_j

probability, client j should be connected to at least $r_j = \lceil \log(1-q_j)/\log p \rceil$ distinct providers. This UFL variant, where each client j has a distinct request r_j, is called $fault-tolerant$ (from now on refered to as $FTFL$). The integral linear program formulation of the problem is as follows:

$$minimize \sum_{i\in\mathcal{F},j\in\mathcal{C}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i y_i$$

$$\begin{aligned}
\text{subject to:} \quad \sum_{i\in\mathcal{F}} x_{ij} &\geq r_j & \forall j\in\mathcal{C} \qquad (2.1)\\
y_i - x_{ij} &\geq 0 & \forall i\in\mathcal{F},\ j\in\mathcal{C} \qquad (2.2)\\
x_{ij} &\in \{0,1\} & \forall i\in\mathcal{F},\ j\in\mathcal{C} \qquad (2.3)\\
y_i &\in \{0,1\} & \forall i\in\mathcal{F} \qquad (2.4)
\end{aligned}$$

The LP-relaxation of the above program is:

$$minimize \sum_{i\in\mathcal{F},j\in\mathcal{C}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i y_i$$

$$\begin{aligned}
\text{subject to:} \quad \sum_{i\in\mathcal{F}} x_{ij} &\geq r_j & \forall j\in\mathcal{C} \qquad (2.5)\\
y_i - x_{ij} &\geq 0 & \forall i\in\mathcal{F},\ j\in\mathcal{C} \qquad (2.6)\\
x_{ij} &\geq 0 & \forall i\in\mathcal{F},\ j\in\mathcal{C} \qquad (2.7)\\
y_i &\geq 0 & \forall i\in\mathcal{F} \qquad (2.8)
\end{aligned}$$

The dual program is:

$$maximize \sum_{j\in\mathcal{C}} r_j\alpha_j - \sum_{i\in\mathcal{F}} z_i$$

$$\begin{aligned}
\text{subject to:} \quad \alpha_j - \beta_{ij} &\leq c_{ij} & \forall i\in\mathcal{F},\ j\in\mathcal{C} \qquad (2.9)\\
\sum_{j\in\mathcal{C}} \beta_{ij} &\leq f_i + z_i & \forall i\in\mathcal{F} \qquad (2.10)\\
\alpha_j &\geq 0 & \forall j\in\mathcal{C} \qquad (2.11)\\
\beta_{ij} &\geq 0 & \forall j\in\mathcal{C},\ i\in\mathcal{F} \qquad (2.12)
\end{aligned}$$

The same applies for the intuitive meaning and the correspondence of the variables used as in the case of the simple metric UFLP presented in the previous section. It should be noted that variable z_i which now appears in the fault-tolerant dual corresponds to the constraint $1 \geq y_i$ and intuitively means that facility i should not "overdo" it's opening $(1 \geq y_i \Rightarrow z_i \geq y_i z_i)$.

Unlike the simple version of the uncapacitated facility location problem, there is no inapproximability result for the fault-tolerant version. However, Byrka, Srinivasan & Swamy in [15] stated the conjecture that the fault-tolerant version has the same approximation threshold as UFL, that is 1.463.

2.2 Primal-dual Schemas

The first non-trivial approximation algorithm for the FTFL was given by Jain & Vazirani in [34]. Their algorithm is similar to an algorithm presented in [25] for the generalized Steiner network problem. It is proven to be a $3 \cdot H_k$-approximation, where k is the maximum requirement and H_k the kth harmonic number.

Once again, as in [35], Jain & Vazirani use a primal-dual schema. The algorithm runs in phases. Each phase p takes into consideration only those clients j who have reached requirement p and lowers (satisfies) their requirement by one unit by connecting them to open facilities, thus dropping their residual requirement to $p-1$. More specifically, the algorithm starts with an empty solution (I_k, C_k), where I_p is the set of $free$ facilities at the beginning of phase p and C_p the set of clients with residual requirement p. Each phase p returns a solution (I_{p-1}, C_{p-1}), where if a client j gets connected to an already opened facility i (meaning $i \in I_p$), we pay only for the connection costs, otherwise we also pay for the opening cost of i. In the first case, the free facility can only be used by clients which are not already connected to it. So, by defining as $C_p(j)$ the set of facilities to which j is already connected at the beginning of phase p we get the following intermediate programs:

$$minimize \sum_{i \in \mathcal{F}, j \in \mathcal{C}_p} c_{ij} x_{ij} + \sum_{i \in \mathcal{F} - I_p} f_i y_i$$

$$\text{subject to:} \quad \sum_{i \in \mathcal{F} - C_p(j)} x_{ij} \geq 1 \qquad \forall j \in \mathcal{C}_p \tag{2.13}$$

$$y_i - x_{ij} \geq 0 \qquad \forall i \in \mathcal{F} - I_p,\ j \in \mathcal{C}_p \tag{2.14}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{F},\ j \in \mathcal{C} \tag{2.15}$$

$$y_i \in \{0, 1\} \qquad \forall i \in \mathcal{F} - I_p \tag{2.16}$$

The LP-relaxation of the above program is:

$$minimize \sum_{i \in \mathcal{F}, j \in \mathcal{C}_p} c_{ij} x_{ij} + \sum_{i \in \mathcal{F} - I_p} f_i y_i$$

$$\text{subject to:} \quad \sum_{i \in \mathcal{F} - C_p(j)} x_{ij} \geq 1 \qquad \forall j \in \mathcal{C}_p \tag{2.17}$$

$$y_i - x_{ij} \geq 0 \qquad \forall i \in \mathcal{F} - I_p,\ j \in \mathcal{C}_p \tag{2.18}$$

$$x_{ij} \geq 0 \qquad \forall i \in \mathcal{F},\ j \in \mathcal{C} \tag{2.19}$$

$$y_i \geq 0 \qquad \forall i \in \mathcal{F} \tag{2.20}$$

The dual program is:

$$maximize \sum_{j \in \mathcal{C}} \alpha_j$$

$$\text{subject to:} \quad \alpha_j - \beta_{ij} \leq c_{ij} \qquad \forall i \in \mathcal{F} - I_p,\ j \in \mathcal{C}_p \tag{2.21}$$

$$\alpha_j \leq c_{ij} \qquad \forall i \in I_p,\ j \in C_p \tag{2.22}$$

$$\sum_{j \in \mathcal{C}} \beta_{ij} \leq f_i \qquad \forall i \in \mathcal{F} - I_p \tag{2.23}$$

$$\alpha_j \geq 0 \qquad \forall j \in \mathcal{C} \tag{2.24}$$

$$\beta_{ij} \geq 0 \qquad \forall j \in \mathcal{C},\ i \in \mathcal{F} \tag{2.25}$$

Theorem. *The optimum solution of the p-phase LP is at most OPT_f/p, where OPT_f is the optimal solution of the original problem's LP.*

Proof: Let the optimum solution of the p-phase LP be OPT_p. From the strong duality theorem, there is a dual feasible solution (α, β) for the p-phase dual of value OPT_p. (α, β) can be extended to a feasible solution for the original problem's LP of value $p \cdot OPT_p$ by the following transformations:

1. $\forall j \in C - C_p,\ \alpha_j := 0.$
2. $\forall j \in C - C_p, i \in F,\ \beta_{ij} := 0.$
3. $\forall j \in C_p, i \in C_p(j),\ \beta_{ij} := \alpha_j.$
4. $\forall i \in I_p,\ z_i = \sum_{j \in C_p} \beta_{ij}$ □

The p-phase algorithm is identical to the primal-dual algorithm for the UFL in [35], taking into account that only clients in C_p are considered, that free facilities, that is facilities in I_p, have no opening costs and that for a client $j \in C_p$ that is already connected to a facility $i \in I_p$ the connection cost is infinite, so as not to be reassigned to it.

The final algorithm runs in two stages. In the first stage, starting from phase $p := k$, it runs the p-phase algorithm until there are no more unconnected clients. So, we a have a (I_0, C_0) feasible solution, where I_0 is the set of temporarily open facilities. Similar to [35], there may be some clients *overpaying*, that is, they may have contributed towards the opening of more than one facilities. Exactly as in [35], a maximal set of the temporarily opened facilities is picked to be permanently opened, so as no client overpays. Following the steps in [35], clients are reassigned accordingly to the maximal set and the other facilities close. Because the method is identical to [35], this step guarantees that the new solution is at most thrice the original and combined with the theorem, it becomes clear that this is a $3 \cdot H_k$-approximation algorithm.

2.3 Combinatorial approaches

In [27], Guha, Meyerson & Munagala gave a constant factor approximation algorithm by using the filtering and decomposition technique of [53], guaranteeing a 3.16-approximation. Afterwards they reduced it to 2.408 by applying a greedy local improvement postprocessing step. However, their results were not on the strict FTFL problem, but rather for a more general version of it, where the service cost of a client j is a *weighted* sum of its distances to the r_j facilities to which it gets connected, weights being part of the input.

Swamy and Shmoys in [56] gave a combinatorial 4-approximation algorithm for the FTFL problem before proceeding to the construction of a more complex randomized version of it, yielding a 2.076-approximation. A key point to their analysis is the use of *complementary slackness conditions*. With the aid of these conditions they managed to overcome the presence of terms with negative coefficients, while maintaining the optimality properties of the dual solution.

Let (x, y) and (α, β, z) be the optimal primal and dual solutions, respectively, and OPT their value. The primal slackness conditions are: $x_{ij} > 0 \Rightarrow \alpha_j = \beta_{ij} + c_{ij}$ and $y_i > 0 \Rightarrow \sum_j \beta_{ij} = f_i + z_i$. The dual slackness conditions are: $\alpha_j > 0 \Rightarrow \sum_i x_{ij} = r_j$, $\beta_{ij} > 0 \Rightarrow x_{ij} = y_i$ and $z_i > 0 \Rightarrow y_i = 1$. Like Chudak & Shmoys's algorithm in [19] for the UFLP, Swamy & Shmoys observe that the optimal solution is α-close ($x_{ij} > 0 \Rightarrow c_{ij} < \alpha_j$), based on the slackness conditions. Although initially it seems that in order to reach an approximation, it is not enough to bound the cost by the term $\sum_j r_j \alpha_j$, due to the presence of the negative term $-\sum_i z_i$ in the dual objective, the authors observe that slackness condition $z_i > 0 \Rightarrow y_i = 1$ implies that all those facilities i for which the condition applies can be opened and thus add the opening cost to the LP.

Considering each client j with requirement r_j as consisting of r_j distinct copies, the algorithm consists of two phases, the allowing exploitation of the optimal LP solution due to the slackness conditions and the second phase constructing the clusters:

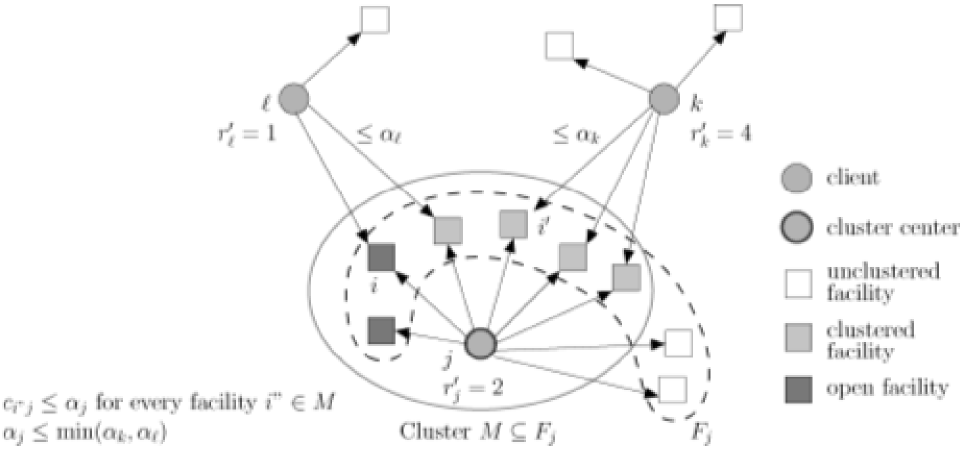

Figure 2.1: Clustering step in phase 2 with j the cluster center. In this iteration 2 copies of j, 2 copies of k and 1 copy of l get connected. j and are l removed from the input after this iteration - image taken from [56]

The cost of Phase 1 is bounded by $\sum_j n_j \alpha_j - \sum_i z_i$. The proof is based on complementary slackness conditions and specially on condition $z_i > 0 \Rightarrow y_i = 1$,

Algorithm

Phase 1: Open all facilities with $y_i = 1$ and define the set of these facilities as L_1. For every client j, if $x_{ij} > 0$ and $y_i = 1$, connect exactly one copy of j to i. Let $|n_j|$ be the number of the copies of j which got connected in this phase.

Phase 2: For every client j, let $r'_j = r_j - n_j$ the residual requirement of j and $F_j = \{i : \; y_i < 1, x_{ij} > 0\}$ the set of facilities not in L_1 that fractionally serve j. Let $\mathcal{S} = \{j : \; r'_j \geq 1\}$. Repeat the following steps until $\mathcal{S} = \emptyset$:

1. Pick $j \in \mathcal{S}$ with minimum a_j as cluster center.

2. Order facilities in F_j by increasing facility cost. Starting from the first facility in F_j, pick $M \subseteq F_j$ such that $\sum_{i' \in M} y_{i'} \geq r'_j$. If $\sum_{i' \in M} y_{i'} > r'_j$, split the last facility i in M, that is the farthest facility which serves j, in two copies i_1 and i_2. Set $y_{i_1} = r'_j - \sum_{i' \in M \setminus \{i\}} y_{i'}$ and $y_{i_2} = y_i - y_{i_1}$. For every client k, including j, with $x_{ik} > 0$ set arbitrarily $x_{i_1 k}$, $x_{i_2 k}$ such that $x_{i_1 k} + x_{i_2 k} = x_{ik}$, $x_{i_1 k} \leq y_{i_1}$, $x_{i_2 k} \leq y_{i_2}$. Include only i_1 in M, so that now $\sum_{i' \in M} y_{i'} = r'_j$.

3. Open the r'_j least expensive facilities in M. For each client k, including j with $F_k \cap M \neq \emptyset$, connect $\min(r'_k, r'_j)$ copies of k to these newly opened facilities and set $r'_k := r'_k - \min(r'_k, r_j)$ and $F_k := F_k \backslash M$. Remove j and facilities in M from the process.

meaning that each such i is in L_1.

The facility opening cost in Phase 2 is bounded by $\sum_i f_i y_i$, which is straightforward since at each iteration, at most the r'_j least expensive facilities are opened at step 2. During Phase 2, if $k^{(c)}$ is a copy of k connected to facility i, then $c_{ik} \leq 3\alpha_k$.

By adding the cost of Phase 1, the facility opening cost of Phase 2 and the connection cost of each of the r'_j copies connected in Phase 2, it follows that the above algorithm returns a solution of cost at most $4 \cdot OPT$.

2.4 A Dependent Rounding algorithm

As stated previously, Swamy and Shmoys in [56] presented a randomized version of their original 4-approximation combinatorial algorithm for the FTFL problem. The randomized version gave a 2.076-approximation and was based on clustered randomized rounding, a technique that was first used by Chudak and Shmoys in [19].

In [15] presented a 1.725-appoximation algorithm, which gives the best to date approximation for the FTFL problem. It is a randomized dependent LP-rounding algorithm, expanding on the clustering and introducing a novel *hierachical* clustering method, based on laminarity properties. They were also the first to apply the dependent rounding technique on a facility location problem. As a note, dependent rounded is a very promising and adaptable technique, first employed by Gandhi, Khuller, Parthasarathy & Srinivasan in [22] on bipartite graphs and later generalized by the same authors in [23] to encompass a broader range of approximation problems. Below follows a quick overview of this techique.

The dependent rounding techique is, in essence, a polynomial-time randomized algorithm which takes as input a $fractional$ vector $y = (y_1, y_2, ..., y_N) \in [0, 1]^N$ and returns a *random* vector $\hat{y} \in \{0, 1\}^N$ satisfying the following properties:

- **(P1): marginals.** $\forall i$, $\Pr[\hat{y}_i = 1] = y_i$
- **(P2): sum-preservation.** $\Pr[\sum_{i=1}^{N} \hat{y}_i = \lfloor \sum_{i=1}^{N} y_i \rfloor \; or \; \lceil \sum_{i=1}^{N} y_i \rceil] = 1$
- **(P3): negative correlation.** $\forall S \subseteq [N], \Pr[\bigwedge_{i \in S}(\hat{y}_i = 0)] \leq \prod_{i \in S}(1 - y_i)$, and $\Pr[\bigwedge_{i \in S}(\hat{y}_i = 1)] \leq \prod_{i \in S} y_i$

Byrka et al. devised a hierarchical clustering method, which returns a laminar family of subsets of facilities[1], in order to ensure property **(P2)** and thus apply the dependent rounding techinque. So, since in the case of the presented algorithm, the family of subsets of indices ($facilities$) $S \subseteq 2^{[N]}$ is laminar:

(P2'): sum-preservation. $\Pr[\sum_{i \in S} \hat{y}_i = \sum_{i \in S} y_i] = 1$ and $|\{i \in S : \hat{y}_i = 1\}| = \lfloor \sum_{i \in S} y_i \rfloor$.

Below follows a short overview of the final algorithm:

Algorithm Overview

Step 1. Solve the LP-relaxation.

Step 2. Scale the fractional solution.

Step 3. Create a laminar family of clusters (containing *close* facilities).

Step 4. Round the fractional openings y_i via dependent rounding.

Step 5. Connect each client j to r_j closest open facilities.

Step 6. Output the solution as $(\tilde{x}, \tilde{y})$.

We proceed by a more thorough presentation, starting from the scaling Step 2. Given *(x*,y*)* to be the OPT solution of the LP, set $\hat{x}_{ij} = \min\{1, \gamma \cdot x^*_{ij}\}$ and $\hat{y}_i = \min\{1, \gamma \cdot y^*_i\}$. For each facility i with $\hat{y}_i = 1$, set $\hat{y}_i = 0$ and $\tilde{y}_i = 1$ and for each pair (i, j) such that. $\hat{x}_{ij} = 1$, set $\hat{x}_{ij} = 0$ and $\tilde{x}_{ij} = 1$ and decrease r_j by one. When these transformations are completed, call the resulting $r, \hat{y}$ and $\hat{x}$ by $\bar{r}, \bar{y}$ and $\bar{x}$.

From now on, the algorithm only tackles $\bar{r}, \bar{y}$ and $\bar{x}$. If a client j connected in this initial phase to a facility i, care is taken so that it will not be re-connected throughout the rest of the algorithm. Define a facility i as *special* for client j $\tilde{y}_i = 1$ and $0 < \bar{x}_{ij} < 1$. There is *at most* one special facility for each client j and it will be at maximal distance among facilities serving j in $\bar{x}_{ij}$.

Once again, Byrka et al. as in [13] use the concept of *close* and *distant* facilities. Because of the scaling, for all clients $j \in \mathcal{C}$ it holds that $\sum_{i \in \mathcal{F}} \bar{x}_{ij} \geq \gamma \cdot \bar{r}_j$. Let

[1] A laminar family $\mathcal{F} \subseteq 2^V$ is a family of subsets of V such that for any $A, B \in \mathcal{F}$:

$$A \subseteq B \text{ or } B \subseteq A \text{ or } A \cap B = \emptyset \ .$$

$i_1, i_2, ..., i_{|\mathcal{F}|}$ be an increasing ordering of the distances c_{ij} to client j. Let i_k such that $\sum_{l=1}^{k-1} \bar{x}_{i_l j} < \bar{r}_j$ and $\sum_{l=1}^{k} \bar{x}_{i_l j} \geq \bar{r}_j$.

$$\bar{x}^{(c)}_{i_l j} = \begin{cases} \bar{x}_{i_l j}, & \text{for } l < k, \\ \bar{r}_j - \sum_{l=1}^{k-1} \bar{x}_{i_l j}, & \text{for } l = k, \\ 0, & \text{for } l > k \end{cases}$$

where $\bar{x}^{(d)}_{i_l j} = \bar{x}_{ij} - \bar{x}^{(c)}_{i_l j}$. Then:

- The set of *close* facilities of j is $C_j = \{i | i \in \mathcal{F} \text{ s.t. } \bar{x}^{(c)}_{ij} > 0\}$.
- The set of distant facilities of j is $D_j = \{i | i \in \mathcal{F} \text{ s.t. } \bar{x}^{(d)}_{ij} > 0\}$.

Afterwards a family $\mathcal{S} \in 2^{\mathcal{F}}$ of subsets of facilities is constructed, where each subset $S \in \mathcal{S}$ is a cluster and each client j is related to *at most* one cluster (S_j). Clients j with $\bar{r}_j = 1$ and a *special* facility do *not* take part in the clustering process, while the remaining clients belong to $\mathcal{C}'$. For each $j \in \mathcal{C}'$, two families A_j and B_j of *disjoint* subsets of facilities are considered, where A_j stores clusters containing only close facilities of j and B_j stores only clusters that contain at least one close facility of j. Initially, $A_j = \{\{i\} \;:\; i \in C_j\}$ and $B_j = \emptyset$. Subsets in $A_j \cup B_j$ will always be pairwise disjoint. Thus, the following routine describes the hierarchical clustering method:

Algorithm of hierarchical clustering method

```
While exists $j \in \mathcal{C}'$ such that $\bar{r}_j - \sum_{S \in (A_j \cup B_j)} \sum_{i \in S} \bar{y}_i \; (\equiv rr_j) > 0$, take $j$ with minimal distance from the farthest of its close facilities ($d_j^{(max)}$) and do:
    1. Take $X_j$ minimal subset of $A_j$ such that $\sum_{S \in X_j} (\sum_{i \in S} \bar{y}_i - \lfloor \sum_{i \in S} \bar{y}_i \rfloor) \geq rr_j$.
       Form new cluster $S_j = \bigcup_{S \in X_j} S$ and $\mathcal{S} \longleftarrow \mathcal{S} \cup \{S_j\}$.
    2. Update $A_j \longleftarrow (A_j \setminus X_j) \cup \{S_j\}$.
    3. For each $j'$ with $rr_{j'} > 0$ do:
        • If $X_j \subseteq A_{j'}$, then set $A_{j'} \longleftarrow (A_{j'} \setminus X_j) \cup \{S_j\}$.
        • If $X_j \cap A_{j'} \neq \emptyset$ and $X_j \setminus A_{j'} \neq \emptyset$, then set
          $A_{j'} \longleftarrow A_{j'} \setminus X_j$ and $B_{j'} \longleftarrow \{S \in B_{j'} : S \cap S_j = \emptyset\} \cup \{S_j\}$
```

Therefore, when a new cluster S_j is created, it becomes the root of a new tree in the laminar family. Due to triangle inequality and way of creating B_j from j' with $d_{j'}^{(max)} \leq d_j^{(max)}$, the following lemma holds:

Lemma 2.4.1. *The family of clusters S contains for each client $j \in \mathcal{C}'$ a collection of disjoint clusters $A_j \cup B_j$ containing only facilities within distance $3 \cdot d_j^{(max)}$ and $\sum_{S \in A_j \cup B_j} \lfloor \sum_{i \in S} \bar{y}_i \rfloor \geq \bar{r}_j$.*

Consequently, the algorithm can proceed to **Step 4.**, which opens the facilities via dependent rounding, applying the following routine:

```
Algorithm  for opening facilities via dependent rounding
  1. While there is more than one fractional entry:
      (a) Select minimal subset $S \in \mathcal{S}$ that contains more than one fractional entry.
      (b) Apply the rounding procedure to entries of $\bar{y}$ indexed by elements of $S$
          until at most one entry in $S$ remains fractional.
  2. If there remanins a fractional entry, round it independently and let $y^R$ be the
     resulting vector.
  3. Combine the results: $\tilde{y} := \tilde{y} + y^R$.
```

In the end, connect each client $j \in \mathcal{C}$ to r_j closest opened facilities and code it in $\tilde{x}$. The above algorithm guarantees a 1.7245-approximation for the FTFL problem and the key to the analysis of bounding solution $(\tilde{x}, \tilde{y})$ is that for a client j it is possible that there is facility both close *and* distant. Once such a facility is opened, it is vital to know the fraction of the demand that is served from the close facilities. To achieve this, the authors in their proof toss an unbiased coin to decide if using this facility counts as using a close facility.

Also, the writers developed two theorems useful not only for the analysis of this specific algorithm but for fault-tolerant related objectives in general:

Consider a real vector $\lambda = (\lambda_0, \lambda_1, \lambda_2, ..., \lambda_{|S|})$ and any $x \in \{0,1\}^n$ such that $g_{\lambda,S}(x) = \lambda_i$, where $i = \mathrm{Sum}_S(x)$ (that is, the number of entries in $\{x_j : j \in S\}$ that are 1). Let $\mathcal{R}(y)$ be a random vector in $\{0,1\}^N$ obtained by *independently* rounding each y_i to 1 with probability y_i and to 0 with probability $1 - y_i$.Define $\hat{y}$ the vector obtained via the *dependent* rounding. Then the following holds true:

Theorem 1. *Suppose we conduct dependent rounding on* $y = (y_1, y_2, ..., y_N)$*. Let* $S \subseteq [N]$ *be any subset with cardinality* $s \geq 2$*, and let* $\lambda = (\lambda_0, \lambda_1, \lambda_2, ..., \lambda_s)$ *be any vector such that for all* r *with* $0 \leq r \leq s-2$ *we have* $\lambda_r - 2\lambda_{r+1} + \lambda_{r+2} \leq 0$*.*
Then, $\mathbf{E}[g_{\lambda,S}(\hat{y})] \geq \mathbf{E}[g_{\lambda,S}(\mathcal{R}(y)]$*.*

In other words, after dependent rounding, more elements of vector y will have been rounded to 1. Also:

Theorem 2. *For any* $y \in [0,1]^N$*,* $S \subseteq [N]$*, and* k *= 1,2,..., we have*

$$\mathbf{E}[\min\{k, Sum_S(\hat{y})\}] \geq \mathbf{E}[\min\{k, Sum_S(\mathcal{R}(y)\}].$$

Intuitively, this final result is important, because it combines the tractability of independence with the benefits of dependent rounding, thus furthering the potential applications of the dependent-rounding technique. By generalizing the result and considering S as an arbitrary subset of the *dependently* rounded variables, all either 0 or 1, and an arbitrary integer $k > 0$, then in fault-tolerant settings, such as one where $X = |s_i \in S : s_i = 1|$ the number of the variables rounded to 1, and $Z = \min\{k, X\}$ the random variable which we wish to "maximize" without violating the "constraint" that $X \leq k$, it is possible to replace X with X_0, where X_0 is the number of how many variables would be rounded to 1 *independently*. Subsequently, if $Z_0 = \min\{k, X_0\}$, then $\mathbf{E}[Z] \geq \mathbf{E}[Z_0]$, which allows working with Z_0 instead of Z in bounding analyses etc., where Z_0 can be handled more easily due to the *independence*, while preserving the desired properties of dependent rounding.

Chapter 3

Other variants of Facility Location Problems

Facility Location problems hold a prominent position amongst combinatorial optimization, not only because of their challenging mathematical nature, but also because they are closely related to many real world network design problems and needs of the industry. Thus, there have arisen numerous variants of facility location problems, each inspired by and trying to model a more specific real world scenario.

For example, it may not be required that all clients get connected to a facility. In the case a client's demand is not satisfied, a penalty (fee) is imposed. In [43] the authors deal with linear and sublinear penalty costs, a case which we will presented more thoroughly in this work.

Other variants impose limits on the resources. For instance, in the well-acknowledged Capacitated Facility Location Problem, each facility i now can has an upper limit u_i on the quantity of products it can produce or, in other words, the units of service it can offer to the clients' demands. There are versions where this capacity is common for all the facilities (i.e. [5]) or where one can open multiple copies of any facility (i.e. [20]). Barahona and Jensen from IBM in [11] tackle another variant, which was inspired by a real life parts warehousing problem. The special case where there is no upper bound on any facility's capacity ($u_i = \infty$, $\forall i \in F$) is, in fact, the classical Uncapacited Facility Location Problem.

Furthermore, there are models which require a hierarchy in the structure of their solution, in the sense that facilities (or the demanded *service*) are now divided into *levels*, where each level (i.e. type of facility or connection type) may serve a different need of a client or in order for a client's demand of level k it is prerequisite that another need of level $k-1$ is covered by another facility/connection of the appropriate type, as for example may be the case in supply chains or, even more prominently, a cost-effective placement of servers in a network. An extensive study by Sahin & Süral [51] groups the various hierarchical facility location problems into categories depending on: *i*) the flow pattern, *ii*) the service varieties, *iii*) spatial configuration and *iv*) the objective. In this work we shall examine the simple k-level Facility Location Problem.

A presentation follows of some of the most noteworthy or representative variants as sample of the vast field covered by facility location problems.

3.1 Universal Facility Location

The Universal Facility Location Problem can be considered as variant of the Capacitated Facility Location Problem and was first introduced by Mahdian & Pál in [45], alongside a $(7.88+\epsilon)$-approximation algorithm for the problem. The problem, in essence, captures those cases where the opening cost of one or more installations on a certain site depends on the location and is formally described as follows:

Typically, let $\mathcal{C}$ be the set of clients and $\mathcal{F}$ the set of facilities, with $n = |\mathcal{C}|$ and $m = \mathcal{F}$. Distances $d(i,j)$ between client j and facility i follow the triangle inequality and $d(i,i')$ is the length of the shortest graph between facilities i and i'. The key difference from other models lies, as mentioned, in the definition of opening costs: the associated opening cost for facility i is now a non-decreasing $function$ $f_i : \mathbb{N} \to \mathbb{R}^+$, $f_i(0) = 0$. In order to serve clients, capacities u are installed at each facility site. A solution $S = (u, x)$, where $x_{ij} = 1$ if client j is connected to facility i;0 otherwise, and $u_i \in \mathbb{N}$ the capacity allocated to facility i, or equivalently the number of clients connected to it, incurs a connection cost $C_s(S) = \sum_{i\in\mathcal{F},j\in\mathcal{C}} d(i,j)x_{ij}$ and a facility cost $C_f(S) = \sum_{i\in\mathcal{F}} f_i(u_i)$. The goal is to find a feasible solution minimizing the total cost $C(S) = C_s(S) + C_f(S)$.

In [7] Angel et.al. present the best known to date $(5.83+\epsilon)$ approximation algorithm for the problem, although in the case of concave facility costs Hajiaghayi et.al. in [28] give a 1.861-approximation algorithm. Angel et.al. algorithm is based on local search and stands out from previous local search algorithms on the problem in defining a new, polynomially computable operation called $Open-close$. In short, their algorithm uses the below operations:

- $Add(s,\delta)$: increase the capacity u_s of facility s by δ and find the minimum cost assignment of demands to facilities, given their capacities.

- *Open*(s, δ): increase capacity of s by sending δ units of flow from one or more facilities $i_1, i_2, ..$ to s via the shortest paths between $i_1, i_2, ..$ and s and decrease the capacity of $i_1, i_2, ...$

- *Close*(s, δ): decrease capacity of s by sending δ units of flow from s to one or more facilities $i_1, i_2, ..$ via the shortest paths between $i_1, i_2, ..$ and s and increase the capacity of $i_1, i_2, ...$

- *Open−close*$(s, t, \delta_s, \delta_t)$: increase the capacity of s by δ and decrease the capacity of t by δ. This operation also includes the exchange of flow units to and from other facilities $i_1, i_2, ...$ from and to, respectively, t and s, again via shortest paths.

Each operation takes polynomial time. Let S be an arbitrary feasible solution. When no improvement in the solution can be made in order for the cost to by reduced by at least $\epsilon C(S)$, where $\epsilon > 0$, the algorithm returns S. The process halts after at most $\frac{1}{\epsilon} \log \frac{C(S)}{C(S*)}$ where $S*$ is a global minimum. Although the solution returned by the algorithm is not a local optimum, it is only $(1 + \epsilon)$ factor worse than the bound of a local optimum. Thus, if the local optimum, according to the aforementioned operations, can by bound by r times the global optimum, then the approximation ratio of the algorithm is $r(1 + \epsilon)$.

3.2 Facility Location with Penalties

The Facility Location Problem with penalties was first introduced in [17] for linear penalties (FLPLP), and in [29] for submodular penalties(FLPSP), where a set function $P : 2^{\mathcal{D}} \rightarrow \mathbb{R}_+$ is considered submodular if $P(X \cap Y) + P(X \cup Y) \leq P(X) + P(Y)$, with a primal-dual 3-approximation and an LP-rounding 2.5-approximation algorithm, respectively. Facility Location with penalties follows exactly the same formulation as the simple Uncapacitated Facility Location problem, with the exception that it is not required that all clients are connected, incurring in that case a p_j penalty cost for each client j that does not get connected. The goal is to find a solution minimizing the total opening, connection and linear/submodular penalty costs.

In 2013 Li et al. in [43] presented that best to date approximation algorithms for both problems, yielding a 2-approximation ratio for the FLSP and a 1.5148-approximation ratio for the FLPLP. In this work, they generalized results of Geunes et al. [24] on linear penalties to a framework for a class of covering problems (FLPs can be considered falling in this category) with submodular penalties, where any LP-based α-approximation for the original problem can be converted to a $(1-\epsilon^{-1/\alpha})^{-1}$-approximation algorithm for the counterpart with submodular penalties.

It should be noted that FLPSP and FLPLP are substantial different in their essence, because linear penalty functions have properties which cannot be exploited in the submodular case. Thus, algorithms for the FLPLP cannot be directly applied to the FLPSP and the latter is harder to approximate than the former.

3.2.1 A rounding approximation for the FLPSP

The FLPSP can be described by the following LP relaxation:

$$\textit{minimize} \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{D}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i y_i + \sum_{S\subseteq\mathcal{D}} P(S)z_S$$

$$\begin{aligned}
\text{subject to:} \quad & \sum_{i\in\mathcal{F}} x_{ij} + \sum_{S\subseteq\mathcal{D}:j\in S} z_S \geq 1 && \forall j\in\mathcal{D} && (3.1)\\
& y_i - x_{ij} \geq 0 && \forall i\in\mathcal{F},\ j\in\mathcal{D} && (3.2)\\
& x_{ij}, y_i, z_S \geq 0 && \forall i\in\mathcal{F},\ j\in\mathcal{D}, S\subseteq\mathcal{D} && (3.3)\\
& && && (3.4)
\end{aligned}$$

where $P(S)$ a nondecreasing submodular function and $P(\emptyset)=0$. Define (x^*,y^*,z^*) the optimal fractional solution. The author in [43] consider the more general model which captures a class of covering problems:

$$\textit{minimize}\ \varphi(w) + \sum_{S\subseteq\mathcal{D}} P(S)z_S$$

$$\begin{aligned}
\text{subject to:} \quad & w_j + \sum_{S\subseteq\mathcal{D}:j\in S} z_S \geq 1 && \forall j\in\mathcal{D} && (3.5)\\
& w_j, z_S \in \{0,1\} && \forall j\in\mathcal{D}, S\subseteq\mathcal{D} && (3.6)
\end{aligned}$$

where w_j denotes whether client j gets connected or not, S being the subset of the rejected clients and $P(S)$ the respective penalty cost. Term φ is an embedding of subproblem ϕ satisfying the following assumption:

Assumption 1. *There exists a function* $\bar{\varphi} : [0,1]^{n_c} \mapsto R_+$ *such that:*

1. *$\bar{\varphi}$ is a lower bound on φ;*

2. *for any fixed $w \in \{0,1\}^{n_c}$, there can be efficiently found a solution to $\phi(w)$ of cost at most $\alpha\bar{\varphi}(w)$, where $\alpha \geq 1$;*

3. *the following optimization problem can be solved efficiently:*

$$\textit{minimize } \bar{\varphi}(w) + \sum_{S \subseteq \mathcal{D}} P(S) z_S$$

$$\begin{aligned} \textit{subject to:} \quad & w_j + \sum_{S \subseteq \mathcal{D}: j \in S} z_S = 1 & \forall j \in \mathcal{D} \qquad (3.7) \\ & w_j, z_S \in [0,1] & \forall j \in \mathcal{D}, S \subseteq \mathcal{D} \qquad (3.8) \end{aligned}$$

Another assumption, based on the *scaling* property from [24], necessary for the central theorem of the paper:

Assumption 2. *The function $\bar{\varphi}(w)$ satisfies the scaling property if*

$$\bar{\varphi}(w) \leq \frac{1}{1-\beta}\bar{\varphi}(w^*), \ \forall w^* \in [0,1]^{n_c}, \forall - \leq \beta < 1.$$

The following rounding algorithm, which is also based on Geunes et al. [24], is one of the important results of the paper and serves as the, earlier mentioned, *general algorithmic frame* for a class of covering problems with submodular penalties, yielding an expected penalty cost of no more than $\delta^{-1} \sum_{S \subseteq \mathcal{D}} P(S) z_S*$:

Algorithm 1

1. Solve the LP relaxation of *Assumption* 1 with optimal fractional solution w^*, z^*.

2. Select parameter β uniformly at random from interval $[0, \delta)$.

3. Reject the subset $S := \{j | 1 - w_j^* \geq \beta\}$, paying therefore the penalty cost $P(S)$. Construct variable $w \in \{0,1\}^{n_c}$ by setting $w := I(\mathcal{D} \backslash S)$.

4. Find a solution to subproblem $\phi(w)$ and cover (serve) the remaining unrejected elements (clients) in $\mathcal{D} \backslash S$.

Theorem. *For $\delta = 1 - \epsilon^{-1/\alpha}$, the approximation ratio of Algorithm 1 is no more than $(1 - \epsilon^{-1/\alpha})^{-1}$.*

The subproblem $\varphi(w)$ is generally NP-hard, so in the case of FLPSP, $\varphi(w)$ is the simple UFLP. So, if $\varphi(w)$ is approximated by the best known to date 1.488-approximation algorithm for the UFL from [42], then Algorithm 1 approximates the FLPSP for no more than $(1 - \epsilon^{-1/\alpha})^{-1} \leq 2.044$. However, Algorithm 1 in its current generic state ignores the fact that some unrejected clients may have paid fractional cost for penalty. By including this case in the algorithm for the FLPSP and applying the greedy algorithm from Jain et al. [32] instead of Li's, the approximation ratio drops down to 2.

3.2.2 A rounding approximation for the FLPLP

The FLPLP is described by the following LP-relaxation:

$$\textit{minimize} \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{D}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i y_i + \sum_{j\in\mathcal{D}} p_j z_j$$

$$\begin{aligned} \text{subject to:} \quad & \sum_{i\in\mathcal{F}} x_{ij} + z_j \geq 1 & \forall j \in \mathcal{D} \qquad (3.9)\\ & y_i - x_{ij} \geq 0 & \forall i \in \mathcal{F},\ j \in \mathcal{D} \qquad (3.10)\\ & x_{ij}, y_i, z_j \geq 0 & \forall i \in \mathcal{F},\ j \in \mathcal{D} \qquad (3.11) \end{aligned}$$

where z_j denotes whether client j is connected or not. Let (x^*, y^*, z^*) be the optimal solution to the LP. The authors prove the, intuitively logical, lemma, that for every client j, if $z_j^* > 0$ and $x_{ij}^* > 0$, then $p_j \geq c_{ij}$ - that is, if a client j is (partially) served by a facilty i and in the same time is (partially) excluded from the solution, then its connection cost is at most as large as its penalty (otherwise, if the connection cost was strictly greater than the penalty, we could have excluded that client to obtain a better than optimal solution, which is absurd). By following closely Li's analysis on a rounding algorithm which is very similar to Byrka's [13], with clustering and discerning the facilities into *close* and *distant*, just as Byrka, and differentiating only the choice of cluster centers (for the FLPLP only unclustered clients in $\mathcal{D}_\gamma = \left\{ j \in \mathcal{D} | \sum_{i\in\mathcal{F}} x_{ij}^* \geq \frac{1}{\gamma} \right\}$ are considered for cluster centers), then by following Li's analysis for γ from [42], the authors arrive to a 1.5148-approximation ratio for the FLPLP.

3.3 K-level Facility Location

Another variant of the standard UFLP is the k-level Facility Location Problem, formally described for the first time in [3] by Aardal, Chudak & Shmoys in 1999. Interestingly enough, the algorithm proposed in that paper, a 3-approximation LP-rounding algorithm, still remains the best for the general k case. However, the algorithm includes a step where a LP relaxation with exponential number of variables needs to be solved and, even if the ellipsoid method is applied, still the implementation of the algorithm is impractical. Therefore, there have been attempts to devise combinatorial approximations for the algorithm, which although fall behind the initial algorithm in aspect of approximation ratio, they ran at least in strongly polynomial time. The combinatorial algorithm which so far yields the best results was proposed in [4]. In the general case where k tends to ∞, the approximation factor tends to 3.25, while for the special cases where $k = 2$ and $k = 3$ it succeeds an approximation guarantee even better than that of Aardal, Chudak & Shmoys, with a guarantee of roughly 2.4211 and 2.8446 respectively.

In [39] Krishnaswamy & Sviridenko prove that there is no polynomial algorithm for the k-level uncapacitated facility location problem with approximation ratio less than 1.61 unless $NP \subseteq DTIME(n^{O(\log \log n)})$. Specifically for the case where $k = 2$ the best attainable ratio becomes 1.539, thus proving that the problem is computationally harder than the simple UFLP.

The k-level Facility Location Problem can be described as follows: A *complete* $(k+1)$-partite graph $G = (\mathcal{D} \cup F_1 \cup F_2 \cup ... \cup F_k, E)$ is given, where the node set V is the union of the $\mathcal{D}, F_1, F_2, ..., F_k$ *disjoint* sets and edge set E contains all the edges between these sets. The nodes in $\mathcal{D}$ are the clients and the nodes in $\mathcal{F} = F_1 \cup F_2 \cup ... \cup F_k$ are the facilities of level 1,2,...,k respectively. Connection costs are induced by a metric c on V while opening costs follow the common pattern. The goal is to open a subset $X_t \subseteq F_k$ for each level $t \in \{1, 2, ..., k\}$ and to connect each client $j \in \mathcal{D}$ to a *chain* $\varphi(j) = i_1(j), i_2(j), ...i_k(j)$, where $i_t(j) \in X_t$, minimizing the total cost $\sum_{i \in X_1 \cup X_2 \cup ... \cup X_k} f_i + \sum_{j \in \mathcal{D}} \big(c(j, i_1(j) + c(i_1(j), i_2(j) + ... + c(i_{k-1}(j), i_k(j))\big)$.

It is interesting to note that there exists a variation of the k-level Facility Location with imposed penalties, presented in [9] and accompanied by a 4-approximation algorithm.

3.4 Fault-Tolerant Facility Allocation

The Fault-Tolerant Facility Allocation Problem (FTFAP) is a generalization of the classical FTFLP and is first presented in [52]. This variant resembles the fault-tolerant version, in the sense that each client j has a demand r_j, and differentiates itself in the aspect that on each site an *unlimited* number of facilities can be opened. Thus, the following formulation for the problem:

$$minimize \sum_{i\in\mathcal{F},j\in\mathcal{C}} c_{ij}x_{ij} + \sum_{i\in\mathcal{F}} f_i y_i$$

$$\text{subject to:} \quad \sum_{i\in\mathcal{F}} x_{ij} \geq r_j \qquad \forall j \in \mathcal{C} \tag{3.12}$$

$$y_i - x_{ij} \geq 0 \qquad \forall i \in \mathcal{F},\ j \in \mathcal{C} \tag{3.13}$$

$$x_{ij},\ y_i \in \mathbb{Z}^+ \qquad \forall i \in \mathcal{F},\ j \in \mathcal{C} \tag{3.14}$$

A practical application of the FTFA problem is in surrogate server deployment in a content distribution network, i.e. the installation of multiple ATMs at one site in order to serve multiple clients simultaneously. The FTFA is less restricted than the FTFL, given that it allows more that one facilities to be opened per site, thus incurring a smaller total cost. In the case where the demand is equal to 1 for each client j, the problem is identical to the classical UFL problem.

The authors prove that $UFL \subseteq FTFA \subseteq FTFL$. For the second inclusion they consider the FTFA as a special case of an FTFL instance, where each site has R copies of the facility i on that site, R the maximum demand r_j amongst all clients, or more formally, as a set $\mathcal{F}'$ of facilities distributed by groups such that $\mathcal{F}' = \mathcal{F} \times \{1, 2, ..., R\}$. It is observed that, although any FTFA instance could be handled as a FTFL one via the above transformation and therefore employ any known FTFL algorithm, the authors prefer to treat the FTFA instance as a transformed UFL one, because algorithms for the latter can deliver a better approximation ratio than the ones for the fault-tolerant setting. To attain the transformed UFL instance, as facility set they consider $\mathcal{F}'$ and as client set $\mathcal{C}' = \{(j,p), j \in \mathcal{C}, 1 \leq p \leq r_j\}$, where (j,p) the p-th port of city j, under the restriction that different ports of a city must be connected with different facilities. Because this constraint is nontrivial, FTFA is harder to solve than UFL, hence leading to the first inclusion. The authors target their work on how to tackle this constraint in order to derive better approximation for the FTFA compared to the FTFL.

The aim is to reach a primal-dual formulation so as to apply a greedy p-phase algorithm closely resembling algorithms already presented for the UFL problem. Let $\mathcal{R} = \{1, 2, 3, ..., R\}$ and set dummy clients for the unused connectivity requirements. Following the UFL transformation described above, each client j has r_j ports, each site R facilities, and all ports must be connected. Let $\mathcal{C}^p = \{j \in \mathcal{C} : r_j \geq p\}$ be the set of clients that get their requirement satisfied by one unit in phase p. Variable $y_i^p \in \{0, 1\}$ denotes whether the p-th facility at site i is opened and variable $x_{ij}^p \in \{0, 1\}$ denotes whether the p-th port of a city is connected with a facility at site i. This leads to the following formulation of the FTFA:

$$minimize \sum_{i \in \mathcal{F}} \sum_{p \in \mathcal{R}} (f_i y_i^p + \sum_{j \in \mathcal{C}} c_{ij} x_{ij}^p)$$

$$\text{subject to:} \quad \sum_{i \in \mathcal{F}} x_{ij}^p \geq 1 \qquad \forall p \in \mathcal{R}, j \in \mathcal{C} \tag{3.15}$$

$$\sum_{p \in \mathcal{R}} y_i^p - \sum_{p \in \mathcal{R}} x_{ij}^p \geq 0 \qquad \forall p \in \mathcal{R}, i \in \mathcal{F},\ j \in \mathcal{C} \tag{3.16}$$

$$x_{ij}^p,\ y_i^p \in \{0, 1\} \qquad \forall p \in \mathcal{R}, i \in \mathcal{F},\ j \in \mathcal{C} \tag{3.17}$$

Authors take special care in the second constraint, as it is not equivalent to the UFL case where $y_i^p \geq x_{ij}^p,\ \forall p \in \mathcal{R}$. That is, there can be a case $y_i^p < x_{ij}^p$ for some p, meaning that the p-th port of client x can be connected to a facility opened on site i in an earlier phase, as $y_i = 0$ and $x_{ij}^p = 1$. Thus, they define cost $f_i^p = f_i$ if an extra facility needs to be opened on site i in phase p, and $f_i^p = 0$ otherwise. Consequently, authors define a variable $z_i^p \in \{0, 1\}$ to denote whether a new facility is opened on site i in phase p, i.e. $z_i^p = 1$ if $\sum_{j \in \mathcal{C}^p} x_{ij}^p > 0$ and $z_i^p = 0$ otherwise, and observe that since $\sum_{p \in \mathcal{R}} z_i^p$ is not necessarily equal to y_i, it can be defined as $y_i = \max_{j \in \mathcal{C}} \sum_{p \in \mathcal{R}} x_{ij}^p$. This implies that the previous formulation can be rewritten as:

$$minimize \sum_{p \in \mathcal{R}} \sum_{i \in \mathcal{F}} (f_i^p z_i^p + \sum_{j \in \mathcal{C}^p} c_{ij} x_{ij}^p)$$

$$\text{subject to:} \quad \sum_{i \in \mathcal{F}} x_{ij}^p \geq 1 \qquad \forall p \in \mathcal{R}, j \in \mathcal{C}^p \tag{3.18}$$

$$\sum_{p \in \mathcal{R}} z_i^p - x_{ij}^p \geq 0 \qquad \forall p \in \mathcal{R}, i \in \mathcal{F},\ j \in \mathcal{C}^p \tag{3.19}$$

$$x_{ij}^p,\ z_i^p \in \{0, 1\} \qquad \forall p \in \mathcal{R}, i \in \mathcal{F},\ j \in \mathcal{C}^p \tag{3.20}$$

Accordingly, this is the dual of the above problem's LP relaxation:

$$maximize \sum_{p\in\mathcal{R}}\sum_{j\in\mathcal{C}^p}\alpha_j^p$$

$$\text{subject to:} \quad \sum_{j\in\mathcal{C}^p}\beta_{ij}^p \leq f_i \qquad \forall p\in\mathcal{R}, i\in\mathcal{F} \tag{3.21}$$

$$\sum_{p\in\mathcal{R}}\alpha_j^p - \beta_{ij}^p \leq c_{ij} \qquad \forall p\in\mathcal{R}, i\in\mathcal{F},\ j\in\mathcal{C}^p \tag{3.22}$$

$$\alpha_j^p, \beta_{ij}^p \geq 0 \qquad \forall p\in\mathcal{R}, i\in\mathcal{F},\ j\in\mathcal{C}^p \tag{3.23}$$

According to weak duality theorem, it is enough to find an effective algorithm producing a feasible solution to the dual problem. The authors decompose the problem into R sub-phases and in each sub-phase p they consider the group of city *ports* p, where $\mathcal{C}^p = \{j \in \mathcal{C}, r_j \geq p\}$ with vectors $\mathbf{X}^b = \sum_{p=1}^b \mathbf{x}^p$ and $\mathbf{Y}^b = \sum_{p=1}^b \mathbf{y}^p$, $1 \leq b \leq R$, establishing one connection for each city j in $\mathcal{C}^p$. As in [32], the *cost-efficiency* of a star is defined as $\text{eff}(i, p, C') = \frac{f_i^p + \sum_{j\in C'} c_{i,j}}{|C'|}$, where f_i^p is the cost to open a facility at site i in phase p and C' the set of clients participating in the star. Dual variables α_j^p and β_{ij}^p can be used to find the most cost effective star; if the dual variables of all unconnected clients get raised simultaneously, the most cost effective star is the first (i, p, C') such that $\sum_{j\in C'} \max(t - c_{ij}, 0) = f_i^p$, where $\alpha_j^p = t$ and $\beta_{ij}^p = \max(t - c_{ij}, 0)$. A description of the p-phase Algorithm, closely reminiscent of the first version of the JMS et al. [32] algorithm:

Algorithm for the p-phase

```
1. Let $U \subseteq \mathcal{C}^p$ be the set of not fully connected cities, initially set $U \leftarrow \mathcal{C}^p$.
2. While $U \neq \emptyset$:
    (a) Find the most cost efficient star $(i, p, C')$.
    (b) Open a facility at site $i$ if not already open, and connect to it all cities in $C'$.
    (c) Set $f_i \leftarrow 0$, $U \leftarrow U \backslash C'$.
```

This subroutine differs from the one used for the UFL in [32] that in each iteration the feasibility of the solution needs to be maintained by assuring that $X_{ij}^p \leq Y_i^p$,

$\forall i \in \mathcal{F}, p \in \mathcal{R}, j \in \mathcal{C}^p$. Without loss of generality, the algorithm can set $y_i^p \leftarrow 1$ when a new facility on site i is opened and $x_{ij}^p \leftarrow 1$ when a new connection between client j and facility i is established. So, in order to maintain condition $X_{ij}^p \leq Y_i^p$, three cases are considered in the FTFA for any $j \in \mathcal{C}^p$:

1. $X_{ij}^{p-1} \leq Y_i^{p-1}$: The feasibility of the solution is maintained if $x_{ij}^p \leftarrow 1$. Then there is no need to open a new facility on site i and, thus, $f_i^p = 0$.

2. $X_{ij}^{p-1} = Y_i^{p-1}$ and $y_i^p = 0$: A new facility need to be opened at site i in order to connect client j, which means $y_i^p \leftarrow 1$ and $f_i^p = p$. The opening cost is shared between a set of clients in C' that need to connect to that facility.

3. $X_{ij}^{p-1} = Y_i^{p-1}$ and $y_i^p = 1$: A new facility was opened at that site i in this phase p by some previous clients in C' before j, meaning that $x_{ij}^p \leftarrow 1$ and $f_i^p = 0$.

Consequently, the p-phase subroutine can be restated as follows:

Algorithm for the p-phase restated

1. Let $U \subseteq \mathcal{C}^p$ be the set of not fully connected cities, initially set $U \leftarrow \mathcal{C}^p$ and $t \leftarrow 0$. Assuming $\forall j \in U$ has r_j ports with a credit associated with the connection cost and which increases from zero simultaneously with time before the port is connected, set $a_j^p = 0$, $\forall \in U$

2. While $U \neq \emptyset$, increase time t until one of the following event occur:

   (a) A city $j \in U$ has enough credit to be connected with an already opened facility on site i, i.e. $t = c_{ij}$ and $X_{ij}^{p-1} \leq Y_i^{p-1}$. In this case, $X_{ij}^p \leftarrow X_{ij}^{p-1}$

   (b) A site i receives enough credit from cities in U to open it's p-th facility, i.e. $\sum_{j \in U} \max(t - c_{ij}, 0) = f_i$. In this case, $C' = \{j \in U : \ c_{ij} \leq t\}$, $Y_i^p \leftarrow Y_i^{p-1} + 1$ and $X_{ij}^p \leftarrow X_{ij}^{p-1} + 1$ for any $j \in C'$.

   (c) A city j in U has enough credit to be connected to a newly opened (in this phase) facility. i.e. $t = c_{ij}$. In this case, $X_{ij}^p \leftarrow X_{ij}^{p-1} + 1$.

3. For any city $j \in U$, set $\alpha_j^p \leftarrow t$ and remove city j from U if it got connected with a facility in phase p.

The complete 1.861-approximation algorithm for the FTFA is the following:

```
Algorithm for the FTFA
  1. Initially $X^0 \leftarrow 0$, $Y^0 \leftarrow 0$, $\mathcal{C}^1 = \mathcal{C}$ and $p \leftarrow 1$.
  2. While $p \leq R$:
      (a) Invoke the $p$-th phase Algorithm with input $(X^{p-1}, Y^{p-1}, \mathcal{F}, \mathcal{C}^p)$ and receive output $(X^p, Y^p)$.
      (b) Set $p \leftarrow p - 1$.
  3. Set $x = X^R$ and $y = Y^R$.
```

For $\mathcal{I}$ and instance of the FTFA and $p \in R$ a phase, let

$$\lambda_{p,\mathcal{I}} = \max_{i \in \mathcal{F}, p \in R, C' \subseteq \mathcal{C}^p} \frac{\sum_{j \in C'} \alpha_j^p}{f_i + \sum_{j \in C'} c_{ij}}$$

be the maximum cost ratio with respect to any star (i, p, C'). The authors claim that the cost of the solution in each phase p is equal to $\sum_{j \in \mathcal{C}^p} \alpha_j^p$ and that the maximum cost ratio $\lambda_{p,\mathcal{I}}$ is bounded by a constant λ for any phase p and any instance $\mathcal{I}$ of the problem. Based on that, they prove the following theorem:

Theorem. *If the p-th phase Algorithm fulfills the claim, then the complete Algorithm for the FTFA is a λ-approximation to the FTFA.*

The proof to the theorem is based on an inverse dual fitting technique, compared to the one in [32], by composing an extra instance of the problem with same size as the original, but different value for facility and connection cost, scaling *up* both costs by factor λ (that is, $f'_i \leftarrow \lambda f_i$ and $c'_{ij} \leftarrow \lambda c_{ij}$). Arguing that the inverse dual fitting technique is more powerful in multifactor approximation analysis based on results by the same authors, and because in the FTFA a client is involved in multiple stars, thus assigning their costs to achieve balance between relevant stars, as would be expected in the traditional dual fitting analysis, is rather complicated, the authors instead of shrinking the dual variables prefer to use the unshrunk duals which are feasible to the composed instance of the dual problem and achieve the λ approximation ratio based on the claim, which in turn is proved using the following lemmas:

Lemma 1. *For any instance $\mathcal{I}$ and phase $p \in R$, $\sum_{j=h}^{k} \max(\alpha_h^p - c_{ij}, 0) \leq f_i$ holds for any facility i and any client h, $1 \leq h \leq k$.*

The lemma above covers the case where a new facility is opened and on how the contribution is received. The following lemma guarantees a property concerning the triangle inequality. Because only ports of the same rank are processed in each phase, it is possible to derive the following property despite the fault-tolerant nature of the problem:

Lemma 2. *For any instance $\mathcal{I}$ and phase $p \in R$, $\alpha_j^p \leq \alpha_h^p + c_{ij} + c_{ih}$ holds for any facility i and clients h and j, $1 \leq h, j \leq k$.*

By evaluating λ_k, where λ_k the upper bound of the following factor-revealing LP as in [32] the authors prove that their algorithm is an 1.861-approximation for the FTFA.

Chapter 4

Leasing Problems

4.1 The Parking Permit Problem

So far, we have examined models where the service nodes, once opened, can be indefinitely used for no extra charge. The first problem in network design dealing with time durations, is considered to be the Parking Permit problem. It is first introduced by Meyerson in [48] and can be seen as a variant of the ski rental problem [12], a classic problem in online algorithms. The Parking Permit is an online problem too, where purchases have durations of various costs which expire regardless of their use, and is thus labeled because of the real-world example presented by Meyerson: A commuter has the choice to go to work either on foot or by driving. However, it is not known in advance what he will opt for. On any driving day, he can apply for a parking permit, picking from a set of parking durations and permits following the subadditive property.

More formally, there are K different types of permits available to purchase, each permit k with duration D_k days and cost C_k. The schedule indicating which are the driving days is revealed one day at a time. The goal is to minimize the competitive ratio $\alpha(K)$ of the cost paid in the online version to cover all days versus the offline cost. Meyerson two observations on the nature of the problem, which would be later applied by Nagarajan & Williamson in [50] to a more general infrastructure leasing context:

Scaling Theorem. *For each permit type $1 < k \leq K$, it can be assumed that $C_k \geq 2C_{k-1}$ and $D_k \geq D_{k-1}$ by a loss of at most a factor of 2 in the competitive ratio*

Interval Model Theorem. *Assume a version of the problem in which each permit* *is available only over specific time spans and each permit of type k has $R_k = \frac{D_k}{D_{k-1}}$ permits of type $k-1$ embedded within. At any given time, there is exactly one possible permit of type k to cover this time. Within a $\Theta(1)$ factor, any online algorithm for this version of the problem is competitive for the original version, and vise-versa.*

A deterministic approach for the ski rental problem, where the skis are rented until the total rental payment equals the purchase cost and upon which point the skis are purchased, yields a $\Theta(1)$ competitive ratio. An adaptation for the parking permit problem, is the following: Consider the interval version of the problem. Each time the commuter chooses to drive, a permit of type 1 is purchased, until a type 2 permit becomes available and the optimum solution would purchase a type 2 permit, assuming we have seen so far the entire schedule. For any interval of type k, as soon as the optimum offline solution would purchase this permit (only based on the schedule seen do far), the proposed algorithm purchases it too. Meyerson proves that the stated deterministic algorithm delivers an $O(K)$-competitive ratio for the parking permit problem and, furthermore, that no deterministic algorithm whose competitive ratio is dependent solely on the number K of permits can deliver better than $\Omega(K)$ competitive ratio.

However, Meyerson obeserved that there is an equivalence between an online deterministic $fractional$ solution and a randomized algorithm, thus designing a randomized algorithm with memory based on the following theorem:

Theorem. *There exists a randomized algorithm for the parking permit proble with competitive ratio $\Theta(\alpha(K))$ iff there exists a deterministic fractional algorithm with competitive ratio $\Theta(\alpha(K))$.*

Consequently, a fractional algorithm can buy fractional permits, maintaining that each driving day, the total sum of the fractional permits is at least equal to 1. In the next page there is a description of the online fractional algorithm:

This online algorithm algorithm can be transformed to a randomized integral algorithm using the method by which the previous theorem was proven. In the online algorithm, it can be shown that each operation increases the fractional cost by at most 2. Therefore, bounding the total number of operations also bounds the total cost. The following theorems:

Theorem. *The online fractional algorithm is $O(\log K)$ - competitive.*

and

Algorithm

1. Initially set all permits to 0.

2. If it is a driving day and the total sum of the fractional permits purchased for this days is less than 1, repeat the following steps until the sum exceeds one:

   (a) For each $1 \leq i \leq K$ multiply by a factor of $1 + \frac{1}{C_i}$ the fraction by which the currently valid permit of type i is purchased.

   (b) For each $1 \leq i \leq K$ increase the fraction by which the currently valid permit of type i is purchased by adding $\frac{1}{KC_i}$.

Theorem. *Any randomized algorithm for the Parking Permit Proble has expected competitive ratio at least* $\Omega(\log K)$.

constitute the parking permit problem closed with respect to designing approximation algorithms with ratios dependent on the number of permits K. Meyerson also gives results on delivering algorithms with competitive ratio dependent on durations or costs:

Theorem. *There is a deterministic algorithm for the parking permit problem obtaining competitive ratio* $O(\log \frac{C_k}{C_1})$ *and a randomized algorithm with expected competitive ratio* $O(\log \log \frac{C_k}{C_1})$*.*

Theorem. *There is a deterministic algorithm for the parking permit problem obtaining competitive ratio* $O(\log \frac{D_k}{D_1})$ *and a randomized algorithm with expected competitive ratio* $O(\log \log \frac{D_k}{D_1})$*.*

It is worth noting that in the randomized lower bound, the ratio of durations is quite large, allowing to choose $D_k = 2^{k^2}$ and still return good results, the lower bound still matching the upper bound.

Theorem. *No deterministic algorithm for the parking permit problem can guarantee a competitive ratio better than* $\Omega(\log \frac{C_k}{C_1})$ *and no randomized algorithm can guarantee an expected competitive ratio better than* $Omega(\log \log \frac{C_k}{C_1})$*.*

Theorem. *No deterministic algorithm for the parking permit problem can guarantee a competitive ratio better than* $\Omega(\log \frac{D_k}{D_1} / \log \log \frac{D_k}{D_1})$ *and no randomized algorithm can guarantee an expected competitive ratio better than* $\Omega(\log \frac{D_k}{D_1} / \log \log \frac{D_k}{D_1})$*.*

4.2 Infrastructure Leasing Problems

In [8] Anthony & Gupta presented a novel variant of facility location introducing the notion of *time* called Facility Leasing as part of a general framework for Infrastructure Leasing problems. The problem can be described as follows: A set of clients D is given and a set F of facility locations. There are distinct time periods from time 1 to T and at each time period t, a subset D_t of clients that must be served by a facility that is open *at that time*. There are K different $l_1, l_2, ..., l_K$ lease lengths available and each facility $iinF$ can be leased at any period t for lease length l_k at a cost f_i^k. The lease cost can be dependent on both the facility and the lease type. Standard metric connection costs c apply, satisfying the triangle inequality and dependent on the distance between a client j and the allocated facility i. The goal is to minimize the total connection and leasing costs while ensuring that for each time period t, clients in D_t can be served by at least one open, at that period, facility.

Anthony & Gupta pointed out the connection between deterministic leasing problems and stochastic optimization problems and, consequently, how algorithms and techniques designed for the latter can also be employed in the former, leading to the following *General Leasing Theorem*:

Theorem 1. *The offline leasing version of a subadditive combinatorial optimization problem* Π *with* $|K| = k$ *lease lengths can be reduced to the stochastic optimization version of* Π *in the model of k-stage stochastic optimization with recourse.*

where the k-stage stochastic optimization with recourse can be defined as follows: The demand set D is revealed on day-k drawn from some known distribution π, but on each of days $1, 2, ..., k1$ we are given additional information about the set D. This process can alternatively be viewed as having a joint distribution over signals $s_1, s_2, ..., s_{k1}$, each s_t received on the various days $t \in \{1, 2, ..., k-1\}$, with actual demand set some known function of this signals. The costs of elements change over time, usually getting more expensive.

The following lemmas lead them to their next central theorem:

Lemma 4.2.1. *Given any instance I of a leasing problem, I can be converted into an instance I' in which the lengths of leases exactly divide each other (i.e., $\ell_i | \ell_j$ for $i < j$), and where the costs satisfy $c(\ell_j) < c(\ell_i) \times (\ell_j/\ell_i)$. Moreover, there is an optimal solution to I' which has cost at most 2 times the optimal cost for I.*

Lemma 4.2.2. *Given an instance I of a leasing problem, there is a solution which has cost at most 2 times the optimum, where a lease of length ℓ is obtained only for intervals of the form $[t, t+\ell)$ with t a multiple of ℓ.*

Graphically this can be represented as:

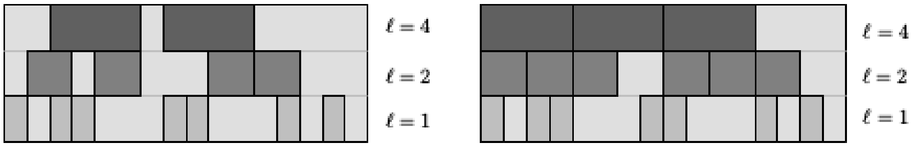

Figure 4.1: image taken from [8]

where on the left lies the solution and on the right the corresponding nested version.

Assuming without loss of generality that lease length $l_1 = 1$ and that the solutions are nested:

Theorem 2. *Any offline problem Π in the above framework with $|K| = k$ lease lengths can be reduced to the standard k-stage stochastic optimization version of Π.*

Consequently, [8] guaranteed for the Facility Leasing Problem an $O(k)$-approximation algorithm.

4.3 The Facility Leasing Problem

The following year, in [50] Nagarajan & Williamson attempted a totally different approach and went over the $O(\log n)$ barrier, thus opening new potentials for the facility leasing problem. By formulating the problem in its LP relaxation, they were able to treat the problem as a generalized UFL and by altering the classical Jain-Vazirani algorithm from [35] they produced a 3-approximation algorithm for the leasing version. In order to express the problem in an LP-relaxation, Nagarajan & Williamson define the following: Let $\mathcal{L}$ be the set of the available lease lengths, where $\mathcal{L} = K$. They extend the definition of a facility to a triple (i, k, t), where i is the facility location starting a lease of duration l_k at time t, meaning that i can serve clients arriving in the time interval $[t, t + l_k)$. Let I_t^k denote that time interval. In the same spirit, the definition of a client is extended to a pair (j, t) where $j \in D_t$, D_t the set of client demanding service on time period t. Denoting the set of facility triples as $\mathcal{F}$ and the set of client demand pairs as $\mathcal{D}$, they give the following LP-relaxation:

$$minimize \sum_{(j,t)\in\mathcal{D}} \sum (i,k,t') \in \mathcal{F} : t \in I^k_{t'} c_{ij} x_{ikt',jt} + \sum_{(i,k,t)\in\mathcal{F}} f^k_i y_{ikt}$$

$$\begin{aligned}
\text{subject to:} \quad & \sum_{(i,k,t')\in\mathcal{F}:t\in I^k_{t'}} x_{ikt',jt} \geq 1 & \forall (j,t) \in \mathcal{D} \qquad (4.1)\\
& y_{ikt'} - x_{ikt',jt} \geq 0 & \forall (i,k,t') \in \mathcal{F},\ (j,t) \in \mathcal{D} \qquad (4.2)\\
& x_{ik't,jt}, y_{ikt} \geq 0 & \forall (j,t) \in \mathcal{D}, (i,k,t), (i,k,t') \in \mathcal{F} \qquad (4.3)
\end{aligned}$$

The dual program is:

$$maximize \sum_{(j,t)\in\mathcal{D}} v_{jt}$$

$$\begin{aligned}
\text{subject to:} \quad & v_{jt} - w_{ikt',jt} \leq c_{ij} & \forall (j,t) \in \mathcal{D}, (i,k,t') \in \mathcal{F}, t \in I^k_{t'} \qquad (4.4)\\
& \sum_{(j,t)\in\mathcal{D}} w_{ikt',jt} \leq f^k_i & \forall (i,k,t') \in \mathcal{F} \qquad (4.5)\\
& v_{jt}, w_{ikt',jt} \geq 0 & \forall (j,t) \in \mathcal{D},\ (i,k,t') \in \mathcal{F} \qquad (4.6)
\end{aligned}$$

where $x_{ikt',jt}$ denotes whether client (j,t) is assigned to facility (i,k,t') and $y_{ikt'}$ indicates whether facility i is chosen to be opened at time t' for lease length of type k.

The algorithm proposed by Nagarajan & Williamson is identical to the Jain-Vazirani primal-dual algorithm for the UFLP, following the same sequence of two *Phases* and *events*. In the leasing variation, as client is considered now the pair (j,t) and as facility set the set of facility triples $\mathcal{F}$, while maintaining for the dual variables that $w_{ikt',jt} = \max(0, v_{jt} - c_{ij})$. Thus, if constraint (3.18) becomes *tight* for a facility $(i,k,t) \in \mathcal{F}$, then the triple (i,k,t) is declared temporarily open and define as $\mathcal{T}$ the set of temporarily open facilities. Similarly, a client (j,t) *contributes* to facility (i,k,t') if $t \in I^k_{t'}$ and $v_{jt} > c_{ij}$ and is considered *connected* to that facility if $t \in I^k_{t'}$ and $v_{jt} \geq c_{ij}$. As in the original Jain-Vazirani algorithm, Phase 1 proceeds increasing uniformly the duals of clients not connected to a temporarily open facility and ends when there no more unconnected clients.

Following the logic of the JV-algorithm, in Phase 1 a client may have contributed towards the opening/*leasing* of more than one facilities/*leases*. In order to ensure that each client contributes to only one facility lease, Nagarajan & Williamson construct in Phase 2 a graph $G(V, E)$ with $V = \mathcal{T}$ and an edge between to facilities in V if there is client that contributes to both facilities. Although in [35] a random maximal independent set in G is found, in the leasing version of the algortithm Nagarajan & Williamson first order the temporarily open facilities in $V = \mathcal{T}$ according to non-increasing lease lengths and then, following this ordering, greedily pick the vertices/facility triples to construct the maximal independent set $\mathcal{I}$. The aim is to give priority to facilities with *longer* lease lengths and thus be able to bound the returned solution. The constructed independent set I is maximal and, moreover, for every temporarily opened $(i, k, t') \in \mathcal{T} \backslash \mathcal{I}$, there exists a facility $(i, k, t) \in \mathcal{I}$ adjacent to it in G and with the same or longer lease length.

This final property is crucial in constructing a feasible solution. For each $(i, k, t) \in \mathcal{I}$, Nagarajan & Williamson include three leases in the final solution: the initial (i, k, t), $(i, k, t + l_k)$ and $(i, k, (t - l_k)_+)$, where $(t - l_k)_+) = max(0, t - l_k)$. Denote by $\mathcal{I}'$ the final set of the above described leases, open those facilities in $\mathcal{I}'$ accordingly and connect each client to the closest open facility. The figure below gives an overview of why this is a feasible solution:

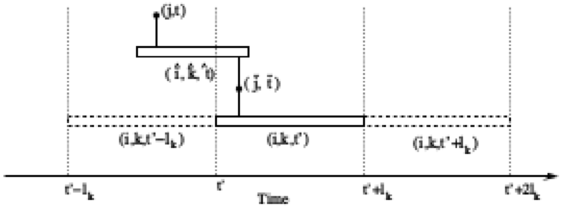

Figure 4.2: image taken from [50]

Consider a client (j, t). If it is connected to a facility $(i, k, t') \in \mathcal{I}$, assign the client to that facility and consider (j, t) *directly* connected to (i, k, t'). If not, then (j, t) is connected to a temporarily opened facility (î,k̂,t̂)$\in \mathcal{T} \backslash \mathcal{I}$, and given that $\mathcal{I}$ is a maximal independent set, this temporarily opened facility must be adjacent to some $(i, k, t')\ in \mathcal{I}$. Client (j, t) can be *indirectly* connected to one of the three opened facilities (i.e. (i, k, t), $(i, k, t + l_k)$ and $(i, k, (t - l_k)_+)$). This is possible, because $t \in [(t' - l_k)_+, t' + 2l_k)$.

Proof: Since (î,$\hat{\text{k}}$,t̂) and (i,k,t') are adjacent, there is some client $(\bar{j},\bar{t})$ contributing to both facilities. Therefore, $\bar{t} \in I_{\hat{t}}^{\hat{k}} \cap I_{t'}^{k}$. Also, lease length $l_{\hat{k}}$ cannot be longer than lease length l_k, due to the ordering of the facilities according to their lease lengths prior to the construction of $\mathcal{I}$. Thus, $I_{\hat{t}}^{\hat{k}} \subseteq [(t'-l_k)_+, t'+l_k)$ and client (j,t) can indeed be serviced by one of the facilities (i,k,t), $(i,k,t+l_k)$ or $(i,k,(t-l_k)_+)$. $\square$

Apart from this remark, Nagarajan & Williamson follow closely the analysis in [35], which is already presented in Chapter 1 under the Primal-Dual schemas, to conclude that their algorithm is a 3-approximation for the Leasing Facility Location.

Chapter 5

Efforts

The introduction of the notion of time in the Facility Location problem opened up a field of new possibilities in this research area. Motivated by this perspective, we initially tried to tackle the Facility Leasing in a fault-tolerant setting. That is, we maintained the natural description of the problem as it was stated in [50], but instead of unit demand, we allowed clients to have multiple requirements per day, in the sense that client j on day t had a demand to be connected to r_{jt} different facility *sites* i serving on that day t.

The problem was formulated as follows:

$$\text{minimize} \sum_{(j,t)\in\mathcal{D}} \sum_{(i,k,t')\in\mathcal{F}:t\in\mathrm{I}_{t'}^k} c_{ij} x_{ikt',jt} + \sum_{(i,k,t)\in\mathcal{F}} f_i^k y_{ikt}$$

$$\begin{aligned}
\text{subject to:} \quad y_{ikt'} - x_{ikt',jt} &\geq 0 && \forall (i,k,t') \in \mathcal{F},\ (j,t) \in \mathcal{D} && (5.1)\\
\sum_{(i,k,t')\in\mathcal{F}:t\in\mathrm{I}_{t'}^k} x_{ikt',jt} &\geq r_{jt} && \forall (j,t) \in \mathcal{D} && (5.2)\\
1 &\geq y_{ikt} && \forall (i,k,t) \in \mathcal{F} && (5.3)\\
1 &\geq \sum_{(i,k,t')\in\mathcal{F}} x_{ikt',jt} && \forall (j,t) \in \mathcal{D}, i \in \mathcal{F}_{loc} && (5.4)\\
x_{ikt',jt}, y_{ikt} &\in \{0,1\} && \forall (i,k,t'), (i,k,t) \in \mathcal{F},\ (j,t) \in \mathcal{D} && (5.5)
\end{aligned}$$

leading to this relaxation:

$$\text{minimize} \sum_{(j,t)\in\mathcal{D}} \sum_{(i,k,t')\in\mathcal{F}:t\in \mathrm{I}^k_{t'}} c_{ij}x_{ikt',jt} + \sum_{(i,k,t)\in\mathcal{F}} f^k_i y_{ikt}$$

$$\text{subject to:} \quad y_{ikt'} - x_{ikt',jt} \geq 0 \qquad \forall (i,k,t') \in \mathcal{F},\ (j,t) \in \mathcal{D} \quad (5.6)$$

$$\sum_{(i,k,t')\in\mathcal{F}:t\in \mathrm{I}^k_{t'}} x_{ikt',jt} \geq r_{jt} \qquad \forall (j,t) \in \mathcal{D} \quad (5.7)$$

$$1 \geq y_{ikt} \qquad \forall (i,k,t) \in \mathcal{F} \quad (5.8)$$

$$1 \geq \sum_{(i,k,t')\in\mathcal{F}} x_{ikt',jt} \qquad \forall (j,t) \in \mathcal{D}, i \in \mathcal{F}_{loc} \quad (5.9)$$

$$x_{ikt',jt}, y_{ikt} \geq 0 \qquad \forall (i,k,t'), (i,k,t) \in \mathcal{F},\ (j,t) \in \mathcal{D} \quad (5.10)$$

and its dual:

$$\text{maximize} \sum_{(j,t)\in\mathcal{D}} \alpha_{jt} r_{jt} \ - \sum_{(i,k,t')\in\mathcal{F}} z_{ikt'} - \sum_{(j,t)\in\mathcal{D}} \sum_{i\in\mathcal{F}_{loc}} w_{i,jt}$$

$$\text{subject to:} \quad \sum_{(j,t)\in\mathcal{D}} \beta_{ikt',jt} \leq f^k_i + z_{ikt'} \qquad \forall (i,k,t') \in \mathcal{F} \quad (5.11)$$

$$\alpha_{jt} - \beta_{ikt',jt} - w_{i,jt} \leq c_{ij} \qquad \forall (j,t) \in \mathcal{D},\ i \in \mathcal{F}_{loc},\ (i,k,t') \in \mathcal{F},\ t \in \mathrm{I}^k_{t'} \quad (5.12)$$

$$\alpha_{jt}, \beta_{ikt',jt} \geq 0 \qquad \forall (j,t) \in \mathcal{D},\ (i,k,t') \in \mathcal{F} \quad (5.13)$$

As one can see, we have added the condition $1 \geq \sum_{(i,k,t')\in\mathcal{F}} x_{ikt',jt} \forall (j,t) \in \mathcal{D}, i \in \mathcal{F}_{loc}$, where $\mathcal{F}_{loc}$ is considered to be the set of facility *sites*, that is the set from which the triplet (i,k,t) draws its first variable. This addition was done to ensure that each unit requirement of client j on a day t would be satisfied by one and only facility site i (and in the formulation of the IP, by one specific (i,k,t')) and consequently, that its requirement would be satisfied by r_{jt} facilities on different site. In other words, we avoided the case where a client would be mistakenly considered connected two to "different" facilities (i,k,t') and (i,k',t''), i.e. connected to the *same* facility site, but to different "triplets", that is to different combinations of lease lengths and lease starting days by the same facility i.

Variable $w_{i,jt}$ in the dual LP is corresponds to the addition of this condition.

As a means to better understand the properties and underlying structure of the problem, we opted to approach the problem from a more traditionally combinatorial, initially using the well studied primal-dual method, and later on the combinatorial clustering by Swamy and Shmoys as presented in [56]. Along the road, we encountered the following difficulties, all of which had to do with time-related issues.

To begin with, it soon became apparent that we could not find a way to use a p-phase primal dual JV like algorithm in the fault-tolerant setting of the facility leasing. A p-phase algorithm would include a subroutine which accepts a $(\mathcal{D}_{p+1}, I_{p+1})$ solution from a previous iteration of the subroutine, where according to usual notation D_{p+1} are clients with requirement $p+1$ and I_{p+1} the set of "opened"/leased facility triplets, and return a $(\mathcal{D}_p, I_p)$ having satisfied the group of clients $(\mathcal{D}_{p+1})$ with maximum residual requirement p by one unit, thus returning a solution $(\mathcal{D}_p, I_p)$ with now current maximum requirement p according to the methodology used by Jain & Vazirani and Nagarajan & Williamson. Now, imagine client j with residual requirement p on a day t getting connected according to the subroutine in phase p to a facility i with a lease length of k opening on day t' - that is, connecting (j,t) to (i,k,t'). In the following iteration (phase $p-1$), the connections already used in $(\mathcal{D}_p, I_p)$ (the solution return from the previous phase) are of infinite cost, so that we do not reuse them... But what about the facilities? Normally, we would have $I_p = \mathcal{F} \backslash I_{p+1}$, that is, we should exclude I_p from $\mathcal{F}$ so as to not "release" the same triplets (i,k,t') that are already in I_p. However, this is not enough, because there is still the case that (j,t) gets connected during phase $p-1$ to the same facility i, just with different lease length and/or starting lease time, for example (j,t) which got connected to (i,k,t') in phase p, in phase $p-1$ could get connected to (i,k',t'') as long as $t \in \mathcal{I}_{t''}^{k'}$.

However, this is not what we wish for, since fault-tolerance means that client (j,t) should get connected to a different facility *site*, not to facility i again. We could not find an effective way to handle this, because we could not find a strategy that would exclude facility triplets containing facility i and in the same time, provide a guarantee for the bound of the returned resolution solution. Even by applying the Scaling and Interval Model theorems mentioned in section 4.2, we could not reach to a bound or a ratio for the solution returned by the p-th iteration, whereas in [34] and [50], for example, the analysis of the approximation was based on the observation that the optimum of p-phase LP is at most OPT_f/p, where OPT_f the optimal solution of the original problem's LP.

Therefore, we tried to change perspective and modify the 4-approximation combinatorial algorithm presented in [56] by Swamy and Shmoys, described in section 2.3 . As in the original algorithm, our version would accept as input the dual optimal solution $(\alpha, \beta, \zeta, w)$ and run in two phases, just like the algorithm in [56], maintaining the same notation. Phase 1 run exactly as in [56], i.e. each facility triplet (i, k, t) with $y_{ikt} = 1$ would be considered opened and included in L_1. Accordingly, the cost of Phase 1 is proven to be $\sum_{(j,t)\in\mathcal{D}} n_{jt}\alpha_{jt} - \sum_{(i,k,t')\in L_1} z_{ik't} - \sum_{i\in\mathcal{F}_{loc}:(i,k,t')\in L_1} \sum_{(j,t)\in\mathcal{D}} w_{i,jt}$.

Still following the notation in [56], plus $maxL$ as the maximum lease length available, we changed Phase 2 as follows:

Step 1. Pick $(j,t) \in \mathcal{S}$ with minimum α_{jt} as cluster center.
Step 2. Order the facilities in F_{jt} by increasing leasing cost. Starting from the first facility in F_{jt}, pick $M \subseteq F_{jt}$ such that $\sum_{(i,k,t')\in M} y_{ikt'} \geq r'_{jt}$. If $\sum_{(i,k,t')\in M} y_{ik't} > r'_{jt}$, split the last (i,k,t') in M, in two copies (i_1,k,t') and (i_2,k,t'). Set $y_{i_1kt'} = r'_j - \sum_{i',k,t'\in M\setminus\{i,k,t'\}} y_{i_1kt'}$ and $y_{i_2kt'} = y_{ikt'} - y_{i_1kt'}$. For every client (j^*,t^*), including (j,t), with $x_{ikt',j^*t^*} > 0$ set arbitrarily x_{i_1kt',j^*t^*}, x_{i_2kt',j^*t^*} such that $x_{i_1kt',j^*t^*} + x_{i_2kt',j^*t^*} = x_{ikt',j^*t^*}$, $x_{i_1kt',j^*t^*} \leq y_{i_1kt'}$, $x_{i_2kt',j^*t^*} \leq y_{i_2kt'}$. Include only i_1kt' in M, so that now $\sum_{(ikt)\in M} y_{ikt} = r_(jt)'$.
Step 3. Open the r'_{jt} least expensive facilities in M. For each (i,k,t') of the opened facilities, lease also the same facility i earlier for $maxL$ duration, with the lease ending exactly on time t' and on release it on $t'+k$ for $maxL$ duration. For each client (j^*,t^*), including (j,t) with $F_\cap M \neq \emptyset$, connect $\min(r'_{j^*t^*}, r'_{jt})$ copies of (j^*,t^*) to these newly opened facilities and set $r'_{j^*,t^*} := r'_{j^*,t^*} - \min(r'_{j^*t^*}, r_{jt})$ and $F_{j^*t^*} := F_{j^*t^*}\setminus\{(i,k,t) \in \mathcal{F} : y_{ik't'} > 0$ and $x_{j^*t^*,ik't'} > 0$, (i,k',t') in the r'_{jt} least expensive facilities in $M\}$. Remove (j,t) and facilities in M from the process.

The cost of opening the least expensive facilities in Step 3 of the second phase is at most $3 \cdot maxL \cdot \sum_{(i,k,t)\in\mathcal{F}} f_i^k y_{ikt}$, based on the assumption that the leases are subadditive and following the proof in [50]. Similarly, we also have $c_{ikt',j^*t^*} \leq 3\alpha_{j^*t^*}$ for a copy of (j^*,t^*) connected to (i,k,t') in Phase 2. However, we could not use the above inequalities to derive to a bound on the total cost of the algorithm, due to the presence of the negative term $-\sum_{(j,t)\in\mathcal{D},\, i\in\mathcal{F}_{loc}} w_{i,jt}$. More specifically, we could not associate the delivered cost of Phase 1 with the cost of Phase 2:

The facility cost in phase 2 is at most $3 \cdot maxL \cdot \sum_{(i,k,t)\in\mathcal{F}} f_i^k y_{ikt} \leq 3 \cdot maxL \cdot OPT = 3 \cdot maxL \cdot \sum_{(j,t)\in\mathcal{D}} r'_{jt}\alpha_{jt} + (\sum_{(j,t)\in\mathcal{D}} n_{jt}\alpha_{jt} - \sum_{(i,k,t')\in F} z_{ikt'} - \sum_{i\in F_loc}\sum_{(j,t)\in\mathcal{D}} w_{i,jt})$. The connection cost of (j,t) is the connection cost for the njt copies connected in Phase 1 added to the connection cost for the rjt' copies connected in Phase 2. Each copy of (j,t) connected in Phase 2 incurs a connection cost of at most $3\alpha_{jt}$. Hence, the total cost is bounded by (cost of phase 1) + (facility cost in phase 2) + (connection cost for r'_{jt} copies in phase 2)

$$\leq (\sum_{(j,t)\in\mathcal{D}} n_{jt}\alpha_{jt} - \sum_{(i,k,t')\in \text{Ł}_1} z_{ik't} - \sum_{i\in\mathcal{F}_{loc}:(i,k,t')\in L_1} \sum_{(j,t)\in\mathcal{D}} w_{i,jt}) + 3 \cdot maxL \cdot \sum_{(i,k,t)\in\mathcal{F}} f_i^k y_{ikt}$$
$$+ 3 \sum_{(j,t)\in\mathcal{D}} r'_{jt}\alpha_{jt} \leq (\sum_{(j,t)\in\mathcal{D}} n_{jt}\alpha_{jt} - \sum_{(i,k,t')\in\mathcal{F}} z_{ik't} - \sum_{i\in\mathcal{F}_{loc}:(i,k,t')\in L_1} \sum_{(j,t)\in\mathcal{D}} w_{i,jt}) +$$
$$3 \cdot maxL \cdot \sum_{(j,t)\in\mathcal{D}} r'_{jt}\alpha_{jt} + (\sum_{(j,t)\in\mathcal{D}} n_{jt}\alpha_{jt} - \sum_{(i,k,t')\in F} z_{ikt'} - \sum_{i\in F_loc}\sum_{(j,t)\in\mathcal{D}} w_{i,jt}) + 3 \sum_{(j,t)\in\mathcal{D}} r'_{jt}\alpha_{jt} \tag{5.14}$$

We comment that we can substitute $\sum_{(i,k,t')\in L_1} z_{ikt'}$ with $\sum_{(i,k,t')\in\mathcal{F}} z_{ikt'}$, because of the complementary slackness condition $z_{ikt} > 0 \Rightarrow y_{ikt} = 1$, meaning that only triplets members of the L_1 can have positive variables z, whereas we cannot apply the same for w, sine the dual conditions state $w_{i,jt} > 0 \Rightarrow \sum_{(i,k,t')} x_{ikt',jt} = 1$. This permanence of L_1 instead of $\mathcal{F}$ in the negative factor $\sum_{i\in\mathcal{F}_{loc}:(i,k,t')\in L_1} \sum_{(j,t)\in\mathcal{D}} w_{i,jt}$ means that we can not somehow group the result so as to bound the algorithm solution with respect to the optimal value $OPT = \sum_{(j,t)\in\mathcal{D}} r'_{jt}\alpha_{jt} + (\sum_{(j,t)\in\mathcal{D}} n_{jt}\alpha_{jt} - \sum_{(i,k,t')\in F} z_{ikt'} - \sum_{i\in F_loc}\sum_{(j,t)\in\mathcal{D}} w_{i,jt})$.

This problem is once again located in the time aspect of the leasing problem, since the dual variable w corresponds to condition $1 \geq \sum_{(i,k,t')\in\mathcal{F}} x_{ikt',jt}$ which was introduced in order to assert that the same facility with different leases would not count as multiple connections for a client on the same day. Observing that discerning between leases of the same facility w.r.t. a certain client on one day was necessary and that there was no obvious way we could somehow handle or "hide" the negative term $\sum_{i\in\mathcal{F}_{loc}:(i,k,t')\in L_1} \sum_{(j,t)\in\mathcal{D}} w_{i,jt}$, i.e. by somehow combining $b_{ikt',jt}$ and $w_{i,jt}$, because $b_{ikt',jt}$'s are explicitly expressed in condition $\sum_{(j,t)\in\mathcal{D}} \beta_{ikt',jt} \leq f_i^k + z_{ikt'}$.

Therefore, we decided to "break" the leasing problem in an intermediate problem, where leases would be *continuous*. We kept the same fault-tolerant setting as before, where clients can have different demands per day, but leases now are considered continuous, ie. a facility if chosen can be opened only in the beginning of time t_0 and is open continuously until the day it is decided to be closed, thus ending the lease, never to be opened again. If a satisfying algorithm were found for this problem, then we could inverse it, i.e. if a facility were to be leased, then the leasing could start any day, but should last till the end of time T, which is known. The goal was was to come up with an approximate solution for the Fault-Tolerant Facility Leasing, by combining those two subproblems and making some speculations on the permitted lease types according to the Nested and Interval Model theorems.

The intermediate problem was initially modeled in the following relaxation:

$$\text{minimize} \sum_{(j,t)\in\mathcal{D}} \sum_{i\in\mathcal{F}_{loc}} c_{ij} x_{i,jt} + \sum_{i\in\mathcal{F}_{loc}} \sum_{t} t \cdot f_i \cdot y_{it}$$

$$\begin{aligned}
\text{subject to:}\quad & y_{it} - x_{i,jt'} \geq 0 && \forall (i,t) \in \mathcal{F},\ (j,t') \in \mathcal{D}:\ t' \geq t && (5.15)\\
& \sum_{i\in\mathcal{F}_{loc}} x_{i,jt} \geq r_{j(t)} && \forall (j,t) \in \mathcal{D} && (5.16)\\
& x_{i,jt}, y_{it} \geq 0 && \forall (i,t) \in \mathcal{F},\ (j,t) \in \mathcal{D} && (5.17)
\end{aligned}$$

where y_{it} denotes "how much" facility i is open on day t. However, this was substituted by the following more convenient LP formulation:

$$\text{minimize} \sum_{(j,t)\in\mathcal{D}} \sum_{i\in\mathcal{F}_{loc}} c_{ij} x_{i,jt} + \sum_{(i,t)\in\mathcal{F}} g_i(t) \cdot y_{it}$$

$$\begin{aligned}
\text{subject to:}\quad & y_{it} - x_{it,jt'} \geq 0 && \forall (i,t) \in \mathcal{F},\ (j,t') \in \mathcal{D}:\ t' \leq t && (5.18)\\
& \sum_{(i,t')\in\mathcal{F}} x_{it',jt} \geq r_{j(t)} && \forall (j,t) \in \mathcal{D} && (5.19)\\
& 1 \geq y_{it} && \forall (i,t) \in \mathcal{F} && (5.20)\\
& x_{it,jt'},\ y_{it} \geq 0 && \forall (i,t) \in \mathcal{F},\ (j,t') \in \mathcal{D} && (5.21)
\end{aligned}$$

where variable y_{it} still denotes "how much" facility i is open on day t, and all the g's where $g_i(t) = f_i(t) - f_i(t-1)$ can be considered already calculated in advance for every i and t since the (i,t) are finite. The second version led to the following dual:

$$\text{maximize} \sum_{(j,t)\in\mathcal{D}} \alpha_{jt} r_{jt} \; - \sum_{(i,t)\in\mathcal{F}} z_{it}$$

$$\text{subject to:} \quad \sum_{(j,t')\in\mathcal{D}\,:\,t\le t'} \beta_{it,jt'} \le g_i(t) + z_{it} \qquad \forall (i,t)\in\mathcal{F} \tag{5.22}$$

$$\alpha_{jt'} - \sum_{t\,:\,t\le t'} \beta_{it,jt'} \le c_{ij} \qquad \forall (j,t')\in\mathcal{D},\; i\in\mathcal{F}_{loc} \tag{5.23}$$

$$\alpha_{jt}, \beta_{it,jt'} \ge 0 \qquad \forall (j,t')\in\mathcal{D},\; (i,t)\in\mathcal{F} \tag{5.24}$$

However, neither for this intermediate problem did we manage to find an effective approximation. Here follows an example of a proposed algorithm, starting from the end day instead of the beginning, and the problems we encountered:

Let an instance of FTF with Continuous Leasing with total duration T and r_{jt} demand of client j on day t:

(algorithm presented in the next page)

Algorithm for the FTF with Continuous Leasing

1. Let t_{curr} the latest day of the problem which has not been examined yet. Start by setting $t_{curr} \leftarrow T$ and repeat until days are over (ie. we have reached the first day).

2. Run as subroutine an α-approximation algorithm for the Fault Tolerant Facility Location problem with the following input:

   - Consider as client set $\mathcal{D}$ the clients with positive demands on day t_{curr}. As their respective demands, consider $r_j \leftarrow r_{jt_{curr}}$ the demand of a client $j \in \mathcal{D}$.
   - Consider set of facilities $\mathcal{F}$ the facility *sites* of the original FTF continuous leasing problem, alongside the following opening costs:
     If a facility i is already open, that is it had already been leased on a day $t > t_{curr}$, consider then that it has zero opening cost $f_i \leftarrow 0$, otherwise consider that is has opening cost equal to the cost of leasing it up till day t_{curr}, ie. $f_i \leftarrow f_i(t_{curr})$.
   - Consider connection cost of client $j \in \mathcal{D}$ with facility i the same c_{ij} as in the original FTF continuous leasing problem.

3. According to the solution returned by the α-approximation algorithm with input the above described fault tolerant facility location instance, connect clients jt_{curr} of the original leasing problem with the facilities to which the corresponding clients $j \in \mathcal{D}$ connected according to this solution, and lease from the beginning up till day t_{curr} whichever of these connected facilities were not already open (ie. they hadn't been already leased for a period longer than t_{curr}).

The above algorithm gives an αT approximation for the Fault Tolerant Facility with Continuous Leasing.

To start with, the algorithm returns a feasible solution, since every client (j, t) with demand r_{jt} fully covers his daily demand via the solution returned by the α-approximation algorithm fault tolerant facility location problem which we run on the subset of clients with positive requirements on day t.

As for the quality of the solution, let OPT^* the optimal solution for the Fault Tolerant Facility with Continuous Leasing, SOL the solution returned by the above algorithm and SOL_t the solution returned by the α-approximation algorithm for the FTF Location when run as subroutine on day t, according to the description given in step 2 of the above algorithm. Also, consider OPT_t^{FTFLoc} the optimal solution for the FTF Location problem on day t as the instance is described in step 2.

We get that the combination of all the OPT_t^{FTFLoc} solutions gives a feasible solution to the original continuous leasing problem, since for every day t client (j,t) connects to the r_{jt} facilities of his demand while maintaining the connection costs c_{ij} of the original leasing problem and also preserving facility leasing costs f_i, since it only adds cost to the solution once if facility i will be open till day t, ie. the leasing cost $f_i(t)$ is only charged once and afterwards, for every day $t' < t$ it considers facility i *free*. Thus, $\sum_{t=1}^{T} OPT_t^{FTFLoc}$ is a feasible solution of continuous leasing $OPT^* \leq \sum_{t=1}^{T} OPT_t^{FTFLoc}$.

It stands that $SOL = \sum_{t=1}^{T} SOL_t$. Also, for every day t we have that

$$OPT_t^{FTFLoc} \leq SOL_t \leq \alpha \cdot OPT_t^{FTFLoc} \text{ and } OPT_t^{FTFLoc} \leq OPT^*$$

The last inequality needs some proving:Given that OPT_t^{FTFLoc} is the optimal solution to the FTF Location instance on day t as decribed in step 2, solution OPT_t^{FTFLoc} cannot have a greater cost than OPT^*, because some of the facilities used in the solution, satisfying clients on that day, are considered as $free$, while in OPT^* we have charged their leasing cost to some other day $t' > t$, and even for those facilities that solution OPT_t^{FTFLoc} needs to open for a lease ending on day t, those facilities have the same leasing cost $f_i(t)$ they would have in the original leasing instance too if they were to be opened up till day t, so the subroutine can freely choose the $free$ facilties for a client if they are closer, ie. have smaller connection cost, and given that connection costs are preserved, the incurred total cost OPT_t^{FTFLoc} is indeed smaller than OPT^*.

From the above inequalities we get that $SOL = \sum_{t=1}^{T} SOL_t \leq \sum_{t=1}^{T} \alpha \cdot OPT_t^{FTFLoc}$ $= \alpha \sum_{t=1}^{T} OPT_t^{FTFLoc} \leq \alpha \sum_{t=1}^{T} OPT^* = \alpha \cdot T \cdot OPT^*$.

Lastly, we also came across the following multi-leasing variant, where each lease, even on the same facility location i, could count as a unit to satisfy the demand of client j on day t, expressed with the following relaxation:

$$\text{minimize} \sum_{(j,t)\in\mathcal{D}} \sum_{(i,k,t')\in\mathcal{F}:t\in \mathrm{I}^k_{t'}} c_{ij}x_{ikt',jt} + \sum_{(i,k,t)\in\mathcal{F}} f_i^k y_{ikt}$$

$$\begin{aligned}
\text{subject to:}\quad y_{ikt'} - x_{ikt',jt} &\geq 0 & \forall (i,k,t') \in \mathcal{F},\ (j,t)\in\mathcal{D} && (5.25)\\
\sum_{(i,k,t')\in\mathcal{F}:t\in \mathrm{I}^k_{t'}} x_{ikt',jt} &\geq r_{jt} & \forall (j,t)\in\mathcal{D} && (5.26)\\
1 &\geq y_{ikt} & \forall (i,k,t)\in\mathcal{F} && (5.27)\\
x_{ikt',jt}, y_{ikt} &\geq 0 & \forall (i,k,t'),(i,k,t)\in\mathcal{F},\ (j,t)\in\mathcal{D} && (5.28)
\end{aligned}$$

and its dual:

$$\text{maximize} \sum_{(j,t)\in\mathcal{D}} \alpha_{jt}r_{jt} - \sum_{(i,k,t')\in\mathcal{F}} z_{ikt'}$$

$$\begin{aligned}
\text{subject to:}\quad \sum_{(j,t)\in\mathcal{D}} \beta_{ikt',jt} &\leq f_i^k + z_{ikt'} & \forall (i,k,t')\in\mathcal{F} && (5.29)\\
\alpha_{jt} - \beta_{ikt',jt} &\leq c_{ij} & \forall (j,t)\in\mathcal{D},\ (i,k,t')\in\mathcal{F},\ t\in \mathrm{I}^k_{t'} && (5.30)\\
\alpha_{jt}, \beta_{ikt',jt} &\geq 0 & \forall (j,t)\in\mathcal{D},\ (i,k,t')\in\mathcal{F} && (5.31)
\end{aligned}$$

Essentially, it looks like the formulation for the Fault=Tolerant version of Facility Leasing, but it lacks the fault-tolerant property because of lacking the condition which guarantees that each unit demand should be covered by a different facility. This multi-leasing formulation would make sense if it were considered the real-life analoque of the problem, where a company wants to promote a product and contacts agencies (aka. the facility sites i) which have as employees promoters each working on his own shift (aka. the leases), with the purpose of *borrowing* the employees to promote the product according to the needs of each day (the requirement analogue). In that case, the company does not care if some promoters came from the same agency, as long as they can cover their daily promotional schedule.

However, it is easily proven that the above is identical to the Fault-Tolerant Facility Location problem, if we consider each client (j, t) as a distinct client j^*_t with requirement $r^*_{j_t} = r_{jt}$, each facility (i, k, t) as a distinct facility i^*_{kt} with opening cost $f^*_{i_k t} = f^k_i$ and connection costs $c^*_{i_{kt'} j_t} = c_i j$ if $t \in \mathcal{I}^k_{t'}$ or infinite otherwise.

Chapter 6

Conclusions

It is notable that the vast majority of the solutions proposed to numerous FL variants is heavily based the ground-techniques used in UFLP and FTFLP . Even more prominent appears to be the impact of LP-based techniques, where in the general case primal-dual methods are more robust than dual fitting, but with weaker approximation results, whereas LP rounding and clustering algorithms seem to do better exploiting the covering nature of the problem. Local search offers a more generic approach, almost always guaranteeing results, albeit much weaker than those of the LP-based algorthims.

It is interesting to point out here that, although linear programming has played a key role in the study of algorithms for combinatorial optimization problems, especially in the case of the Uncapacitated Facility Location Problem, until recently none of these powerful LP-based techniques, such as LP-rounding and primal-dual schemas, had been successfully applied to the Capacitated version of the problem, except for the special case where all facility costs are equal. Instead, local search was the main technique to tackle the CFLP and it was not until 2014 that a linear programming relaxation managed to successfully approximate the capacitated facility location problem in [6]. By employing theory from multi-commodity flows and matchings, the authors arrived to a strong relaxation, with a constant integrality gap, and thus were the first to achieve application of a strong LP-based technique such as LP-rounding to the Capacitated version of the problem. Although their algorithm is a 288-approximation, their result is important because it unlocked a means to employ LP-based techniques to the CFLP. Consequently, with the addition of such a powerful tool as are LP-based techniques and the results concerning their applications on approximation problems in general (i.e. [23]) , it remains open how usage of them could improve approximation ratios for the CFLP.

On the other side, the work from Kolliopoulos & Moysolgou in [36] and [37] do not leave much space for essential improvement. They give negative results as far as the "computational quality" of a natural LP formulation for the Capacitated Facility Location is concerned, wrapped up in the following theorem:

Theorem. *Every approximate formulation for metric Capacitated Facility Location that uses the natural encoding and has integrality gap at most g for some constant* $g > 0$*, has* $2^{\Omega}(n \ \log n)$ *constraints.*

They furthermore strengthen the impossibility result by showing that the gap remains unbounded with the addition of effective capacity inequalities or even submodular inequalities to the relaxation for the Capacitated version, disproving this way also a conjecture from [41]. The case of proper relaxations (the equivalent to *star*-like LP relaxations in UFL) is also examined, with the same discouraging results, although they find that there are proper relaxations with a gap of 1, when classes (sets with an arbitrary number of facilities and clients together with an assignment of each client to a facility in the set) are allowed to examine the total number of feasibly openable facilities that is allowed in (classes of maximum complexity 1).

A contribution would be to take into consideration the real-world analogues of facility location variants. For example, there are some practical aspects of FTF Allocation open for future research. Suppose the scenario, i.e. in cloud service delivery, where the downtime (aka the percent of time to rest) of links is predictable. In a network where each serving node (facility) needs a fraction of time to rest, the downtime is uniformly b for all facilities and the percent of time that a client j requires service is b_j, then that network design problem can be modeled as a FTFA instance with $r_j = \lceil b_j/(1-b) \rceil$, otherwise, if the downtime is unpredictable, $r_j = \lceil \log_b(1-b_j) \rceil$. Either way, the algorithm proposed in [52] suitable. Nevertheless, if the downtime is nonuniform, the constraints on connectivity become $\sum_{i \in \mathcal{F}}(1-b_{ij})x_{ij} \geq b_j$ for the deterministic case and $\Pi_{i \in \mathcal{F}: x_{ij}=1} b_{ij} \leq 1 - b_j$ for the stochastic case, both of which remain open for future research.

As a last remark, the Leasing variants, where the notion of time is introduced, are a promising, yet challenging field. They are hard to reduct to some already known version of facility location problems, due to the addition of the dimension of time. Furthermore, the generally straightforward tool to apply in various cases of FL problem, the local search method, is not at all so straightforward in the leasing case, as there is no obvious way on how it could be implemented, so as to work on a multi-level local optima quest and grasp the dimension of time. A strategy could be to embed time/the lease lengths into the distance metric, a device similar to the one proposed in [49], where Meyerson et al. construct a Steiner tree which optimizes the sum of edge costs on one metric and the sum of source-sink distances on another metric, very much different in nature from the first one, as would be in our case the metric of connection costs and possibly a metric encoding the leases. To date, there has been no significant advancement in the Leasing aspect of the Facility Location problems since the work of Nagarajan & Williamson, although there are still open many core questions, such as whether the 3-approximation bound can drop or if there can be constructed a constant factor approximation algorithm for the fault-tolerant version.

Bibliography

[1] *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA*. ACM/SIAM, 2001.

[2] *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings*, volume 6080 of *Lecture Notes in Computer Science*. Springer, 2010.

[3] K. Aardal, F. A. Chudak, and D. B. Shmoys. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Inf. Process. Lett.*, 72(5-6), 1999.

[4] A. A. Ageev, Y. Ye, and J. Zhang. Improved combinatorial approximation algorithms for the k-level facility location problem. *SIAM J. Discrete Math.*, 18(1), 2004.

[5] A. Aggarwal, A. Louis, M. Bansal, N. Garg, N. Gupta, S. Gupta, and S. Jain. A 3-approximation for facility location with uniform capacities. In *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings* [2].

[6] H. An, M. Singh, and O. Svensson. Lp-based algorithms for capacitated facility location. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. IEEE Computer Society, 2014.

[7] E. Angel, N. K. Thang, and D. Regnault. Improved local search for universal facility location. *J. Comb. Optim.*, 29(1), 2015.

[8] B. M. Anthony and A. Gupta. Infrastructure leasing problems. In M. Fischetti and D. P. Williamson, editors, *Integer Programming and Combinatorial Optimization, 12th International IPCO Conference, Ithaca, NY, USA, June 25-27,*

2007, Proceedings, volume 4513 of *Lecture Notes in Computer Science*. Springer, 2007.

[9] M. Asadi, A. Niknafs, and M. Ghodsi. An approximation algorithm for the k-level uncapacitated facility location problem with penalties. In *Advances in Computer Science and Engineering*, volume 6 of *Communications in Computer and Information Science*. Springer Berlin Heidelberg, 2009.

[10] M. L. Balinksi. On finding integer solutions to linear program. In *Proceedings of the IBM Science Computing Symposium on Combinatorial Problems*, IBM, 1966.

[11] F. Barahona and D. Jensen. Plant location with minimum inventory. *Math. Program.*, 83, 1998.

[12] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

[13] J. Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings*, 2007.

[14] J. Byrka and K. Aardal. The approximation gap for the metric facility location problem is not yet closed. *Oper. Res. Lett.*, 35(3), 2007.

[15] J. Byrka, A. Srinivasan, and C. Swamy. Fault-tolerant facility location: A randomized dependent lp-rounding algorithm. In *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings* [2].

[16] M. Charikar and S. Guha. Improved combinatorial algorithms for facility location problems. *SIAM J. Comput.*, 34(4), 2005.

[17] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.* [1].

[18] F. A. Chudak. Improved approximation algorithms for uncapitated facility location. In *Integer Programming and Combinatorial Optimization, 6th International IPCO Conference, Houston, Texas, USA, June 22-24, 1998, Proceedings*, 1998.

[19] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput.*, 33(1), 2003.

[20] F. A. Chudak and D. P. Williamson. Improved approximation algorithms for capacitated facility location problems. *Math. Program.*, 102(2), 2005.

[21] G. L. N. G. Cornuejols and L. A. Wolsey. The uncapacitated facility location problem. In *Discrete Location Theory.* John Wiley and Sons, Inc., 1990.

[22] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding in bipartite graphs. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings.* IEEE Computer Society, 2002.

[23] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53(3), 2006.

[24] J. Geunes, R. Levi, H. E. Romeijn, and D. B. Shmoys. Approximation algorithms for supply chain planning and logistics problems with market choice. *Math. Program.*, 130(1), 2011.

[25] M. X. Goemans, A. V. Goldberg, S. A. Plotkin, D. B. Shmoys, É. Tardos, and D. P. Williamson. Improved approximation algorithms for network design problems. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. 23-25 January 1994, Arlington, Virginia.* ACM/SIAM, 1994.

[26] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *J. Algorithms*, 31(1), 1999.

[27] S. Guha, A. Meyerson, and K. Munagala. Improved algorithms for fault tolerant facility location. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.* [1].

[28] M. T. Hajiaghayi, M. Mahdian, and V. S. Mirrokni. The facility location problem with general cost functions. *Networks*, 42(1), 2003.

[29] A. Hayrapetyan, C. Swamy, and É. Tardos. Network design for information networks. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*. SIAM, 2005.

[30] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(2), 1982.

[31] V. N. Hsu, T. J. Lowe, and A. Tamir. Structured p-facility location problems on the line solvable in polynomial time. *Oper. Res. Lett.*, 21(4), 1997.

[32] 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*, 50(6), 2003.

[33] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *STOC*. ACM, 2002.

[34] K. Jain and V. V. Vazirani. An approximation algorithm for the fault tolerant metric facility location problem. *Algorithmica*, 38(3), 2003.

[35] J. K. and V. V. V. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press.

[36] S. G. Kolliopoulos and Y. Moysoglou. Exponential lower bounds on the size of approximate formulations in the natural encoding for capacitated facility location. *CoRR*, abs/1312.1819, 2013.

[37] S. G. Kolliopoulos and Y. Moysoglou. Sherali-adams gaps, flow-cover inequalities and generalized configurations for capacity-constrained facility location. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, volume 28 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

[38] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Algorithms*, 37(1), 2000.

[39] R. Krishnaswamy and M. Sviridenko. Inapproximability of the multi-level uncapacitated facility location problem. In *Proceedings of the Twenty-Third Annual*

ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012. SIAM, 2012.

[40] A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. (9), 1963.

[41] R. Levi, D. B. Shmoys, and C. Swamy. Lp-based approximation algorithms for capacitated facility location. *Math. Program.*, 131(1-2):365–379, 2012.

[42] S. Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, 2011.

[43] Y. Li, D. Du, N. Xiu, and D. Xu. Improved approximation algorithms for the facility location problems with linear/submodular penalty. In *Computing and Combinatorics, 19th International Conference, COCOON 2013, Hangzhou, China, June 21-23, 2013. Proceedings*, volume 7936 of *Lecture Notes in Computer Science*. Springer, 2013.

[44] J. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Inf. Process. Lett.*, 44(5), 1992.

[45] M. Mahdian and M. Pál. Universal facility location. In *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, 2003.

[46] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. In *Approximation Algorithms for Combinatorial Optimization, 5th International Workshop, APPROX 2002, Rome, Italy, September 17-21, 2002, Proceedings*, 2002.

[47] A. S. Manne. Plant location under economies of scale decentralization and computation. (11), 1964.

[48] A. Meyerson. The parking permit problem. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, 2005.

[49] A. Meyerson, K. Munagala, and S. A. Plotkin. Cost-distance: Two metric network design. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*. IEEE Computer Society, 2000.

[50] C. Nagarajan and D. P. Williamson. Offline and online facility leasing. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *Integer Programming and Combinatorial Optimization, 13th International Conference, IPCO 2008, Bertinoro, Italy, May 26-28, 2008, Proceedings*, volume 5035 of *Lecture Notes in Computer Science*. Springer, 2008.

[51] G. Sahin and H. Süral. A review of hierarchical facility location models. *Computers & OR*, 34(8), 2007.

[52] H. Shen and S. Xu. Approximation algorithms for fault tolerant facility allocation. *SIAM J. Discrete Math.*, 27(3), 2013.

[53] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, 1997.

[54] J. F. Stollsteimer. A working model for plant numbers and locations. (45), 1963.

[55] M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. In *IPCO*, volume 2337 of *Lecture Notes in Computer Science*. Springer, 2002.

[56] C. Swamy and D. B. Shmoys. Fault-tolerant facility location. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA.*, 2003.