Approximation Algorithms for Facility Location Problems
(Lecture Notes)

Jens Vygen

Research Institute for Discrete Mathematics, University of Bonn
Lennéstraße 2, 53113 Bonn, Germany

Abstract

This paper surveys approximation algorithms for various facility location problems, mostly with detailed proofs. It resulted from lecture notes of a course held at the University of Bonn in the winter term 2004/2005.

Contents

1 Introduction **2**

2 The Fermat-Weber Problem **3**
2.1 Geometric Solution for Three Points 3
2.2 The General Problem: Weiszfeld's Algorithm 6
2.3 Completing Weiszfeld's Algorithm 7
2.4 Proving Convergence 10

3 The Uncapacitated Facility Location Problem **13**
3.1 Relation to Set Covering 14
3.2 Metric Service Costs 15
3.3 Notation 15
3.4 Linear Programming 16
3.5 Rounding the LP solution 18

4 Greedy and Primal-Dual Algorithms **21**
4.1 Two Primal-Dual Algorithms 21
4.2 The Jain-Mahdian-Saberi Algorithm 23
4.3 Scaling and Greedy Augmentation 26
4.4 Lower Bound on Approximation Guarantees 28

5 Reductions for More General Problems 32
5.1 Soft Capacities 32
5.2 Bounding the Number of Facilities 34

6 Local Search 37
6.1 Single Swaps for the k-Median Problem 37
6.2 Better Performance Guarantee by Multiswaps 40
6.3 Local Search for Uncapacitated Facility Location 42

7 Capacitated Facility Location 44
7.1 Capacitated and Universal Facility Location 44
7.2 Add Operation 46
7.3 Pivot Operation 47
7.4 Bounding the Facility Cost 49
7.5 The Performance Guarantee 52

8 Conclusions 53

References 54

1 Introduction

Many economical decision problems concern selecting and/or placing certain facilities to serve given demands efficiently. Examples are manufacturing plants, storage facilities, depots, warehouses, libraries, fire stations, hospitals, base stations for wireless services (like TV broadcasting or mobile phone service), etc. The problems have in common that a set of facilities, each with a certain position, has to be chosen, and the objective is to meet the demand (of customers, users etc.) best. Facility location problems, which occur also in less obvious contexts, indeed have numerous applications.

The most widely studied model in discrete facility location is the so-called UNCAPACITATED FACILITY LOCATION PROBLEM, also known as plant location problem or warehouse location problem. Here we are given two finite sets of customers and potential facilities, respectively, a fixed cost associated with opening each single facility, and a nonnegative distance for any two elements, satisfying the triangle inequality. The goal is to select a subset of the potential facilities (open them) and assign each customer to a selected (open) facility, such that the total opening cost plus the total service cost is minimum.

Although intensively studied since the 1960s (see, e.g., Stollsteimer [1963], Balinski and Wolfe [1963], Kuehn and Hamburger [1963], Manne [1964]), no approximation algorithm was known for this problem until about ten years ago. Then several quite different approaches succeeded to prove an approximation guarantee. We will present them in this paper, and also consider extensions to more general problems, such as capacitated variants, the k-MEDIAN PROBLEM, and the UNIVERSAL FACILITY LOCATION PROBLEM.

However, we start with the FERMAT-WEBER PROBLEM, which was probably historically the first facility location problem, studied as early as in the 17th century. This is the simplest continuous facility location model, but there are still questions resolved only very recently.

2 The Fermat-Weber Problem

The FERMAT-WEBER PROBLEM is defined as follows: Given finitely many distinct points $A_1, A_2, \ldots, A_m$ in $\mathbb{R}^n$ and positive multipliers $w_1, w_2, \ldots, w_m \in \mathbb{R}_+$, find a point $P \in \mathbb{R}^n$ that minimizes

$$f(P) \quad = \quad \sum_{i=1}^{m} w_i ||P - A_i||.$$

Here $||X||$ denotes the Euclidean norm of $X \in \mathbb{R}^n$, i.e. $||(x_1, \ldots, x_n)|| = \sqrt{x_1^2 + \cdots + x_n^2}$.

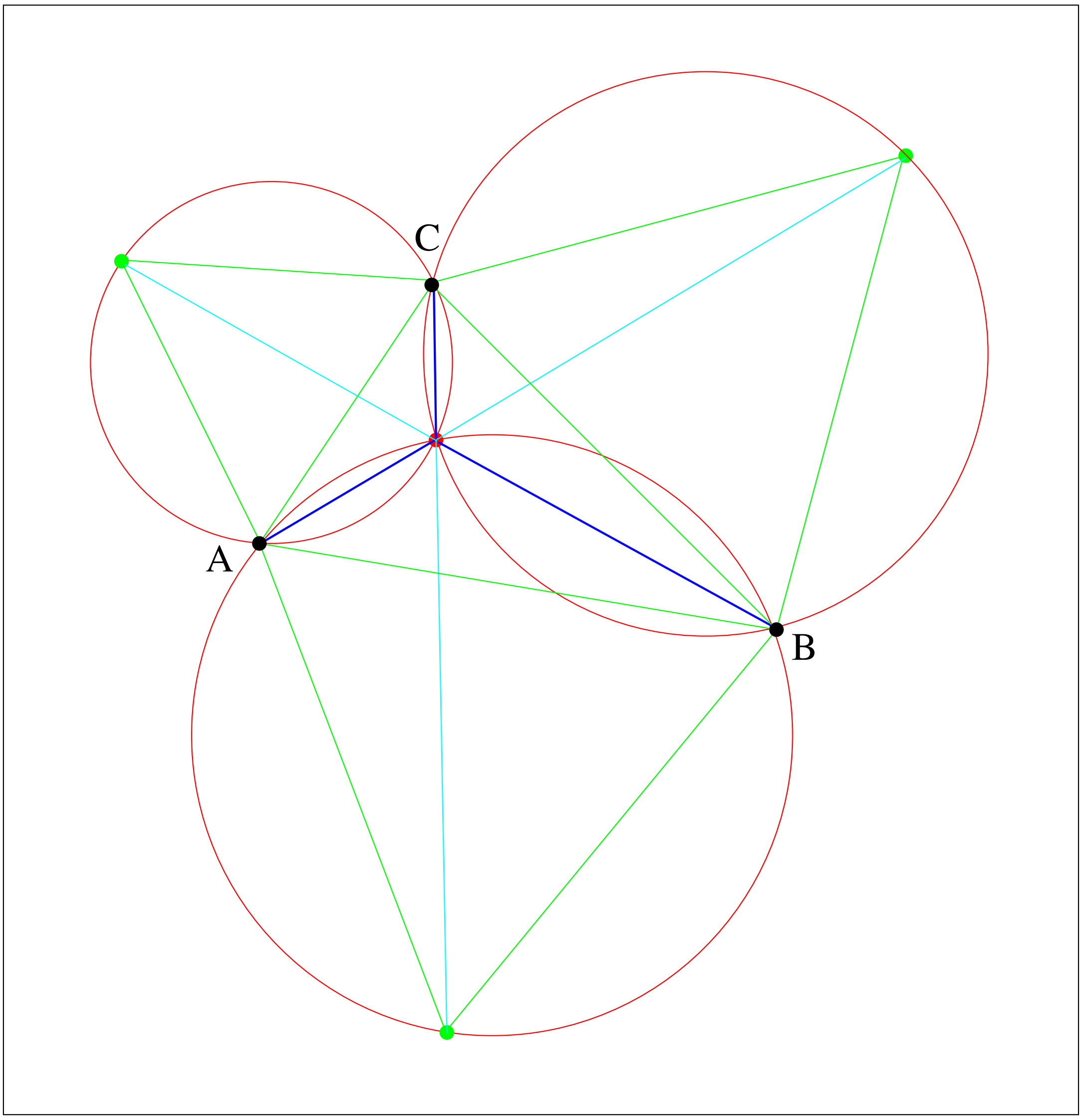
Special cases often considered are the ones with unit weights ($w_i = 1$ for all i) and with points in the plane ($n = 2$). So the simplest version of the problem is: Given m points in the plane, find a point x such that the sum of the (Euclidean) distances from x to the given points is minimum.

2.1 Geometric Solution for Three Points

We first consider the case $m = 3$ (with unit weights), proposed by Pierre de Fermat (1601–1665): Given three points A, B, and C in the plane, construct a fourth point D minimizing the sum of the distances A-D, B-D, and C-D. The history of this problem can be sketched as follows.

Before 1640, Torricelli showed that D can be constructed as the intersection of the circumcircles bounding the equilateral triangles on A-B, B-C and C-A. Cavalieri (1647) showed that the three angles at D are all 120°. Simpson (1750) proved that the line segments from the far corner of the equilateral triangles to the opposite corner of the triangle ABC intersect in D, too. Heinen (1834) showed that the three Simpson-lines have equal length, as has the optimum Steiner tree.

The geometric solution by Torricelli (red) and Simpson (blue) is visualized by the following figure. It works if and only if all angles of the triangle ABC are less than 120°.

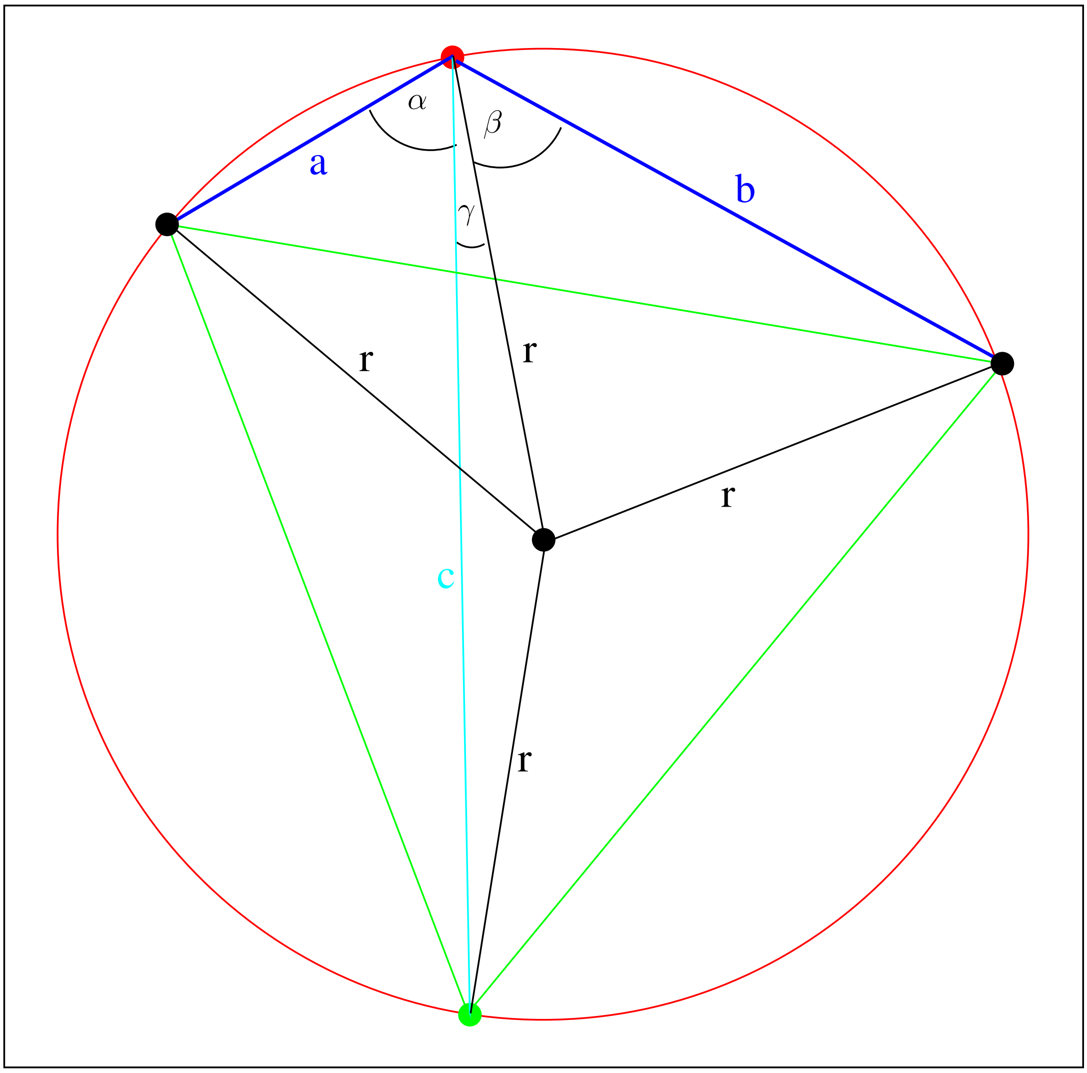


To prove correctness, we look at the lower equilateral (green) triangle and its (red) circumcircle. We show (see next figure):

(a) $\alpha = \beta + \gamma = 60°$.

(b) $a + b = c$.

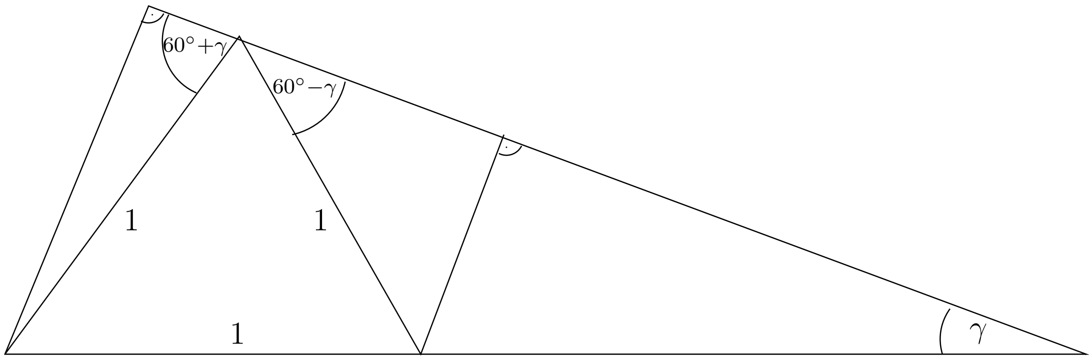
(a) shows that the constructions by Torricelli and Simpson are identical, and the resulting point has three 120° angles. (b) is Heinen's result.



The proof uses elementary trigonometry only.

To prove (a), consider the angle sums of the triangles to the left and to the right of the Simpson line c. To the left, we have (starting at α and continuing clockwise) $\alpha + (30° - \gamma) + (30° + 30° + \alpha + \gamma - 30°) = 180°$, implying $\alpha = 60°$. To the right, we have $(\gamma + \beta) + (\beta - 30° + 30° + 30°) + 30° + \gamma = 180°$, implying $\gamma + \beta = 60°$.

To prove (b), observe that $\cos(\alpha + \gamma) = \frac{a}{2r}$, $\cos \beta = \frac{b}{2r}$, and $\cos \gamma = \frac{c}{2r}$. Using $\cos(60° + \gamma) + \cos(60° - \gamma) = \cos \gamma$ for $\gamma \in [0°, 30°]$ (which can be seen easily from the figure below), this implies $a + b = c$. □



2.2 The General Problem: Weiszfeld's Algorithm

The general Fermat-Weber Problem has been studied by Simpson (1750), and in the 19th century, among others, by Weber and Steiner. Weiszfeld [1937] proposed an algorithm which we will discuss below. Bajaj [1988] proved that a construction by ruler and compasses only is impossible.

Note that the Fermat-Weber Problem reduces to a weighted median problem (see Korte and Vygen [2000], Section 17.1) if the given points are collinear. Thus this special case can be solved in linear time.

(This also implies that the variant of the Fermat-Weber Problem where we consider ℓ_1-norm instead of Euclidean norm can be solved in linear time. Namely, in this case we can solve the problem separately for each coordinate. This also holds if we minimize the sum of squared Euclidean distances, in which case the problem reduces to finding the center of gravity.)

Therefore we assume henceforth that the given points are not collinear. Then we note (cf. Kuhn [1973]):

Proposition 2.1 *If the points are not collinear, then f is strictly convex.*

Proof: For $P, Q, A \in \mathbb{R}^n$ with $P \neq Q$ and $0 < \lambda < 1$ the Cauchy-Schwarz inequality implies

$$\begin{aligned} & ||\lambda(P-A) + (1-\lambda)(Q-A)||^2 \\ = \; & ||\lambda(P-A)||^2 + 2(\lambda(P-A))^T((1-\lambda)(Q-A)) + ||(1-\lambda)(Q-A)||^2 \\ \leq \; & ||\lambda(P-A)||^2 + 2||\lambda(P-A)|| \cdot ||(1-\lambda)(Q-A)|| + ||(1-\lambda)(Q-A)||^2 \\ = \; & (\lambda||P-A|| + (1-\lambda)||Q-A||)^2 \end{aligned}$$

with strict inequality if and only if P, Q and A are not collinear. Hence

$$f(\lambda P + (1-\lambda)Q) < \lambda f(P) + (1-\lambda) f(Q),$$

i.e. f is strictly convex. □

Thus f is minimized at a unique point $M \in \mathbb{R}^n$. The idea behind Weiszfeld's algorithm, starting anywhere and trying to converge to M, is actually very simple. If $P \notin \mathcal{A} = \{A_1, A_2, \ldots, A_m\}$, then the negative of the gradient of f at P equals

$$R(P) \quad = \quad \sum_{i=1}^{m} w_i \frac{A_i - P}{||A_i - P||}. \tag{1}$$

Hence, if also $M \notin \mathcal{A}$, then necessarily $R(M) = 0$ which is equivalent to

$$M \quad = \quad \frac{\sum_{i=1}^{m} \frac{w_i A_i}{||M - A_i||}}{\sum_{i=1}^{m} \frac{w_i}{||M - A_i||}}. \tag{2}$$

In view of (2), Weiszfeld [1937] defined $T(P)$ — the next iterate — for given some $P \notin \mathcal{A}$ as

$$T(P) \quad = \quad \frac{\sum_{i=1}^{m} \frac{w_i A_i}{||P - A_i||}}{\sum_{i=1}^{m} \frac{w_i}{||P - A_i||}} \tag{3}$$

and claimed that the sequence of points

$$P_0, T(P_0), T^2(P_0) = T(T(P_0)), T^3(P_0), \ldots$$

converges to the optimum solution M for every choice of the starting point P_0. He ignored the possibility that $T^s(P_0) \in \mathcal{A}$ for some $s \geq 0$ in which case (3) would not be defined.

2.3 Completing Weiszfeld's Algorithm

Kuhn [1973] observed that it is easy to decide whether some $A_i \in \mathcal{A}$ is the optimum point M. Setting

$$R_k \quad = \quad \sum_{i=1, i \neq k}^{m} w_i \frac{A_i - A_k}{||A_i - A_k||}$$

for $1 \leq k \leq m$, we have

$$\frac{d}{dt} f(A_k + tZ) \Big|_{t=0} \quad = \quad w_k - R_k^T Z$$

for $Z \in \mathbb{R}^n$ with $||Z|| = 1$. Therefore, the direction of greatest decrease of f at A_k is $Z = \frac{R_k}{||R_k||}$, in which case $\frac{d}{dt} f(A_k + tZ)|_{t=0}$ equals $w_k - ||R_k||$, and we immediately obtain the following observation.

Proposition 2.2 *$A_k = M$ if and only if $w_k \geq ||R_k||$.* □

Now let us assume that $A_k \in \mathcal{A}$ is not the optimum point. Proposition 2.2 implies that $w_k - ||R_k|| < 0$ and we can really decrease the value of f by moving from A_k a little bit in the direction $\frac{R_k}{||R_k||}$. To make this quantitatively precise, we consider $\frac{d}{dt} f\left(A_k + t\frac{R_k}{||R_k||}\right)$ for $t \geq 0$. For $Z = \frac{R_k}{||R_k||}$ elementary calculus yields

$$\begin{aligned}
\frac{d}{dt} f(A_k + tZ) &= \left(\sum_{i=1, i\neq k}^{m} w_i \frac{A_k + tZ - A_i}{||A_k + tZ - A_i||}\right)^T Z + w_k \\
&= \sum_{i=1, i\neq k}^{m} \frac{w_i}{||A_k + tZ - A_i||} \left((A_k - A_i)^T Z + t\right) + w_k \\
&= \frac{R_k^T}{||R_k||} \sum_{i=1, i\neq k}^{m} \frac{w_i}{||A_k + tZ - A_i||} (A_k - A_i) \\
&\quad + t \sum_{i=1, i\neq k}^{m} \frac{w_i}{||A_k + tZ - A_i||} + w_k.
\end{aligned}$$

Defining the vector $V_k(t)$ by the following equation

$$-R_k + V_k(t) = \sum_{i=1, i\neq k}^{m} \frac{w_i}{||A_k + tZ - A_i||} (A_k - A_i)$$

we obtain for $t \leq t'_k$ with

$$t'_k = \min\left\{\frac{1}{2}||A_k - A_i|| : 1 \leq i \leq m, i \neq k\right\} > 0$$

that

$$\begin{aligned}
||V_k(t)|| &= \left|\left|\sum_{i=1, i\neq k}^{m} \frac{w_i}{||A_k + tZ - A_i||}(A_k - A_i) + R_k\right|\right| \\
&= \left|\left|\sum_{i=1, i\neq k}^{m} w_i (A_k - A_i)\left(\frac{1}{||A_k + tZ - A_i||} - \frac{1}{||A_k - A_i||}\right)\right|\right| \\
&\leq \sum_{i=1, i\neq k}^{m} w_i ||A_k - A_i|| \left|\frac{||A_k - A_i|| - ||A_k + tZ - A_i||}{||A_k + tZ - A_i|| ||A_k - A_i||}\right| \\
&\leq \sum_{i=1, i\neq k}^{m} w_i ||A_k - A_i|| \left|\frac{t}{\left(||A_k - A_i|| - t\right) ||A_k - A_i||}\right| \\
&\leq \left(\sum_{i=1, i\neq k}^{m} \frac{2w_i}{||A_k - A_i||}\right) t.
\end{aligned}$$

Similarly, for $t \leq t'_k$ we have

$$\begin{aligned}\sum_{i=1, i\neq k}^{m} \frac{w_i}{||A_k + tZ - A_i||} &\leq \sum_{i=1, i\neq k}^{m} \frac{w_i}{||A_k - A_i|| - t} \\ &\leq \sum_{i=1, i\neq k}^{m} \frac{2w_i}{||A_k - A_i||}.\end{aligned}$$

Putting everything together, we obtain

$$\begin{aligned}\frac{d}{dt} f\left(A_k + t\frac{R_k}{||R_k||}\right) &\leq \frac{R_k^T}{||R_k||}(-R_k + V_k(t)) + \left(\sum_{i=1, i\neq k}^{m} \frac{2w_i}{||A_k - A_i||}\right) t + w_k \\ &\leq w_k - ||R_k|| + ||V_k(t)|| + \left(\sum_{i=1, i\neq k}^{m} \frac{2w_i}{||A_k - A_i||}\right) t \\ &\leq w_k - ||R_k|| + \left(\sum_{i=1, i\neq k}^{m} \frac{4w_i}{||A_k - A_i||}\right) t.\end{aligned}$$

If we define

$$t_k = \max\left\{0, \min\left\{t'_k, \frac{||R_k|| - w_k}{\sum\limits_{i=1, i\neq k}^{m} \frac{4w_i}{||A_k - A_i||}}\right\}\right\}$$

for all $1 \leq k \leq m$, then Proposition 2.2 together with the above estimates and the mean value theorem imply the following lemma.

Lemma 2.3 *Using the above notation, $t_k = 0$ for some $1 \leq i \leq k$ if and only if $A_k = M$. Furthermore,*

$$f(A_k) > f\left(A_k + t_k\frac{R_k}{||R_k||}\right)$$

for all $A_k \in \mathcal{A}$ with $A_k \neq M$. □

In view of Lemma 2.3, Rautenbach et al. [2004] proposed the following (non-continuous) extension T^* of T to $\mathbb{R}^n$:

$$T^*(P) = \begin{cases} T(P) & , P \in \mathbb{R}^n \setminus \mathcal{A} \\ A_k + t_k\frac{R_k}{||R_k||} & , P = A_k, 1 \leq k \leq m. \end{cases} \quad (4)$$

We will show that the sequence $P_0, T^*(P_0), T^*(T^*(P_0)), \ldots$ converges to the optimum solution M for every point $P_0 \in \mathbb{R}^n$.

2.4 Proving Convergence

We start with some simple observations:

Lemma 2.4 *Let $P \in \mathbb{R}^n$.*

(i) $T^*(P) = P$ if and only if $P = M$.

(ii) If $T^*(P) \neq P$, then $f(T^*(P)) < f(P)$.

Proof: Part (i) is trivial in view of the definition of T^*, (1), (2), (3), Lemma 2.3 and the strict convexity of f. For $P \in \mathcal{A}$, part (ii) follows immediately from Lemma 2.3.

Hence it remains to prove part (ii) for $P \notin \mathcal{A}$. Clearly, $T^*(P) = T(P)$ is the unique minimum of the strictly convex function

$$g_P(Q) = \sum_{i=1}^{m} \frac{w_i ||Q - A_i||^2}{||P - A_i||}$$

which implies $g_P(T(P)) < g_P(P)$. Now, $g_P(P) = f(P)$ and

$$\begin{aligned}
g_P(T(P)) &= \sum_{i=1}^{m} \frac{w_i ||T(P) - A_i||^2}{||P - A_i||} \\
&= \sum_{i=1}^{m} \frac{w_i \left((||T(P) - A_i|| - ||P - A_i||) + ||P - A_i|| \right)^2}{||P - A_i||} \\
&= \sum_{i=1}^{m} \frac{w_i \left(||T(P) - A_i|| - ||P - A_i|| \right)^2}{||P - A_i||} + 2(f(T(P)) - f(P)) + f(P) \\
&\geq 2f(T(P)) - f(P).
\end{aligned}$$

Combining these (in)equalities implies $f(T(P)) < f(P)$ and the proof is complete. □

Lemma 2.5 $\lim_{\substack{P \to A_k \\ P \neq A_k}} \frac{||T^*(P) - A_k||}{||P - A_k||} = \frac{||R_k||}{w_k}$ *for $k = 1, \ldots, m$.*

Proof:

$$\begin{aligned}
\lim_{\substack{P \to A_k \\ P \neq A_k}} \frac{||T^*(P) - A_k||}{||P - A_k||} &= \lim_{\substack{P \to A_k \\ P \notin \mathcal{A}}} \frac{||T(P) - A_k||}{||P - A_k||} \\
&= \lim_{\substack{P \to A_k \\ P \neq A_k}} \frac{\left\| \frac{\sum_{i=1}^{m} \frac{w_i A_i}{||P - A_i||}}{\sum_{i=1}^{m} \frac{w_i}{||P - A_i||}} - A_k \right\|}{||P - A_k||} \\
&= \lim_{\substack{P \to A_k \\ P \neq A_k}} \frac{\left\| \frac{\sum_{i=1}^{m} \frac{w_i A_i}{||P - A_i||}}{\sum_{i=1}^{m} \frac{w_i}{||P - A_i||}} - \frac{\sum_{i=1}^{m} \frac{w_i A_k}{||P - A_i||}}{\sum_{i=1}^{m} \frac{w_i}{||P - A_i||}} \right\|}{||P - A_k||}
\end{aligned}$$

$$
\begin{aligned}
&= \lim_{\substack{P\to A_k\\ P\neq A_k}} \frac{\left\|\sum_{i=1}^{m} \frac{w_i(A_i-A_k)}{||P-A_i||}\right\|}{\sum_{i=1}^{m} \frac{w_i||P-A_k||}{||P-A_i||}} \\
&= \frac{\lim_{\substack{P\to A_k\\ P\neq A_k}} \left\|\sum_{i=1}^{m} \frac{w_i(A_i-A_k)}{||P-A_i||}\right\|}{\lim_{\substack{P\to A_k\\ P\neq A_k}} \sum_{i=1}^{m} \frac{w_i||P-A_k||}{||P-A_i||}} \\
&= \frac{||R_k||}{w_k}
\end{aligned}
$$

□

Thus we get:

Lemma 2.6 *If $A_k \neq M$ for some $1 \leq k \leq m$, then there are $\epsilon, \delta > 0$ such that*

$$
||T^*(P) - A_k|| \geq \begin{cases} (1+\epsilon)||P - A_k|| & , P \in \mathbb{R}^n,\ 0 < ||P - A_k|| \leq \delta \\ \delta & , P = A_k. \end{cases}
$$

Proof: Since $A_k \neq M$, Proposition 2.2 implies $\frac{||R_k||}{w_k} > 1$. Therefore, the existence of ϵ and δ with the desired properties for P with $||P - A_k|| > 0$ follows from Lemma 2.5. Choosing without loss of generality $\delta \leq t_k$ and observing that $||T^*(A_k) - A_k|| = t_k > 0$, by Lemma 2.3, completes the proof. □

We can now proceed to the main convergence result, due to Rautenbach et al. [2004]:

Theorem 2.7 *If $P_0 \in \mathbb{R}^n$ and $P_k = T^*(P_{k-1})$ for all $k \in \mathbb{N}$, then $\lim_{k\to\infty} P_k = M$.*

Proof: Since $\lim_{||P||\to\infty} f(P) = \infty$ and $(f(P_k))_{k\geq 0}$ is non-negative and non-increasing, the sequence $(P_k)_{k\geq 0}$ is bounded and the sequence $(f(P_k))_{k\geq 0}$ converges. The main observation is expressed in the following claim.

Claim A subsequence of $(P_k)_{k\geq 0}$ converges to a point in $\mathbb{R}^n \setminus (\mathcal{A} \setminus \{M\})$.

Proof of the claim: For contradiction we assume that the set $\mathcal{A}'$ of all accumulation points of the bounded sequence $(P_k)_{k\geq 0}$ is a subset of $\mathcal{A} \setminus \{M\}$. This implies that for every $\delta > 0$ there are only finitely many elements of the sequence $(P_k)_{k\geq 0}$ outside of the union

$$
\bigcup_{A\in\mathcal{A}'} U_\delta(A)
$$

of open δ-neighbourhoods $U_\delta(A) = \{Q \in \mathbb{R}^n : ||Q - A|| < \delta\}$ of the elements A in $\mathcal{A}'$.

Lemma 2.6 easily implies that $|\mathcal{A}'| \geq 2$. Now the pigeonhole principle together with the finiteness of $\mathcal{A}'$ and once again Lemma 2.6 imply the existence of two distinct elements $A_1, A_2 \in \mathcal{A}'$ and a subsequence $(P_{k_l})_{l\geq 0}$ of $(P_k)_{k\geq 0}$ contained in $\mathbb{R}^n \setminus \mathcal{A}$ with

$$\begin{aligned} A_1 &= \lim_{l\to\infty} P_{k_l}, \\ A_2 &= \lim_{l\to\infty} T^*(P_{k_l}). \end{aligned}$$

Since $\lim_{P\to A, P\neq A} T^*(P) = A$ for all $A \in \mathcal{A}$ (cf. Lemma 2.5)), this implies the contradiction $A_1 = A_2$ and the proof of the claim is complete.

By the claim there is a convergent subsequence $(P_{k_l})_{l\geq 0}$ of $(P_k)_{k\geq 0}$ whose limit

$$P = \lim_{l\to\infty} P_{k_l}$$

lies outside of $\mathcal{A} \setminus \{M\}$ which implies that T^* is continuous at P.

By the convergence of $(f(P_k))_{k\geq 0}$, we have

$$\lim_{k\to\infty} f(P_k) = \lim_{k\to\infty} f(T^*(P_k)).$$

By the continuity of T^* at P, we have

$$\lim_{l\to\infty} T^*(P_{k_l}) = T^*(P).$$

Since f is continuous

$$f(P) = \lim_{l\to\infty} f(P_{k_l}) = \lim_{l\to\infty} f(T^*(P_{k_l})) = f(T^*(P))$$

and Lemma 2.4 implies $P = T^*(P) = M$.

Finally, the fact that $(f(P_k))_{k\geq 0}$ converges and f is strictly convex, implies that not just a subsequence of $(P_k)_{k\geq 0}$ converges to M but the whole sequence and the proof is complete. □

A different extension of Weiszfeld's algorithm was proposed by Vardi and Zhang [2001]. Struzyna [2004] and Szegedy [2005] partially extended Weiszfeld's algorithm to a more general problem, where some vertices of a given graph are associated with points in $\mathbb{R}^n$, and the task is to find points for the other vertices in order to minimize the total Euclidean distance of the edges of the graph. Again, the version for ℓ_1-distances or squared Euclidean distances can be solved quite easily by minimum-cost flows and linear algebra, respectively, but the problem is not fully solved for Euclidean distances.

Unfortunately, Weiszfeld's algorithm converges quite slowly. See Drezner et al. [2002] for more information, and also for other, more difficult, continuous facility location problems. We shall not consider them here.

3 The Uncapacitated Facility Location Problem

The rest of this paper deals with discrete facility location problems, where the number of possible locations is finite. The most basic problem, for which we shall present many results, is the UNCAPACITATED FACILITY LOCATION PROBLEM. It is defined as follows.

Given:

- a finite set $\mathcal{D}$ of customers (or clients);
- a finite set $\mathcal{F}$ of potential facilities;
- a fixed cost $f_i \in \mathbb{R}_+$ for opening each facility $i \in \mathcal{F}$;
- a service cost $c_{ij} \in \mathbb{R}_+$ for each $i \in \mathcal{F}$ and $j \in \mathcal{D}$;

we look for:

- a subset S of facilities (called *open*) and
- an assignment $\sigma : \mathcal{D} \to S$ of customers to open facilities,
- such that the sum of facility costs and service costs
$$\sum_{i \in S} f_i + \sum_{j \in \mathcal{D}} c_{\sigma(j)j}$$
is minimum.

The problem has been studied intensively in the operations research literature. See Cornuéjols, Nemhauser and Wolsey [1990] or Shmoys [2000] for survey papers.

We look for *approximation algorithms*, i.e. algorithms computing a feasible solution for any instance such that

- the algorithm terminates after a number of steps that is bounded by a polynomial in the instance size (e.g. in the number of customers and facilities).
- There is a constant k such that the cost of the computed solution does not exceed k times the optimum cost for any instance.

k is called the approximation ratio or performance guarantee; we speak of a k-factor approximation algorithm. If $k = 1$, we have an exact polynomial-time algorithm. However, such an algorithm would imply $P = NP$, as the problem contains many NP-hard problems. One of them will be discussed now. See Korte and Vygen [2000] for more general information on approximation algorithms.

3.1 Relation to Set Covering

The well-known Set Covering Problem is defined as follows. Given a pair $(U, \mathcal{S})$, where U is a finite set and $\mathcal{S}$ is a family of subsets of U with $\bigcup_{S \in \mathcal{S}} S = U$, and weights $c : \mathcal{S} \to \mathbb{R}_+$, the task is to find a set $\mathcal{R} \subseteq \mathcal{S}$ with $\bigcup_{S \in \mathcal{R}} S = U$ such that the total weight $\sum_{R \in \mathcal{R}} c(R)$ is minimum.

This problem is notoriously hard. Raz and Safra [1997] proved that there exists a constant $\chi > 0$ such that, unless $P = NP$, there is no polynomial-time algorithm that produces for each instance a solution whose cost is at most $\chi \ln |U|$ times the optimum. Feige [1998] proved that such an algorithm does not even exist for any $\chi < 1$ unless every problem in NP can be solved in $O(n^{O(\log \log n)})$ time. On the other hand, a simple greedy algorithm, which iteratively picks the set for which the ratio of weight over newly covered elements is minimum, yields a solution whose weight is at most $1 + \ln |U|$ times the optimum. This is a result of Chvátal [1979]; see Theorem 16.3 of Korte and Vygen [2000].

The above negative results directly transfer to the Uncapacitated Facility Location Problem. Namely, it is easy to see that the Set Covering Problem is a special case of the Uncapacitated Facility Location Problem: Given an instance $(U, \mathcal{S}, c)$ as above, define $\mathcal{D} := U$, $\mathcal{F} := \mathcal{S}$, $f_S = c(S)$ for $S \in \mathcal{S}$, and let the service cost c_{Sj} be zero for $j \in S \in \mathcal{S}$ and ∞ for $j \in U \setminus S$. Therefore the best we can hope for is a logarithmic approximation factor.

Conversely, let an instance of the Uncapacitated Facility Location Problem be given. By a *star* we mean a pair (i, D) with $i \in \mathcal{F}$ and $D \subseteq \mathcal{D}$. The *cost* of a star (i, D) is $f_i + \sum_{j \in D} c_{ij}$, and its *effectiveness* is $\frac{f_i + \sum_{j \in D} c_{ij}}{|D|}$. Then the Uncapacitated Facility Location Problem is a special case of the Minimum Weight Set Cover Problem: set $U := \mathcal{D}$ and let $\mathcal{S} = 2^{\mathcal{D}}$, where $c(D)$ is the minimum cost of a star (i, D) $(i \in \mathcal{F})$.

However, the resulting set cover instance has exponential size, and therefore this reduction cannot be used directly. Nevertheless we can apply the greedy algorithm for set covering without generating the instance explicitly, as proposed by Hochbaum [1982]:

Namely, in each step, we have to find a most effective star, open the corresponding facility and henceforth disregard all customers in this star. Although there are exponentially many stars, it is easy to find a most effective one as it suffices to consider stars (i, D_k^i) for $i \in \mathcal{F}$ and $k \in \{1, \ldots, |\mathcal{D}|\}$, where D_k^i denotes the first k customers in a linear order with nondecreasing c_{ij}. Clearly, other stars cannot be more effective. Hence we get an approximation ratio of $1 + \ln |\mathcal{D}|$ also for the Uncapacitated Facility Location Problem. In view of Feige's result mentioned above, this seems to be almost best possible.

3.2 Metric Service Costs

The previous section shows that we have to make additional assumptions in order to obtain constant-factor approximations. The usual assumption is that service costs stem from a metric, or equivalently satisfy

$$c_{ij} + c_{i'j} + c_{i'j'} \geq c_{ij'} \quad \text{for all } i, i' \in \mathcal{F} \text{ and } j, j' \in \mathcal{D}.$$

Indeed, if this condition holds, we can define $c_{ii} := 0$ and $c_{ii'} := \min_{j \in \mathcal{D}}(c_{ij} + c_{i'j})$ for $i, i' \in \mathcal{F}$, $c_{jj} := 0$ and $c_{jj'} := \min_{i \in \mathcal{F}}(c_{ij} + c_{ij'})$ for $j, j' \in \mathcal{D}$, and $c_{ji} := c_{ij}$ for $j \in \mathcal{D}$ and $i \in \mathcal{F}$, and obtain a metric c on $\mathcal{D} \cup \mathcal{F}$. Therefore we speak of *metric* service costs if the above condition is satisfied. We make this assumption in the following sections, and will occasionally work with the metric c on $\mathcal{D} \cup \mathcal{F}$. In many practical problems service costs are proportional to geometric distances, or to travel times, and hence are metric.

Jain et al. [2003] showed that the performance guarantee of the above greedy algorithm is $\Omega(\log n / \log \log n)$ even for metric instances, where $n = |\mathcal{D}|$. Indeed, before the paper of Shmoys, Tardos and Aardal [1997] no constant-factor approximation algorithm was known even for metric service costs. Since then, this has changed dramatically. The following sections show different techniques for obtaining constant-factor approximations for the Uncapacitated Facility Location Problem with metric service costs.

An even more restricted problem is the special case when facilities and customers are points in the plane and service costs are the geometric distances. Here Arora, Raghavan and Rao [1998] showed that the problem has an approximation scheme, i.e. a k-factor approximation algorithm for any $k > 1$. This was improved by Kolliopoulos and Rao [1999], but their algorithm seems to be still too slow for practical purposes.

For general metric service costs, the approximation guarantee that can be achieved is known to be between 1.46 (a result due to Guha und Khuller [1999] and Sviridenko [unpublished]; see Section 4.4) and 1.52 (Mahdian, Ye und Zhang [2002]; see Sections 4.2 and 4.3).

In the rest of this paper we assume metric service costs.

3.3 Notation

When we work with an instance of the Uncapacitated Facility Location Problem, we assume the notation $\mathcal{D}, \mathcal{F}, f_i, c_{ij}$ as above. For a given instance and a given nonempty subset X of facilities, a best assignment $\sigma : \mathcal{D} \to X$ satisfying $c_{\sigma(j)j} = \min_{i \in X} c_{ij}$ can be computed easily. Therefore we will often call a nonempty set $X \subseteq \mathcal{F}$ a feasible solution, with facility cost $c_F(X) := \sum_{i \in X} f_i$ and service cost $c_S(X) := \sum_{j \in \mathcal{D}} \min_{i \in X} c_{ij}$. The task is to find a nonempty subset $X \subseteq \mathcal{F}$ such that $c_F(X) + c_S(X)$ is minimum. We denote the optimum by OPT.

3.4 Linear Programming

The Uncapacitated Facility Location Problem can be formulated as an integer linear program as follows:

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{D}} c_{ij}x_{ij} \\ \text{subject to} & \end{array}$$

$$\begin{array}{rcll} x_{ij} & \leq & y_i & (i\in\mathcal{F}, j\in\mathcal{D}) \\ \sum_{i\in\mathcal{F}} x_{ij} & = & 1 & (j\in\mathcal{D}) \\ x_{ij} & \in & \{0,1\} & (i\in\mathcal{F}, j\in\mathcal{D}) \\ y_i & \in & \{0,1\} & (i\in\mathcal{F}) \end{array}$$

By relaxing the integrality constraints we get the linear program:

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{D}} c_{ij}x_{ij} \\ \text{subject to} & \end{array}$$

$$\begin{array}{rcll} x_{ij} & \leq & y_i & (i\in\mathcal{F}, j\in\mathcal{D}) \\ \sum_{i\in\mathcal{F}} x_{ij} & = & 1 & (j\in\mathcal{D}) \\ x_{ij} & \geq & 0 & (i\in\mathcal{F}, j\in\mathcal{D}) \\ y_i & \geq & 0 & (i\in\mathcal{F}) \end{array} \tag{5}$$

This was first formulated by Balinski [1965]. The dual of this LP is:

$$\begin{array}{ll} \text{maximize} & \sum_{j\in\mathcal{D}} v_j \\ \text{subject to} & \end{array}$$

$$\begin{array}{rcll} v_j - w_{ij} & \leq & c_{ij} & (i\in\mathcal{F}, j\in\mathcal{D}) \\ \sum_{j\in\mathcal{D}} w_{ij} & \leq & f_i & (i\in\mathcal{F}) \\ w_{ij} & \geq & 0 & (i\in\mathcal{F}, j\in\mathcal{D}) \end{array} \tag{6}$$

We will need some basic facts from LP theory. First, linear programs can be solved in polynomial time. Second, the primal and dual LP have the same optimum value. Hence for every feasible dual solution (v, w), $\sum_{j\in\mathcal{D}} v_j \le \mathrm{OPT}$. Moreover, primal and dual feasible solutions (x, y) and (v, w) are both optimum if and only if they satisfy the complementary slackness conditions: $x_{ij} > 0$ implies $v_i - w_{ij} = c_{ij}$, $y_i > 0$ implies $\sum_{j\in\mathcal{D}} w_{ij} = f_i$, and $w_{ij} > 0$ implies $x_{ij} = y_i$.

In approximation algorithms, one often has approximate complementary slackness: For example, if (x, y) is an integral feasible primal solution, and (v, w) is a feasible dual solution such that $y_i > 0$ implies $f_i \le 3\sum_{j\in\mathcal{D}} w_{ij}$ and $x_{ij} > 0$ implies $c_{ij} \le 3(v_j - w_{ij})$, then the cost of the solution (x, y) is $\sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{D}} c_{ij}x_{ij} \le 3\sum_{j\in\mathcal{D}} v_j \le 3\,\mathrm{OPT}$.

We close this section by mentioning a different integer programming formulation. Here we have a 0/1-variable z_S for each star $S \in \mathcal{S} := \mathcal{F} \times 2^{\mathcal{D}}$. The cost of a star $S = (i, D)$ is $c(S) = f_i + \sum_{j\in D} c_{ij}$. Then the Uncapacitated Facility Location Problem can be formulated equivalently as:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{S\in\mathcal{S}} c(S) z_S \\ \text{subject to} & \\ & \displaystyle\sum_{S=(i,D)\in\mathcal{S}:j\in D} z_S \ \ge\ 1 \qquad (j \in \mathcal{D}) \\ & \qquad\qquad\qquad z_S \ \in\ \{0,1\} \qquad (S \in \mathcal{S}) \end{array}$$

As above, we relax integrality conditions:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{S\in\mathcal{S}} c(S) z_S \\ \text{subject to} & \\ & \displaystyle\sum_{S=(i,D)\in\mathcal{S}:j\in D} z_S \ \ge\ 1 \qquad (j \in \mathcal{D}) \\ & \qquad\qquad\qquad z_S \ \ge\ 0 \qquad (S \in \mathcal{S}) \end{array} \tag{7}$$

The dual of (7) is:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{j\in\mathcal{D}} v_j \\ \text{subject to} & \\ & \displaystyle\sum_{j\in D} v_j \ \le\ c(S) \qquad (S = (i, D) \in \mathcal{S}) \\ & \qquad v_j \ \ge\ 0 \qquad (j \in \mathcal{D}) \end{array} \tag{8}$$

Any solution z to (7) implies a solution (x, y) to (5) with the same value, by setting $y_i := \sum_{S=(i,D)} z_S$ and $x_{ij} := \sum_{S=(i,D), j\in D} z_S$. Conversely, any solution (x, y) to (5) implies a solution z to (7) with the same value by setting $z_S := x_{ik}$ for $i \in \mathcal{F}$, $k \in \mathcal{D}$ and $S = (i, \{j \in D : x_{ij} \leq x_{ik}\})$. Thus (7) can be solved in polynomial time despite its exponential number of variables.

The same holds for the dual LPs. For any feasible solution (v, w) to (6), v is a feasible solution to (8). Conversely, for any feasible solution v to (8) we can define $w_{ij} := \max\{0, v_j - c_{ij}\}$ in order to obtain a feasible solution to (6): note that for $i \in \mathcal{F}$ and $D := \{j \in \mathcal{D} : v_j > c_{ij}\}$ we have

$$\sum_{j\in\mathcal{D}} w_{ij} = \sum_{j\in\mathcal{D}} \max\{0, v_j - c_{ij}\} = \sum_{j\in D} v_j - \sum_{j\in D} c_{ij} \leq c(i, D) - \sum_{j\in D} c_{ij} = f_i.$$

Hence the two LP relaxations can be considered equivalent.

3.5 Rounding the LP solution

LP rounding algorithms work with integer programming formulations, solve the LP relaxation, and round the resulting fractional solution. However, straightforward rounding does not work for facility location problems. Nevertheless Shmoys, Tardos and Aardal [1997] obtained the first constant-factor approximation by this technique. We now present their approach.

We first compute an optimum solution (x^*, y^*) to (5), and also an optimum solution (v^*, w^*) to the dual (6). We shall produce an integral solution whose cost is at most four times the cost of an optimum fractional solution.

Let G be the bipartite graph with vertex set $\mathcal{F} \cup \mathcal{D}$ containing an edge $\{i, j\}$ iff $x^*_{ij} > 0$. By complementary slackness, $x^*_{ij} > 0$ implies $v^*_j - w^*_{ij} = c_{ij}$, and thus $c_{ij} \leq v^*_j$.

We assign clients to clusters iteratively as follows. In iteration k, let j_k be a customer $j \in \mathcal{D}$ not assigned yet and with v^*_j smallest. Create a new cluster containing j_k and those vertices of G that have distance 2 from j_k and are not assigned yet. Continue until all clients are assigned to clusters.

For each cluster k we choose a neighbour i_k of j_k with f_{i_k} minimum, open i_k, and assign all clients in this cluster to i_k.

Then the service cost for customer j in cluster k is at most

$$c_{i_k j} \leq c_{ij} + c_{ij_k} + c_{i_k j_k} \leq v^*_j + 2v^*_{j_k} \leq 3v^*_j,$$

where i is a common neighbour of j and j_k.

Finally, the facility cost f_{i_k} can be bounded by

$$f_{i_k} \leq \sum_{i\in\mathcal{F}} x^*_{ij_k} f_i = \sum_{i\in\mathcal{F}:\{i,j_k\}\in E(G)} x^*_{ij_k} f_i \leq \sum_{i\in\mathcal{F}:\{i,j_k\}\in E(G)} y^*_i f_i.$$

As j_k and $j_{k'}$ cannot have a common neighbour for $k \neq k'$, the total facility cost is at most $\sum_{i \in \mathcal{F}} y_i^* f_i$.

Summing up, the total cost is $3\sum_{j \in \mathcal{D}} v_j^* + \sum_{i \in \mathcal{F}} y_i^* f_i$, which is at most four times the LP value, and hence at most $4\,\mathrm{OPT}$. We conclude:

Theorem 3.1 *(Shmoys, Tardos and Aardal [1997]) The above LP rounding algorithm is a 4-factor approximation algorithm.* □

Note that the facility cost of the computed solution is at most OPT, while the service cost can be as large as $3\,\mathrm{OPT}$. This raises the question whether the performance guarantee can be improved by opening additional facilities that reduce the service cost. This is indeed true, and we will return to this in Section 4.3.

The rounding itself can be improved quite easily, as shown by Chudak and Shmoys [1998]: First find a solution (x, y) to (5) such that $x_{ij}^* > 0$ implies $x_{ij}^* = y_i^*$. This can be by generating a solution to (7) as shown at the end of Section 3.4, and duplicating facilities when transforming it back to a solution to (5).

Then assign clients to clusters as above. In cluster k, choose a neighbour i_k of j_k, where i is chosen with probability $x_{ij_k}^*$, and open i_k. Finally, for each facility i that does not belong to any cluster, open it with probability y_i^*.

Clearly, the expected total facility cost is $\sum_{i \in \mathcal{F}} y_i^* f_i$. Moreover, for each customer j, the probability that a neighbour is open is at least $1 - \prod_{i \in \mathcal{F}:\{i,j\} \in E(G)}(1 - y_i^*)$. Since $y_i^* \geq x_{ij}^*$ and $\sum_{i \in \mathcal{F}:\{i,j\} \in E(G)} x_{ij}^* = 1$, we have $\prod_{i \in \mathcal{F}:\{i,j\} \in E(G)}(1 - y_i^*) \leq \frac{1}{e}$. Thus, for each $j \in \mathcal{D}$, we have service cost at most v_j^* with probability at least $1 - \frac{1}{e}$, and service cost at most $3v_j^*$ otherwise. Hence the total expected service cost is $\sum_{j \in \mathcal{D}} v_j^*(1 + \frac{2}{e})$, yielding an overall expected cost of $2 + \frac{2}{e}$ times the optimum.

This can be further improved. We will need a well-known inequality (see, e.g., Hardy, Littlewood and Pólya [1964], Theorem 43):

Lemma 3.2 *Let $p_1, \dots, p_n, a_1, \dots, a_n, b_1, \dots, b_n \geq 0$ with $(a_i - a_j)(b_i - b_j) \leq 0$ for all $i, j \in \{1, \dots, n\}$. Then*

$$\left(\sum_{i=1}^n p_i\right)\left(\sum_{i=1}^n p_i a_i b_i\right) \leq \left(\sum_{i=1}^n p_i a_i\right)\left(\sum_{i=1}^n p_i b_i\right).$$

Proof:

$$\begin{aligned}\left(\sum_{i=1}^n p_i\right)\left(\sum_{i=1}^n p_i a_i b_i\right) - \left(\sum_{i=1}^n p_i a_i\right)\left(\sum_{i=1}^n p_i b_i\right)&\\ = \sum_{i=1}^n \sum_{j=1}^n p_i p_j a_j b_j - \sum_{i=1}^n \sum_{j=1}^n p_i a_i p_j b_j&\end{aligned}$$

$$\begin{aligned} &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} p_i p_j (a_i b_i + a_j b_j - a_i b_j - a_j b_i) \\ &\leq 0. \end{aligned}$$ □

We now cluster in a slightly different way: In iteration k, choose j_k among the unassigned customers so that $v^*_{j_k} + \sum_{i\in\mathcal{F}} x^*_{ij_k} c_{ij_k}$ is minimum.

Then the expected service cost for j, even if no neighbour is open, can be bounded by

$$v^*_j + v^*_{j_k} + \sum_{i\in\mathcal{F}} x^*_{ij_k} c_{ij_k} \leq 2v^*_j + \sum_{i\in\mathcal{F}} x^*_{ij} c_{ij}.$$

Let $i_1, \dots, i_l$ be the neighbours of j in G, with $c_{i_1 j} \leq c_{i_2 j} \leq \cdots \leq c_{i_l j}$. Applying the Lemma with $p_k = x^*_{i_k j}$, $a_k = \prod_{h=1}^{k-1}(1 - x^*_{i_h j})$, and $b_k = c_{i_k j}$, the expected service cost for j can be bounded by

$$\begin{aligned} & \sum_{k=1}^{l}\prod_{h=1}^{k-1}(1 - x^*_{i_h j}) x^*_{i_k j} c_{i_k j} + \prod_{h=1}^{l}(1 - x^*_{i_h j}) \left(2v^*_j + \sum_{i\in\mathcal{F}} x^*_{ij} c_{ij}\right) \\ = \; & \sum_{k=1}^{l} a_k p_k b_k + a_l \left(2v^*_j + \sum_{k=1}^{l} p_k b_k\right) \\ \leq \; & \left(\sum_{k=1}^{l} p_k b_k\right)\left(\sum_{k=1}^{l} p_k a_k\right) + a_l \left(2v^*_j + \sum_{k=1}^{l} p_k b_k\right). \end{aligned}$$

Using $\sum_{k=1}^{l} p_k a_k + a_l = 1$ and $\log a_l = \sum_{k=1}^{l} \log(1 - x^*_{i_k j}) \leq \sum_{k=1}^{l}(-x^*_{i_k j}) = -1$ we get that the expected service cost for j is at most

$$\left(\sum_{k=1}^{l} p_k b_k\right) + 2a_l v^*_j \;\; \leq \;\; \sum_{i\in\mathcal{F}} x^*_{ij} c_{ij} + \frac{2}{e} v^*_j.$$

Summing over all customers, and adding the expected total facility cost $\sum_{i\in\mathcal{F}} f_i y^*_i$, the total expected cost is at most $1 + \frac{2}{e} \approx 1.736$ times the optimum. This result is due to Chudak and Shmoys [1998]. They also show how to derandomize the algorithm and obtain a deterministic algorithm with the same performance guarantee.

Sviridenko [2002] further refined the algorithm and improved the performance guarantee to 1.582.

Meanwhile, better performance guarantees have been obtained with simpler and faster algorithms, which do not need a linear programming algorithm. These will be presented in the next section.

4 Greedy and Primal-Dual Algorithms

Although a straighforward greedy algorithm does not produce good results, constant-factor approximations can be achieved by finding approximate primal and dual solutions simultaneously. The algorithms in this section are much faster than LP rounding algorithms, because they do not use a linear programming algorithm as subroutine.

4.1 Two Primal-Dual Algorithms

The first primal-dual approimxation algorithm for the metric Uncapacitated Facility Location Problem was due to Jain and Vazirani [2001]. It computes feasible primal and dual solutions (to the LPs presented in Section 3.4) simultaneously. The primal solution is integral, and the approximation guarantee will follow from approximate complementary slackness conditions.

Let $t = 0$ (t is interpreted as time), and set all dual variables to zero. With proceeding time t, all v_j are increased simultaneously, i.e. they are all equal to t until they are frozen and then remain constant until the end. (v_j can be viewed as the price that customer j pays for being served.)

There are three types of events:

- $v_j = c_{ij}$ for some i and j, where i is not tentatively open.

  Then start to increase w_{ij} at the same rate, maintaining $v_j - w_{ij} = c_{ij}$.

  (w_{ij} can be regarded as the amount that j offers to contribute to the opening cost of facility i. At any stage $w_{ij} = \max\{0, v_j - c_{ij}\}$.)

- $\sum_{j\in\mathcal{D}} w_{ij} = f_i$ for some i.

  Then tentatively open i. For all unconnected customers $j \in \mathcal{D}$ with $v_j \geq c_{ij}$: connect j to i, and freeze v_j and all $w_{i'j}$ for all $i' \in \mathcal{F}$.

- $v_j = c_{ij}$ for some i and j, where i is tentatively open.

  Then connect j to i and freeze v_j.

Several events can occur at the same time and are then processed in arbitrary order. This continues until all customers are connected.

Now let V be the set of facilities that are tentatively open, and let E be the set of pairs $\{i, i'\}$ of distinct tentatively open facilities such that there is a customer j with $w_{ij} > 0$ and $w_{i'j} > 0$. Choose a maximal stable set X in the graph (V, E). Open the facilities in X. For each customer j that is connected to a facility $i \notin X$, connect j to an open neighbour of i in (V, E).

Theorem 4.1 *(Jain and Vazirani [2001]) The above primal-dual algorithm opens a set X of facilities with $3c_F(X) + c_S(X) \leq 3\,\mathrm{OPT}$. In particular, the above is* *a 3-factor approximation algorithm. It can be implemented to run in $O(m \log m)$ time, where $m = |\mathcal{F}||\mathcal{D}|$.*

Proof: (v, w) constitutes a feasible dual solution, thus $\sum_{j \in \mathcal{D}} v_j \leq \mathrm{OPT}$. For each open facility i, all customers j with $w_{ij} > 0$ are connected to i, and $f_i = \sum_{j \in \mathcal{D}} w_{ij}$. Moreover, we claim that the assignment cost for each customer j is at most $3(v_j - w_{i^*j})$, where i^* the facility that j is assigned to.

We distinguish two cases. If $c_{i^*j} = v_j - w_{i^*j}$, this is clear. Otherwise $c_{i^*j} > v_j$ and $w_{i^*j} = 0$. Then there is a (closed) facility i with $c_{ij} = v_j - w_{ij}$ and a customer j' with $w_{ij'} > 0$ and $w_{i^*j'} > 0$, and hence $c_{ij'} = v_{j'} - w_{ij'} < v_{j'}$ and $c_{i^*j'} = v_{j'} - w_{i^*j'} < v_{j'}$. Note that $v_{j'} \leq v_j$, because j' is connected to i^* before j. We conclude that $c_{i^*j} \leq c_{i^*j'} + c_{ij'} + c_{ij} \leq 3v_j$.

To obtain the stated running time, we sort all c_{ij} once in advance. Next, let $t_2 = \min\{t_2^i : i \in Y\}$, where

$$t_2^i = \frac{1}{|U_i|}\left(f_i + \sum_{j \in \mathcal{D} \setminus U : v_j > c_{ij}} (c_{ij} - v_j) + \sum_{j \in U_i} c_{ij}\right)$$

and $U_i := \{j \in U : t \geq c_{ij}\}$. We maintain t_2, t_2^i and $|U_i|$ throughout. With this information it is easy to compute the next event. The numbers $|U_i|$, t_2^i, and possibly t_2, are updated when a new customer is connected or when $t = c_{ij}$ for some $j \in \mathcal{D}$. Hence there are $2m$ such updates overall, each of which takes constant time. □

The first statement of the theorem can be used to get a better performance guarantee. Note that the factor 3 guarantee is tight only if service costs dominate. In this case we may benefit from opening additional facilities. We show how Section 4.3.

The primal-dual algorithm itself has been improved by Jain et al. [2003]. They consider only the set U of unconnected customers. Each customer withdraws all offered contributions to facility opening costs once it is connected. On the other hand, facilities are opened once and for all when they are paid for. Let initially be $U := \mathcal{D}$. There are the following events:

- $v_j = c_{ij}$, where $j \in U$ and i is not open. Then start to increase w_{ij} at the same rate, in order to maintain $v_j - w_{ij} = c_{ij}$.
- $\sum_{j \in U} w_{ij} = f_i$. Then open i. For all j with $v_j \geq c_{ij}$: freeze v_j and all $w_{i'j}$ for all i' (including i), and remove j from U.
- $v_j = c_{ij}$, where $j \in U$ and i is open. Then freeze v_j and remove j from U.

Note that this amounts to the greedy algorithm discussed in Section 3.1, with the only difference that facility costs of facilities that have already been chosen in previous steps are set to zero.

Now the total cost of the solution is clearly $\sum_{j\in\mathcal{D}} v_j$. However, (v,w) is not a feasible dual solution anymore. But Jain et al. [2003] showed that $\frac{1}{\gamma}v$, for $\gamma = 1.861$, is a feasible solution to (8). Hence $\frac{1}{\gamma}\sum_{j\in\mathcal{D}} v_j \le \mathrm{OPT}$, and the above algorithm uses at most γ times this cost. We shall not prove this, as this algorithm is outperformed by the one presented in the next section.

4.2 The Jain-Mahdian-Saberi Algorithm

A slightly modified version of the above primal-dual algorithm, due to Jain, Mahdian and Saberi, and published in Jain et al. [2003], has an even better performance guarantee. Indeed, it leads to the best approximation guarantee known today.

The idea is that connected customers can still offer a certain amount to other facilities if they are closer and re-connecting would save service cost. The algorithm proceeds as follows.

Start with $U := \mathcal{D}$ and time $t = 0$. Increase t, maintaining $v_j = t$ for all $j \in U$. Consider the following events:

- $v_j = c_{ij}$, where $j \in U$ and i is not open. Then start to increase w_{ij} at the same rate, in order to maintain $v_j - w_{ij} = c_{ij}$.

- $\sum_{j\in\mathcal{D}} w_{ij} = f_i$. Then open i. For all $j \in \mathcal{D}$ with $w_{ij} > 0$: freeze v_j and set $w_{i'j} := \max\{0, c_{ij} - c_{i'j}\}$ for all $i' \in \mathcal{F}$, and remove j from U.

- $v_j = c_{ij}$, where $j \in U$ and i is open. Then freeze v_j and set $w_{i'j} := \max\{0, c_{ij} - c_{i'j}\}$ for all $i' \in \mathcal{F}$, and remove j from U.

This algorithm can be implemented in $O(|\mathcal{F}|^2|\mathcal{D}|)$ time. Clearly the total cost of the computed solution is $\sum_{j\in\mathcal{D}} v_j$. We will find a number γ such that $\sum_{j\in D} v_j \le \gamma c(S)$ for each star $S = (i, D)$ (in other words: $\frac{1}{\gamma}v$ is a feasible solution to (8)). This will imply the performance ratio γ.

Consider a star $S = (i, D)$, with $|D| = d$. Renumber the customers in D in the order in which they are connected in the algorithm; w.l.o.g. $D = \{1, \dots, d\}$. We have $v_1 \le v_2 \le \cdots \le v_d$.

Let $k \in D$. Note that k is connected at time $t = v_k$ in the algorithm, and consider the time $t = v_k - \epsilon$ for sufficiently small positive ϵ. For $j = 1, \dots, k-1$ let

$$r_{j,k} := \begin{cases} c_{i(j,k)j} & \text{if } j \text{ is connected to } i(j,k) \in \mathcal{F} \text{ at time } t \\ v_k & \text{otherwise, i.e. if } v_j = v_k \end{cases}.$$

We now write down valid inequalities for these variables. First, for $j = 1, \dots, d$,

$$r_{j,j+1} \ge r_{j,j+2} \ge \cdots \ge r_{j,d} \tag{9}$$

because the service cost decreases if customers are re-connected. Next, for $k = 1, \ldots, d$,

$$\sum_{j=1}^{k-1} \max\{0, r_{j,k} - c_{ij}\} + \sum_{l=k}^{d} \max\{0, v_k - c_{il}\} \leq f_i, \tag{10}$$

as $\sum_{j \in \mathcal{D}} w_{ij}$ at time $t = v_k - \epsilon$ converges to the left-hand side if $\epsilon \to 0$. Finally, for $1 \leq j < k \leq d$,

$$v_k \leq r_{j,k} + c_{ij} + c_{ik}, \tag{11}$$

which is trivial if $r_{j,k} = v_k$, and otherwise follows from observing that the right-hand side is at most $c_{i(j,k)k}$ due to metric service costs, and facility $i(j,k)$ is open at time t.

To prove a performance ratio, we consider the following optimization problem for $\gamma_F \geq 1$ and $d \in \mathbb{N}$:

$$\text{maximize} \quad \frac{\sum_{j=1}^{d} v_j - \gamma_F f_i}{\sum_{j=1}^{d} c_{ij}}$$

subject to

$$\begin{array}{rcll}
v_j & \leq & v_{j+1} & (1 \leq j < d) \\
r_{j,k} & \geq & r_{j,k+1} & (1 \leq j < k < d) \\
r_{j,k} + c_{ij} + c_{ik} & \geq & v_k & (1 \leq j < k \leq d) \\
\displaystyle\sum_{j=1}^{k-1} \max\{r_{j,k} - c_{ij}, 0\} + \sum_{l=k}^{d} \max\{v_k - c_{il}, 0\} & \leq & f_i & (1 \leq k \leq d) \\
\sum_{j=1}^{d} c_{ij} & > & 0 & \\
v_j, c_{ij}, f_i, r_{j,k} & \geq & 0 & (1 \leq j \leq k \leq d)
\end{array}$$

Note that this optimization problem can be easily reformulated as a linear program; it is often referred to as the *factor-revealing LP*. Its optimum values imply performance guarantees for the Jain-Mahdian-Saberi algorithm:

Theorem 4.2 *Let $\gamma_F \geq 1$, and let γ_S be the supremum of the optimum values of the factor-revealing LP over all $d \in \mathbb{N}$. Let an instance be given, and let $X^* \subseteq \mathcal{F}$ be any solution. Then the cost of the solution produced by the algorithm by Jain et al. on this instance is at most $\gamma_F c_F(X^*) + \gamma_S c_S(X^*)$.*

Proof: The algorithm produces numbers v_j and, implicitly, $r_{j,k}$ for all $j, k \in \mathcal{D}$ with $v_j \leq v_k$. For each star (i, D), the numbers $f_i, c_{ij}, v_j, r_{j,k}$ satisfy the conditions (9), (10) and (11) and thus constitute a feasible solution of the above optimization problem unless $\sum_{j=1}^{d} c_{ij} = 0$. Hence $\sum_{j=1}^{d} v_j - \gamma_F f_i \le \gamma_S \sum_{j=1}^{d} c_{ij}$. Choosing $\sigma^* : \mathcal{D} \to X^*$ such that $c_{\sigma^*(j)j} = \min_{i \in X^*} c_{ij}$, and summing over all pairs $(i, \{j \in \mathcal{D} : \sigma^*(j) = i\})$ $(i \in X^*)$, we get

$$\sum_{j \in \mathcal{D}} v_j \le \gamma_F \sum_{i \in X^*} f_i + \gamma_S \sum_{j \in \mathcal{D}} c_{\sigma^*(j)j} = \gamma_F c_F(X^*) + \gamma_S c_S(X^*).$$

As the solution computed by the algorithm has total cost at most $\sum_{j \in \mathcal{D}} v_j$, this proves the theorem. □

To apply this, we observe:

Lemma 4.3 *Consider the above factor-revealing LP for some $d \in \mathbb{N}$.*

(a) *For $\gamma_F = 1$, the optimum is at most 2.*

(b) *(Jain et al. [2003]) For $\gamma_F = 1.61$, the optimum is at most* 1.61.

(c) *(Mahdian, Ye and Zhang [2002]) For $\gamma_F = 1.11$, the optimum is at most* 1.78.

Proof: Here we only prove (a). For a feasible solution we have

$$\begin{aligned} d\left(f_i + \sum_{j=1}^{d} c_{ij}\right) &\ge \sum_{k=1}^{d} \left(\sum_{j=1}^{k-1} r_{j,k} + \sum_{l=k}^{d} v_k \right) \\ &\ge \sum_{k=1}^{d} d v_k - (d-1) \sum_{j=1}^{d} c_{ij}, \end{aligned} \tag{12}$$

implying that $d \sum_{j=1}^{d} v_j \le d f_i + (2d-1) \sum_{j=1}^{d} c_{ij}$, i.e. $\sum_{j=1}^{d} v_j \le f_i + 2 \sum_{j=1}^{d} c_{ij}$. □

The proofs of (b) and (c) are quite long and technical. (a) directly implies that $\frac{1}{2}v$ is a feasible dual solution, and the Jain-Mahdian-Saberi algorithm is a 2-approximation. (b) implies a performance ratio of 1.61. Even better results can be obtained by combining the Jain-Mahdian-Saberi algorithm with scaling and greedy augmentation. This will be shown in the next section. For later use we summarize what follows from Theorem 4.2 and Lemma 4.3:

Corollary 4.4 *Let $(\gamma_F, \gamma_S) \in \{(1,2), (1.61, 1.61), (1.11, 1.78)\}$. Let an instance of the metric* Uncapacitated Facility Location Problem *be given, and let $X^* \subseteq \mathcal{F}$ be any solution. Then the cost of the solution produced by the Jain-Mahdian-Saberi algorithm on this instance is at most $\gamma_F c_F(X^*) + \gamma_S c_S(X^*)$.* □

4.3 Scaling and Greedy Augmentation

Many of the previous results are asymmetric in terms of facility cost and service cost. Often the service cost is higher and could be reduced by opening additional facilities. Indeed, this can be exploited to improve several performance guarantees.

For a solution $X \subseteq \mathcal{F}$ and a facility $i \in \mathcal{F}$, we denote by $g_X(i) := c_S(X) - c_S(X \cup \{i\})$. Then we have:

Proposition 4.5 *Let $\emptyset \neq X, X^* \subseteq \mathcal{F}$. Then $\sum_{i \in X^*} g_X(i) \geq c_S(X) - c_S(X^*)$.*

Proof: For $j \in \mathcal{D}$ let $\sigma(j) \in X$ such that $c_{\sigma(j)j} = \min_{i \in X} c_{ij}$, and let $\sigma^*(j) \in X^*$ such that $c_{\sigma^*(j)j} = \min_{i \in X^*} c_{ij}$. Then $g_X(i) \geq \sum_{j \in \mathcal{D}: \sigma^*(j)=i} (c_{\sigma(j)j} - c_{ij})$ for all $i \in X^*$. Summation yields the lemma. □

In particular, there exists an $i \in X^*$ with $\frac{g_X(i)}{f_i} \geq \frac{c_S(X)-c_S(X^*)}{c_F(X^*)}$. By *greedy augmentation* of a set X we mean iteratively picking an element $i \in \mathcal{F}$ maximizing $\frac{g_X(i)}{f_i}$ until $g_X(i) \leq f_i$ for all $i \in \mathcal{F}$. We need the following lemma:

Lemma 4.6 *(Charikar and Guha [1999]) Let $\emptyset \neq X, X^* \subseteq \mathcal{F}$. Apply greedy augmentation to X, obtaining a set $Y \supseteq X$. Then*

$$c_F(Y) + c_S(Y) \leq \\ c_F(X) + c_F(X^*) \ln\left(\max\left\{1, \frac{c_S(X) - c_S(X^*)}{c_F(X^*)}\right\}\right) + c_F(X^*) + c_S(X^*).$$

Proof: If $c_S(X) \leq c_F(X^*) + c_S(X^*)$, the above inequality evidently holds even before augmentation ($Y = X$). Otherwise, let $X = X_0, X_1, \ldots, X_k$ be the sequence of augmented sets, such that k is the first index for which $c_S(X_k) \leq c_F(X^*) + c_S(X^*)$. By renumbering facilities we may assume $X_i \setminus X_{i-1} = \{i\}$ $(i = 1, \ldots, k)$. By Proposition 4.5,

$$\frac{c_S(X_{i-1}) - c_S(X_i)}{f_i} \geq \frac{c_S(X_{i-1}) - c_S(X^*)}{c_F(X^*)}$$

for $i = 1, \ldots, k$. Hence $f_i \leq c_F(X^*) \frac{c_S(X_{i-1})-c_S(X_i)}{c_S(X_{i-1})-c_S(X^*)}$ (note that $c_S(X_{i-1}) > c_S(X^*)$), and

$$c_F(X_k) + c_S(X_k) \quad \leq \quad c_F(X) + c_F(X^*) \sum_{i=1}^{k} \frac{c_S(X_{i-1}) - c_S(X_i)}{c_S(X_{i-1}) - c_S(X^*)} + c_S(X_k).$$

As the right-hand side increases with increasing $c_S(X_k)$ (the derivative is $1 - \frac{c_F(X^*)}{c_S(X_{k-1})-c_S(X^*)} > 0$), we may assume $c_S(X_k) = c_F(X^*) + c_S(X^*)$. By using

$x - 1 \geq \ln x$ for $x > 0$, we get

$$\begin{aligned} c_F(X_k) + c_S(X_k) &\leq c_F(X) + c_F(X^*) \sum_{i=1}^{k} \left(1 - \frac{c_S(X_i) - c_S(X^*)}{c_S(X_{i-1}) - c_S(X^*)} \right) + c_S(X_k) \\ &\leq c_F(X) - c_F(X^*) \sum_{i=1}^{k} \ln \frac{c_S(X_i) - c_S(X^*)}{c_S(X_{i-1}) - c_S(X^*)} + c_S(X_k) \\ &= c_F(X) - c_F(X^*) \ln \frac{c_S(X_k) - c_S(X^*)}{c_S(X) - c_S(X^*)} + c_S(X_k) \\ &= c_F(X) + c_F(X^*) \ln \frac{c_S(X) - c_S(X^*)}{c_F(X^*)} + c_F(X^*) + c_S(X^*). \end{aligned}$$

□

This can be used to improve several of the previous performance guarantees. Sometimes it is good to combine greedy augmentation with scaling. We get the following general result:

Theorem 4.7 *Suppose there are positive constants $\beta, \gamma_S, \gamma_F$ and an algorithm A which, for every instance, computes a solution X such that $\beta c_F(X) + c_S(X) \leq \gamma_F c_F(X^*) + \gamma_S c_S(X^*)$ for each $\emptyset \neq X^* \subseteq \mathcal{F}$. Let $\delta \geq \frac{1}{\beta}$.*

Then, scaling facilities by δ, applying A to the modified instance, scaling back, and applying greedy augmentation to the original instance yields a solution of cost at most $\max\{\frac{\gamma_F}{\beta} + \ln(\beta\delta), 1 + \frac{\gamma_S - 1}{\beta\delta}\}$ times the optimum.

Proof: Let X^* be the set of facilities of an optimum solution to the original instance. We have $\beta\delta c_F(X) + c_S(X) \leq \gamma_F \delta c_F(X^*) + \gamma_S c_S(X^*)$. If $c_S(X) \leq c_S(X^*) + c_F(X^*)$, then we have $\beta\delta(c_F(X) + c_S(X)) \leq \gamma_F \delta c_F(X^*) + \gamma_S c_S(X^*) + (\beta\delta - 1)(c_S(X^*) + c_F(X^*))$, so X is a solution that costs at most $\max\{1 + \frac{\gamma_F \delta - 1}{\beta\delta}, 1 + \frac{\gamma_S - 1}{\beta\delta}\}$ times the optimum. Note that $1 + \frac{\gamma_F \delta - 1}{\beta\delta} \leq \frac{\gamma_F}{\beta} + \ln(\beta\delta)$ as $1 - \frac{1}{x} \leq \ln x$ for all $x > 0$.

Otherwise we apply greedy augmentation to X and get a solution of cost at most

$$\begin{aligned} & c_F(X) + c_F(X^*) \ln \frac{c_S(X) - c_S(X^*)}{c_F(X^*)} + c_F(X^*) + c_S(X^*) \\ \leq\ & c_F(X) + c_F(X^*) \ln \frac{(\gamma_S - 1) c_S(X^*) + \gamma_F \delta c_F(X^*) - \beta\delta c_F(X)}{c_F(X^*)} + c_F(X^*) + c_S(X^*). \end{aligned}$$

The derivative of this expression with respect to $c_F(X)$ is

$$1 - \frac{\beta\delta c_F(X^*)}{(\gamma_S - 1) c_S(X^*) + \gamma_F \delta c_F(X^*) - \beta\delta c_F(X)} ,$$

which is zero for $c_F(X) = \frac{\gamma_F - \beta}{\beta} c_F(X^*) + \frac{\gamma_S - 1}{\beta\delta} c_S(X^*)$. Hence we get a solution of cost at most

$$\left(\frac{\gamma_F}{\beta} + \ln(\beta\delta)\right) c_F(X^*) + \left(1 + \frac{\gamma_S - 1}{\beta\delta}\right) c_S(X^*). \qquad \Box$$

With Corollary 4.4 we can apply this result to the Jain-Mahdian-Saberi algorithm with $\beta = \gamma_F = 1$ and $\gamma_S = 2$: by setting $\delta = 1.76$ we obtain an approximation guarantee of 1.57. With $\beta = 1$, $\gamma_F = 1.11$ and $\gamma_S = 1.78$ (cf. Corollary 4.4) we can do even better:

Corollary 4.8 *(Mahdian, Ye and Zhang [2002]) Multiply all facility costs by* $\delta = 1.504$*, apply the Jain-Mahdian-Saberi algorithm, scale back the facility costs, and apply greedy augmentation. Then this algorithm has an approximation guarantee of* 1.52. □

This is the best performance ratio that is currently known for the metric Uncapacitated Facility Location Problem.

Theorem 4.7 can also be used to improve the performance ratio of other algorithms. Applied to the Jain-Vazirani algorithm, where we have $\beta = \gamma_S = \gamma_F = 3$, we can set $\delta = 0.782$ and obtain a performance ratio of 1.853 (instead of 3). For the LP rounding algorithm by Shmoys, Tardos and Aardal (Section 3.5), the prerequisites hold for any β and $\gamma_F = \gamma_S = 3 + \beta$. By choosing β and δ appropriately (e.g. $\beta = 6$, $\delta = 0.7$), the performance ratio can be reduced from 4 to below 3.

In some cases, it suffices to scale facility costs before applying the algorithm:

Proposition 4.9 *Suppose there are constants* $\alpha_F, \alpha_S, \beta_F, \beta_S$ *and an algorithm* A*, such that for every instance* A *computes a solution* X *such that* $c_F(X) \le \alpha_F c_F(X^*) + \alpha_S c_S(X^*)$ *and* $c_S(X) \le \beta_F c_F(X^*) + \beta_S c_S(X^*)$ *for each* $\emptyset \ne X^* \subseteq \mathcal{F}$.

Let $\delta := \frac{\beta_S - \alpha_F + \sqrt{(\beta_S - \alpha_F)^2 + 4\beta_F \alpha_S}}{2\beta_F} = \frac{2\alpha_S}{\alpha_F - \beta_S + \sqrt{(\beta_S - \alpha_F)^2 + 4\beta_F\alpha_S}}$. *Then, multiplying all facility costs by* δ *and applying* A *to the modified instance yields a solution of cost at most* $\frac{\beta_S + \alpha_F + \sqrt{(\beta_S - \alpha_F)^2 + 4\beta_F\alpha_S}}{2}$ *times the optimum.*

Proof: Let be X^* the set of facilities of an optimum solution to the original instance. We have $c_F(X) + c_S(X) \le \frac{1}{\delta}(\alpha_F \delta c_F(X^*) + \alpha_S c_S(X^*)) + \beta_F \delta c_F(X^*) + \beta_S c_S(X^*)) = (\alpha_F + \delta\beta_F) c_F(X^*) + (\frac{1}{\delta}\alpha_S + \beta_S) c_S(X^*)$. □

4.4 Lower Bound on Approximation Guarantees

In this section we denote by α the solution of the equation $\alpha + 1 = \ln\frac{2}{\alpha}$; we have $0.463 \le \alpha \le 0.4631$. A simple calculation shows that $\alpha = \frac{\alpha}{\alpha+1}\ln\frac{2}{\alpha} = \max\{\frac{\xi}{\xi+1}\ln\frac{2}{\xi} : \xi > 0\}$.

We will show that $1+\alpha$ is the approximation ratio that can be achieved for distances 1 and 3. More precisely, we have the following two results, both by Guha and Khuller [1999]:

Theorem 4.10 *Consider the* Uncapacitated Facility Location Problem *restricted to instances where all service costs are within the interval* $[1,3]$. *This problem has an* $(1+\alpha+\epsilon)$*-factor approximation algorithm for every* $\epsilon > 0$.

Proof: Let $\epsilon > 0$, and let $k := \lceil \frac{1}{\epsilon} \rceil$. Enumerate all solutions $X \subseteq \mathcal{F}$ with $|X| \le k$.

We compute another solution as follows. We first open one facility i with minimum opening cost f_i, and then apply greedy augmentation to obtain a solution Y. We claim that the best solution costs at most $1+\alpha+\epsilon$ times the optimum.

Let X^* be an optimum solution and $\xi = \frac{c_F(X^*)}{c_S(X^*)}$. We may assume that $|X^*| > k$, as otherwise we have found X^* above. Then $c_F(\{i\}) \le \frac{1}{k} c_F(X^*)$. Moreover, as the service costs are between 1 and 3, $c_S(\{i\}) \le 3|\mathcal{D}| \le 3c_S(X^*)$.

By Lemma 4.6, the cost of Y is at most

$$\begin{aligned}
& \frac{1}{k}c_F(X^*) + c_F(X^*)\ln\left(\max\left\{1, \frac{2c_S(X^*)}{c_F(X^*)}\right\}\right) + c_F(X^*) + c_S(X^*) \\
= \;& c_S(X^*)\left(\frac{\xi}{k} + \xi \ln\left(\max\left\{1, \frac{2}{\xi}\right\}\right) + \xi + 1\right) \\
\le \;& (1+\alpha+\epsilon)(1+\xi)c_S(X^*) \\
= \;& (1+\alpha+\epsilon)(c_F(X^*) + c_S(X^*)).
\end{aligned}$$

□

The performance guarantee seems to be best possible in view of the following:

Theorem 4.11 *If there is an* $\epsilon > 0$ *and a* $(1+\alpha-\epsilon)$*-factor approximation algorithm for the metric* Uncapacited Facility Location Problem, *then there is an algorithm for the* Minimum Cardinality Set Cover Problem *which for every instance* $(U, \mathcal{S})$ *computes a set cover* $\mathcal{R}$ *with* $|\mathcal{R}| \le (1-\epsilon^2)\ln|U|\,\mathrm{OPT}(U,\mathcal{S})$.

By Feige's [1998] result, the latter would imply that every problem in *NP* can be solved in $n^{O(\log\log n)}$ time, where n is the input size.

Proof: Let $(U, \mathcal{S})$ be a set system. If $|U| \le e^{\frac{\alpha+2}{\epsilon}}$, we solve the instance by complete enumeration (in $2^{2^{|U|}}$ time, which is a constant depending on ϵ only).

Otherwise we run the following algorithm for each $t \in \{1, \dots, |U|\}$:

① Set $\mathcal{R} := \emptyset$. Set $\mathcal{F} := \mathcal{S}$, $\mathcal{D} := U$, and let $c_{ij} := 1$ for $j \in i \in \mathcal{F}$ and $c_{ij} := 3$ for $i \in \mathcal{F}$ and $j \in U \setminus \{i\}$.

② Set $f_i := \frac{\alpha|\mathcal{D}|}{t}$.

③ Find a $(1+\alpha-\epsilon)$-approximate solution X to the instance $(\mathcal{F}, \mathcal{D}, c, f)$ of the Uncapacitated Facility Location Problem.

```
④        Set $\mathcal{R} := \mathcal{R} \cup X$ and $\mathcal{D}' := \{j \in \mathcal{D} : c_{ij} = 3 \text{ for all } i \in X\}$.
⑤        If $\mathcal{D}' = \mathcal{D}$, then stop (without output).
⑥        Set $\mathcal{D} := \mathcal{D}'$. If $\mathcal{D} \neq \emptyset$, then go to ②.
```

Note that the service costs defined in ① are metric. Clearly $\mathcal{R}$ covers $U \setminus \mathcal{D}$ at any stage. We claim that if there is a set cover of cardinality t, then the algorithm terminates with $\mathcal{D} = \emptyset$ and a set cover $\mathcal{R}$ with $|\mathcal{R}| \leq t(1-\epsilon^2)\ln|U|$. As we run the algorithm for all possible values of t, this will conclude the proof.

Let us consider the k-th iteration of the algorithm, in which we denote $n_k := |\mathcal{D}|$, $\beta_k := |X|$, and $\gamma_k := \frac{|\mathcal{D}'|}{n_k}$. In ③ of iteration k, the algorithm computes a solution of cost $\beta_k \frac{\alpha n_k}{t} + n_k + 2|\mathcal{D}'|$. Suppose there is a set cover of cardinality t, and thus a solution of cost $t\frac{\alpha n_k}{t} + n_k$ to the facility location instance. Then we have $\beta_k \frac{\alpha n_k}{t} + n_k + 2|\mathcal{D}'| \leq (1+\alpha-\epsilon)(\alpha+1)n_k$, and hence

$$\begin{aligned}
\frac{\beta_k \alpha}{t} + 2\gamma_k &\leq (1+\alpha-\epsilon)(\alpha+1) - 1 \\
&= \alpha^2 + 2\alpha - \epsilon(\alpha+1) \\
&< \alpha^2 + 2\alpha - \epsilon\alpha(\alpha+2) \\
&= (1-\epsilon)\alpha(\alpha+2),
\end{aligned}$$

which implies

$$\frac{\beta_k}{t} < (1-\epsilon)(\alpha+2) \tag{13}$$

and

$$\begin{aligned}
\frac{\beta_k \alpha}{t} + 2\gamma_k &< (1-\epsilon)\alpha\left(1 + \ln\frac{2}{\alpha}\right) \\
&\leq (1-\epsilon)\alpha\left(1 + \ln\frac{2}{(1-\epsilon)\alpha}\right) \\
&\leq x\alpha + 2e^{-\frac{x}{1-\epsilon}}
\end{aligned}$$

for all $x > 0$, as the right-hand side is minimized for $x = (1-\epsilon)\ln\frac{2}{(1-\epsilon)\alpha}$. Setting $x = \frac{\beta_k}{t}$, we get

$$\gamma_k < e^{-\frac{\beta_k}{t(1-\epsilon)}}, \tag{14}$$

and hence, for all $k < l$,

$$\beta_k < t(1-\epsilon)\ln\frac{1}{\gamma_k}. \tag{15}$$

In particular, $\gamma_k < 1$ for all k, and the algorithm indeed terminates with $\mathcal{D} = \emptyset$, say after iteration l. Using (13) and (15), we conclude that the the algorithm computes a set cover of cardinality

$$\sum_{k=1}^{l} \beta_k \leq t(1-\epsilon)\left(\sum_{k=1}^{l-1} \ln\frac{1}{\gamma_k} + \alpha + 2\right)$$

$$\begin{aligned} &= t(1-\epsilon)\left(\ln \prod_{k=1}^{l-1} \frac{1}{\gamma_k} + \alpha + 2\right) \\ &= t(1-\epsilon)(\ln |U| - \ln n_l + \alpha + 2) \\ &\leq t(1-\epsilon)(\ln |U| + \alpha + 2) \\ &\leq t(1-\epsilon^2) \ln |U| \end{aligned}$$

since $|U| \geq e^{\frac{\alpha+2}{\epsilon}}$. □

M. Sviridenko [unpublished] observed that this result can be strengthened using the following result:

Theorem 4.12 *For every fixed $\delta > 0$ and $\zeta > 0$, the following decision problem is NP-hard: An instance consists of a set system $(U, \mathcal{S})$ and a number $t \in \mathbb{N}$, such that either*

(a) *there exists a $\mathcal{R} \subseteq \mathcal{S}$ with $|\mathcal{R}| = t$ and $\bigcup \mathcal{R} = U$, or*

(b) *for each $\mathcal{R} \subseteq \mathcal{S}$ with $|\mathcal{R}| \leq \zeta t$ we have $|\bigcup \mathcal{R}| \leq \left(1 - e^{-\frac{|\mathcal{R}|}{t}} + \delta\right) |U|$.*

The task is to decide whether (a) or (b) holds.

For the case $\zeta = 1$ this was implicitly proved by Feige [1998], implying that the problem that asks for covering as many elements as possible with a given number t of sets in a given set system cannot be approximated with a factor smaller than $1 - \frac{1}{e}$ unless $P = NP$ (cf. Theorem 5.3 in Feige's paper; it is easy to see that the greedy algorithm achieves precisely this factor). Theorem 4.12 can be proved similarly (see also the hints in Section 5 of Feige, Lovász and Tetali [2004]). This implies:

Theorem 4.13 *(Sviridenko [unpublished]) If there is an $\epsilon > 0$ and a $(1+\alpha-\epsilon)$-factor approximation algorithm for the metric* Uncapacitated Facility Location Problem, *then $P = NP$.*

Proof: Suppose that for some $0 < \epsilon < 1$ there is a $(1+\alpha-\epsilon)$-factor approximation algorithm for the metric Uncapacitated Facility Location Problem. Choose $\zeta := \alpha + 3$ and $\delta := e^{-\zeta} - e^{-\frac{\zeta}{1-\epsilon}}$. Let $(U, \mathcal{S}, t)$ be an instance of the decision problem defined in Theorem 4.12 (with parameters δ and ζ).

We run the algorithm in the proof of Theorem 4.11 for the given t, except that we stop when $\mathcal{D} = \emptyset$ or $|\mathcal{R}| \geq t$; say after l iterations. We show that the output $\mathcal{R}$ can be used to decide between cases (a) and (b) in Theorem 4.12. More precisely, assume that case (a) holds. We show that then $|\mathcal{R}| < \zeta t$ and $|\bigcup \mathcal{R}| > \left(1 - e^{-\frac{|\mathcal{R}|}{t}} + \delta\right) |U|$.

Indeed, we get from (13) in the proof of Theorem 4.11 that $|\mathcal{R}| < t + (1-\epsilon)(\alpha+2)t < \zeta t$. Hence we are done if $|\bigcup \mathcal{R}| = |U|$. Otherwise, (14) implies

$$\frac{|\bigcup \mathcal{R}|}{|U|} = 1 - \prod_{k=1}^{l} \gamma_k > 1 - e^{-\frac{|\mathcal{R}|}{t(1-\epsilon)}} > 1 - e^{-\frac{|\mathcal{R}|}{t}} + \delta,$$

as the function $x \mapsto e^{-x} - e^{-\frac{x}{(1-\epsilon)}}$ is monotonically decreasing for $x \geq 1$ (the derivative $\frac{1}{1-\epsilon} e^{-\frac{x}{1-\epsilon}} - e^{-x}$ is negative as $\frac{1}{1-\epsilon} = 1 + \frac{\epsilon}{1-\epsilon} < e^{\frac{\epsilon}{1-\epsilon}} \leq e^{\frac{x\epsilon}{1-\epsilon}} = e^{-x} e^{\frac{x}{1-\epsilon}}$), and $1 \leq \frac{|\mathcal{R}|}{t} < \zeta$.

This means that we have a polynomial-time algorithm for the *NP*-hard decision problem defined in Theorem 4.12. □

5 Reductions for More General Problems

We will now discuss two generalizations of the metric Uncapacitated Facility Location Problem, which we can also solve with primal-dual algorithms discussed above, by two different techniques. Both techniques have been proposed originally by Jain and Vazirani [2001].

5.1 Soft Capacities

In the Soft-Capacitated Facility Location Problem, each facility $i \in \mathcal{F}$ has a *capacity* u_i. If we assign a set $D \subseteq \mathcal{D}$ of customers to i, then we have to open $\lceil \frac{|D|}{u_i} \rceil$ copies of facility i, and consequently pay an opening cost of $\lceil \frac{|D|}{u_i} \rceil f_i$.

Another problem does not allow to open multiple copies of the facilities, i.e. we have hard capacity bounds. This is called the Capacitated Facility Location Problem and will be discussed in Section 7. It can probably not be reduced easily to the Uncapacitated Facility Location Problem.

For soft capacities, however, the reduction is easy. Again we assume metric service costs.

Theorem 5.1 *(Mahdian, Ye and Zhang [2002]) Let γ_F and γ_S be constants and A a polynomial-time algorithm such that, for every instance of the metric* Uncapacitated Facility Location Problem, *A computes a solution X with $c_F(X) + c_S(X) \leq \gamma_F c_F(X^*) + \gamma_S c_S(X^*)$ for each $\emptyset \neq X^* \subseteq \mathcal{F}$. Then there is a $(\gamma_F + \gamma_S)$-factor approximation algorithm for the metric* Soft-Capacitated Facility Location Problem.

Proof: Consider an instance $I = (\mathcal{F}, \mathcal{D}, f, u, c)$ of the metric Soft-Capacitated Facility Location Problem. We transform it to the instance $I' = (\mathcal{F}, \mathcal{D}, f, c')$ of the metric Uncapacitated Facility Location Problem, where $c'_{ij} := c_{ij} + \frac{f_i}{u_i}$ for $i \in \mathcal{F}$ and $j \in \mathcal{D}$. (Note that c' is metric whenever c is metric.)

We apply A to I' and find a solution $X \in \mathcal{F}$ and an assignment $\sigma : \mathcal{D} \to X$. If $\sigma^* : \mathcal{D} \to \mathcal{F}$ is an optimum solution to I, where $X^* := \{i \in \mathcal{F} : \exists j \in \mathcal{D} : \sigma^*(j) = i\}$ is the set of facilities opened at least once,

$$\begin{aligned}
\sum_{i \in X} \left\lceil \frac{|\{j \in \mathcal{D} : \sigma(j) = i\}|}{u_i} \right\rceil f_i + \sum_{j \in \mathcal{D}} c_{\sigma(j)j} \\
&\le \sum_{i \in X} f_i + \sum_{j \in \mathcal{D}} c'_{\sigma(j)j} \\
&\le \gamma_F \sum_{i \in X^*} f_i + \gamma_S \sum_{j \in \mathcal{D}} c'_{\sigma^*(j)j} \\
&\le (\gamma_F + \gamma_S) \sum_{i \in X^*} \left\lceil \frac{|\{j \in \mathcal{D} : \sigma^*(j) = i\}|}{u_i} \right\rceil f_i + \gamma_S \sum_{j \in \mathcal{D}} c_{\sigma^*(j)j}.
\end{aligned}$$

□

In particular, using the Jain-Mahdian-Saberi algorithm we get a 2.89-factor approximation algorithm for the metric Soft-Capacitated Facility Location Problem by Corollary 4.4; here $\gamma_F = 1.11$ and $\gamma_S = 1.78$. In a subsequent paper, Mahdian, Ye and Zhang [2003] observed that the last inequality can be strengthened:

Theorem 5.2 *(Mahdian, Ye and Zhang [2003]) There is a 2-factor approximation algorithm for the metric* Soft-Capacitated Facility Location Problem.

Proof: Indeed, here we have $c'_{ij} = c_{ij} + \frac{f_i}{u_i}$, and in the analysis of the Jain-Mahdian-Saberi algorithm we get

$$v_k \le r_{j,k} + c_{ij} + c_{ik}$$

instead of (11) (with the original service costs c), and thus the analysis in (12) yields

$$d \sum_{j=1}^{d} v_j \le d f_i + d \sum_{j=1}^{d} c'_{ij} + (d-1) \sum_{j=1}^{d} c_{ij}.$$

This implies that the cost of the solution is at most

$$\sum_{i \in X^*} f_i + \sum_{j \in \mathcal{D}} c'_{\sigma^*(j)j} + \sum_{j \in \mathcal{D}} c_{\sigma^*(j)j} \le 2 \sum_{i \in X^*} \left\lceil \frac{\sum_{j:\sigma^*(j)=i} d_j}{u_i} \right\rceil f_i + 2 \sum_{j \in \mathcal{D}} c_{\sigma^*(j)j}.$$

□

5.2 Bounding the Number of Facilities

For $k \in \mathbb{N}$, the k-Facility Location Problem is the Uncapacitated Facility Location Problem with the additional constraint that no more than k facilities may be opened. A special case, where facility opening costs are zero, is the well-known k-Median Problem. In this section we describe an approximation algorithm for the metric k-Facility Location Problem.

When we have a problem which becomes much easier if a certain type of constraints is omitted, Lagrangian relaxation is a common technique. In our case, we will add a constant λ to each facility opening cost.

Theorem 5.3 *(Jain and Vazirani [2001]) If there is a constant γ_S and a polynomial-time algorithm A, such that for every instance of the metric* Uncapacitated Facility Location Problem *A computes a solution X such that $c_F(X)+c_S(X) \leq c_F(X^*) + \gamma_S c_S(X^*)$ for each $\emptyset \neq X^* \subseteq \mathcal{F}$, then there is a $(2\gamma_S)$-factor approximation algorithm for the metric* k-Facility Location Problem *with integral data.*

Proof: Let an instance of the metric k-Facility Location Problem be given. We assume that service costs are integers within $\{0, 1, \dots, c_{\max}\}$ and facility opening costs are integers within $\{0, 1, \dots, f_{\max}\}$.

First we check if OPT $= 0$, and find a solution of zero cost if one exists. This is easy; see the proof of Lemma 5.4. Hence we assume OPT ≥ 1. Let X^* be an optimum solution (we will use it for the analysis only).

Let $A(\lambda) \subseteq \mathcal{F}$ be the solution computed by A for the instance where all facility opening costs are increased by λ but the constraint on the number of facilities is omitted. We have $c_F(A(\lambda)) + |A(\lambda)|\lambda + c_S(A(\lambda)) \leq c_F(X^*) + |X^*|\lambda + \gamma_S c_S(X^*)$, and hence

$$c_F(A(\lambda)) + c_S(A(\lambda)) \leq c_F(X^*) + \gamma_S c_S(X^*) + (k - |A(\lambda)|)\lambda \qquad (16)$$

for all $\lambda \geq 0$. If $|A(0)| \leq k$, then $A(0)$ is a feasible solution costing at most γ_S times the optimum, and we are done.

Otherwise $A(0) > k$, and note that $|A(f_{\max} + \gamma_S|\mathcal{D}|c_{\max} + 1)| = 1 \leq k$. Set $\lambda' := 0$ and $\lambda'' := f_{\max} + \gamma_S|\mathcal{D}|c_{\max} + 1$, and apply binary search, maintaining $|A(\lambda'')| \leq k < |A(\lambda')|$. After $O(\log|\mathcal{D}| + \log f_{\max} + \log c_{\max})$ iterations, in each of which we set one of λ', λ'' to their arithmetic mean depending on whether $A(\frac{\lambda'+\lambda''}{2}) \leq k$ or not, we have $\lambda'' - \lambda' \leq \frac{1}{|\mathcal{D}|^2}$. (Note that this binary search works although $\lambda \mapsto |A(\lambda)|$ is in general not monotonic.)

If $|A(\lambda'')| = k$, then (16) implies that $A(\lambda'')$ is a feasible solution costing at most γ_S times the optimum, and we are done. However, we will not always encounter such a λ'', because $\lambda \mapsto |A(\lambda)|$ is not always monotonic and can jump by more than 1 (Archer, Rajagopalan and Shmoys [2003] showed how to fix this by perturbing costs, but were unable to do it in polynomial time).

Thus we consider $X := A(\lambda')$ and $Y := A(\lambda'')$ and assume henceforth $|X| > k > |Y|$. Let $\alpha := \frac{k-|Y|}{|X|-|Y|}$ and $\beta := \frac{|X|-k}{|X|-|Y|}$.

Choose a subset X' of X with $|X'| = |Y|$ such that $\min_{i\in X'} c_{ii'} = \min_{i\in X} c_{ii'}$ for each $i' \in Y$, where we write $c_{ii'} := \min_{j\in\mathcal{D}}(c_{ij} + c_{i'j})$.

We open either all elements of X' (with probability α) or all elements of Y (with probability $\beta = 1-\alpha$). In addition, we open a set of $k - |Y|$ facilities of $X \setminus X'$, chosen uniformly at random. Then the expected facility cost is $\alpha c_F(X) + \beta c_F(Y)$.

Let $j \in \mathcal{D}$, and let i' be a closest facility in X, and let i'' be a closest facility in Y. Connect j to i' if it is open, else to i'' if it is open. If neither i' nor i'' is open, connect j to a facility $i''' \in X'$ minimizing $c_{i''i'''}$.

This yields an expected service cost $\alpha c_{i'j} + \beta c_{i''j}$ if $i' \in X'$ and at most

$$\begin{aligned}
& \alpha c_{i'j} + (1-\alpha)\beta c_{i''j} + (1-\alpha)(1-\beta)c_{i'''j} \\
\leq\ & \alpha c_{i'j} + \beta^2 c_{i''j} + \alpha\beta\left(c_{i''j} + \min_{j'\in\mathcal{D}}(c_{i''j'} + c_{i'''j'})\right) \\
\leq\ & \alpha c_{i'j} + \beta^2 c_{i''j} + \alpha\beta(c_{i''j} + c_{i''j} + c_{i'j}) \\
=\ & \alpha(1+\beta)c_{i'j} + \beta(1+\alpha)c_{i''j}
\end{aligned}$$

if $i' \in X \setminus X'$.

Thus the total expected service cost is at most

$$(1+\max\{\alpha,\beta\})(\alpha c_S(X) + \beta c_S(Y)) \leq \left(2 - \frac{1}{|\mathcal{D}|}\right)(\alpha c_S(X) + \beta c_S(Y)).$$

Overall, using (16), we get an expected cost of at most

$$\begin{aligned}
& \left(2 - \frac{1}{|\mathcal{D}|}\right)(\alpha(c_F(X) + c_S(X)) + \beta(c_F(Y) + c_S(Y))) \\
\leq\ & \left(2 - \frac{1}{|\mathcal{D}|}\right)\left(c_F(X^*) + \gamma_S c_S(X^*) + (\lambda'' - \lambda')\frac{(|X|-k)(k-|Y|)}{|X|-|Y|}\right) \\
\leq\ & \left(2 - \frac{1}{|\mathcal{D}|}\right)\left(c_F(X^*) + \gamma_S c_S(X^*) + (\lambda'' - \lambda')\frac{|X|-|Y|}{4}\right) \\
\leq\ & \left(2 - \frac{1}{|\mathcal{D}|}\right)\left(c_F(X^*) + \gamma_S c_S(X^*) + \frac{1}{4|\mathcal{D}|}\right) \\
\leq\ & \left(2 - \frac{1}{|\mathcal{D}|}\right)\left(1 + \frac{1}{4|\mathcal{D}|}\right)(c_F(X^*) + \gamma_S c_S(X^*)) \\
\leq\ & \left(2 - \frac{1}{2|\mathcal{D}|}\right)(c_F(X^*) + \gamma_S c_S(X^*))
\end{aligned}$$

and thus at most $2\gamma_S(c_F(X^*) + c_S(X^*))$.

Note that the expected cost is easy to compute even under the condition that a subset Z is opened with probability 1 and randomly chosen $k - |Z|$ facilities of some other set are opened. Hence one can derandomize this algorithm by the method of conditional probabilities: first open X' or Y depending on where the bound on the expected cost is at most $(2-\frac{1}{|\mathcal{D}|})(\alpha(c_F(X)+c_S(X))+\beta(c_F(Y)+c_S(Y)))$, and then successively open a facility of $X \setminus X'$ such that this bound continues to hold. □

In particular, by the above Jain-Mahdian-Saberi algorithm (Corollary 4.4), we obtain a 4-factor approximation algorithm for the metric k-Facility Location Problem. The first constant-factor approximation algorithm for the metric k-Facility Location Problem was due to Charikar et al. [2002].

The running time of the binary search is weakly polynomial and works for integral data only. However we can make it strongly polynomial by discretizing the input data:

Lemma 5.4 *For any instance I of the metric k-Facility Location Problem, $\gamma_{\max} \geq 1$ and $0 < \epsilon \leq 1$, we can decide whether $\mathrm{OPT}(I) = 0$, and otherwise generate another instance I' in $O(|\mathcal{F}||\mathcal{D}|\log(|\mathcal{F}||\mathcal{D}|))$ time, such that all service and facility costs are integers in $\{0, 1, \dots, \frac{2\gamma_{\max}(k+|\mathcal{D}|)^3}{\epsilon}\}$, and for each $1 \leq \gamma \leq \gamma_{\max}$, each solution to I' with cost at most $\gamma\,\mathrm{OPT}(I')$ is a solution to I with cost at most $\gamma(1+\epsilon)\,\mathrm{OPT}(I)$.*

Proof: Let $n := k + |\mathcal{D}|$. Given an instance I, we first compute an upper bound and a lower bound on $\mathrm{OPT}(I)$ differing by a factor $2n^2 - 1$ as follows. For each $B \in \{f_i : i \in \mathcal{F}\} \cup \{c_{ij} : i \in \mathcal{F}, j \in \mathcal{D}\}$ we consider the bipartite graph $G_B := (\mathcal{D} \cup \mathcal{F}, \{\{i,j\} : i \in \mathcal{F}, j \in \mathcal{D}, f_i \leq B, c_{ij} \leq B\})$.

The smallest B for which the elements of $\mathcal{D}$ belong to at most k different connected components of G_B, each of which contains at least one facility, is a lower bound on $\mathrm{OPT}(I)$. This number B can be found in $O(|\mathcal{F}||\mathcal{D}|\log(|\mathcal{F}||\mathcal{D}|))$ time by a straightforward variant of Krukskal's Algorithm for minimum spanning trees.

Moreover, for this B we can choose an arbitrary facility in each connected component of G_B that contains an element of $\mathcal{D}$, and connect each customer with service cost at most $(2|\mathcal{D}|-1)B$ (using that service costs are metric). Thus $\mathrm{OPT}(I) \leq kB + (2|\mathcal{D}|-1)|\mathcal{D}|B < (2n^2-1)B$ unless $B = 0$, in which case we are done.

Thus we can ignore facilities and service costs exceeding $B' := 2\gamma_{\max}n^2B$. We obtain I' from I by rounding each c_{ij} to $\lceil\frac{\min\{B', c_{ij}\}}{\delta}\rceil$ and each f_i to $\lceil\frac{\min\{B', f_i\}}{\delta}\rceil$, where $\delta = \frac{\epsilon B}{n}$. Now all input numbers are integers in $\{0, 1, \dots, \lceil\frac{2\gamma_{\max}n^3}{\epsilon}\rceil\}$.

We have

$$\mathrm{OPT}(I') \leq \frac{\mathrm{OPT}(I)}{\delta} + n = \frac{\mathrm{OPT}(I) + \epsilon B}{\delta} < \frac{(2n^2-1)B + \epsilon B}{\delta} \leq \frac{2n^2 B}{\delta} = \frac{B'}{\gamma_{\max}\delta},$$

and thus a solution to I' of cost at most $\gamma\,\mathrm{OPT}(I')$ contains no element of cost $\lceil \frac{B'}{\delta} \rceil$, and hence is a solution to I of cost at most

$$\delta\gamma\,\mathrm{OPT}(I') \le \gamma(\mathrm{OPT}(I) + \epsilon B) \le \gamma(1+\epsilon)\,\mathrm{OPT}(I). \qquad \square$$

Corollary 5.5 *There is a strongly polynomial 4-factor approximation algorithm for the metric k-Facility Location Problem.*

Proof: Apply Lemma 5.4 with $\gamma_{\max} = 4$ and $\epsilon = \frac{1}{4|\mathcal{D}|}$, and apply Theorem 5.3 with the Jain-Mahdian-Saberi algorithm to the resulting instance. We have $\gamma_S = 2$ by Corollary 4.4 and get a solution of total cost at most

$$\left(2 - \frac{1}{2|\mathcal{D}|}\right)\left(1 + \frac{1}{4|\mathcal{D}|}\right)(c_F(X^*) + \gamma_S c_S(X^*)) \le 4\left(c_F(X^*) + c_S(X^*)\right). \qquad \square$$

6 Local Search

Local search is a technique that is often applied successfully in practice, although usually no good approximation guarantees can be shown. It was therefore a surprise to learn that facility location problems can be approximated well by local search. This was first explored by Korupolu, Plaxton and Rajaraman [2000] and led to several strong results subsequently. We shall present some of them in this and the next section.

The main advantage of local search algorithms is their flexibility; they can be applied to arbitrary cost functions and even in the presence of complicated additional constraints. We will see this in Section 7 when we deal with a quite general problem, including hard capacities.

In this section we consider the k-Median Problem and the Uncapacitated Facility Location Problem, both with metric service costs. For the metric k-Median Problem, local search yields the best known performance ratio. Before presenting this result, we start with the simplest possible local search algorithm.

6.1 Single Swaps for the k-Median Problem

We start with an arbitrary feasible solution (set of k facilities) and improve it by certain "local" steps. Let us first consider single swaps only.

Theorem 6.1 *(Arya et al. [2004]) Consider an instance of the metric k-Median Problem. Let X be a feasible solution and X^* an optimum solution. If $c_S((X \setminus \{x\}) \cup \{y\}) \ge c_S(X)$ for all $x \in X$ and $y \in X^*$, then $c_S(X) \le 5c_S(X^*)$.*

Proof: Let us consider optimum assignments σ and σ^* of the customers to the k facilities in X and X^*, respectively. We say that $x \in X$ *captures* $y \in X^*$ if $|\{j \in \mathcal{D} : \sigma(j) = x, \sigma^*(j) = y\}| > \frac{1}{2}|\{j \in \mathcal{D} : \sigma^*(j) = y\}|$. Each $y \in X^*$ is captured by at most one $x \in X$.

Let $\pi : \mathcal{D} \to \mathcal{D}$ be a bijection such that for all $j \in \mathcal{D}$:

- $\sigma^*(\pi(j)) = \sigma^*(j)$; and
- if $\sigma(\pi(j)) = \sigma(j)$ then $\sigma(j)$ captures $\sigma^*(j)$.

Such a mapping π can be obtained easily by ordering, for each $y \in X^*$, the elements of $\{j \in \mathcal{D} : \sigma^*(j) = y\} = \{j_0, \dots, j_{t-1}\}$ such that customers j with identical $\sigma(j)$ are consecutive, and setting $\pi(j_k) := j_{k'}$, where $k' = (k + \lfloor \frac{t}{2} \rfloor) \bmod t$.

We now define k swaps (x, y) with $x \in X$ and $y \in X^*$. Each $y \in X^*$ will serve as target in exactly one of these swaps.

If an $x \in X$ captures only one facility $y \in X^*$, we consider a swap (x, y). If there are l such swaps, then there are $k - l$ elements left in X and in X^*. Some of the remaining elements of X (at most $\frac{k-l}{2}$) may capture at least two facilities of X^*; we will not consider these. For each remaining facility $y \in X^*$ we choose an $x \in X$ such that x does not capture any facility, and such that each $x \in X$ is source of at most two such swaps.

We now analyze the swaps one by one. Consider the swap (x, y), i.e. the set of facilities changes from X to $X' := (X \setminus \{x\}) \cup \{y\}$; note that the case $y \in X$ is not excluded. If $|X'| < |X| = k$, then X' can be extended by an arbitrary facility without increasing the cost. Transform $\sigma : \mathcal{D} \to X$ to a new assignment $\sigma' : \mathcal{D} \to X'$ by reassigning customers as follows:

Customers $j \in \mathcal{D}$ with $\sigma^*(j) = y$ are assigned to y. Customers $j \in \mathcal{D}$ with $\sigma(j) = x$ and $\sigma^*(j) = y' \in X^* \setminus \{y\}$ are assigned to $\sigma(\pi(j))$; note that $\sigma(\pi(j)) \neq x$ as x does not capture y'. For all other customers, the assignment does not change.

We have

$$\begin{aligned}
0 &\leq c_S(X') - c_S(X) \\
&\leq \sum_{j \in \mathcal{D} : \sigma^*(j) = y} \left(c_{\sigma^*(j)j} - c_{\sigma(j)j}\right) + \sum_{j \in \mathcal{D} : \sigma(j) = x, \sigma^*(j) \neq y} \left(c_{\sigma(\pi(j))j} - c_{\sigma(j)j}\right) \\
&\leq \sum_{j \in \mathcal{D} : \sigma^*(j) = y} \left(c_{\sigma^*(j)j} - c_{\sigma(j)j}\right) + \sum_{j \in \mathcal{D} : \sigma(j) = x} \left(c_{\sigma(\pi(j))j} - c_{\sigma(j)j}\right)
\end{aligned}$$

as $c_{\sigma(\pi(j))j} \geq \min_{i \in X} c_{ij} = c_{\sigma(j)j}$ by definition of σ.

We now sum over all swaps. Note that each facility of X^* is target of exactly one swap, thus the sum of the first terms is $c_S(X^*) - c_S(X)$. Moreover, each $x \in X$ is source of at most two swaps. Hence

$$0 \leq \sum_{j \in \mathcal{D}} \left(c_{\sigma^*(j)j} - c_{\sigma(j)j}\right) + 2 \sum_{j \in \mathcal{D}} \left(c_{\sigma(\pi(j))j} - c_{\sigma(j)j}\right)$$

$$\begin{aligned}
&\leq c_S(X^*) - c_S(X) + 2\sum_{j\in\mathcal{D}}(c_{\sigma^*(j)j} + c_{\sigma^*(j)\pi(j)} + c_{\sigma(\pi(j))\pi(j)} - c_{\sigma(j)j}) \\
&= c_S(X^*) - c_S(X) + 2\sum_{j\in\mathcal{D}}(c_{\sigma^*(j)j} + c_{\sigma^*(\pi(j))\pi(j)}) \\
&= c_S(X^*) - c_S(X) + 4c_S(X^*),
\end{aligned}$$

because π is a bijection. □

Thus a local optimum is a 5-approximation. However, this does not make any statement about the running time to achieve a local optimum, and the number of steps to reach a local optimum could in fact be exponential. However, by applying Lemma 5.4 (i.e. by discretizing costs) we obtain a strongly polynomial running time.

Another possibility is to make only moves that are sufficiently profitable:

Lemma 6.2 *(Arya et al. [2004]) Let $0 < \epsilon < 1$, and let a discrete minimization problem be given with objective function cost. Let $k \in \mathbb{N}$, let $\mathcal{N}(X)$ be a neighbourhood defined for each feasible solution X, and $\mathcal{N}'(X) \subseteq \mathcal{N}(X)$ with $|\mathcal{N}'(X)| \leq q$. Suppose that, for some $\alpha \geq 1$, each feasible solution X satisfies*

$$\sum_{X'\in\mathcal{N}'(X)} (cost(X') - cost(X)) \leq \alpha\,\mathrm{OPT} - cost(X). \tag{17}$$

Then consider a local search algorithm that starts with any feasible solution X and moves to an $X' \in \mathcal{N}(X)$ with $cost(X') < (1-\frac{\epsilon}{2\alpha q})cost(X)$ as long as there is such an X'. Then after a number of iterations which is polynomial in the input size, q and $\frac{1}{\epsilon}$, this algorithm stops with an $(\alpha+\epsilon)$-approximate solution.

Proof: Each iteration decreases the cost by at least a factor $1-\frac{\epsilon}{2\alpha q}$. If we start with solution X and end up with solution Y, the number of iterations is at most

$$\frac{\log\frac{cost(X)}{cost(Y)}}{\log\frac{1}{1-\frac{\epsilon}{2\alpha q}}} \leq \frac{2\alpha q}{\epsilon}(\log cost(X) - \log cost(Y)),$$

as $-\log(1-x) > x$ for $0 < x < 1$. Hence the number of iterations is polynomial in the input size, q and $\frac{1}{\epsilon}$.

We end up with a solution X such that $cost(X') \geq (1-\frac{\epsilon}{2\alpha q})cost(X)$ for all $X' \in \mathcal{N}(X)$. Summing over all $X' \in \mathcal{N}'(X)$ we get

$$\begin{aligned}
-\frac{\epsilon}{2\alpha}cost(X) &\leq -\frac{\epsilon}{2\alpha q}|\mathcal{N}'(X)|cost(X) \\
&\leq \sum_{X'\in\mathcal{N}'(X)} (cost(X') - cost(X)) \\
&\leq \alpha\,\mathrm{OPT} - cost(X)
\end{aligned}$$

and hence

$$cost(X) < \frac{\alpha}{1 - \frac{\epsilon}{2\alpha}} \,\mathrm{OPT} \leq (\alpha + \epsilon)\,\mathrm{OPT}$$

as $\epsilon \leq 1 \leq \alpha$. $\square$

In the above theorem as well as in the following ones, we showed (17) for a polynomially bounded q. The most efficient running time is obtained by combining Lemma 5.4 and Lemma 6.2

A related result on achieving approximate local optima has been obtained by Orlin, Punnen and Schulz [2004]: They showed that for a general discrete minimization problem with modular objective function and a neighbourhood for which each local minimum costs at most α times the optimum, and for which the question whether a given feasible solution is a local optimum can be decided in polynomial time, an $(\alpha + \epsilon)$-approximate solution can be obtained for any $\epsilon > 0$ by a combination of local search and scaling, in a running time that depends polynomially on the input size and linearly on $\frac{1}{\epsilon}$.

6.2 Better Performance Guarantee by Multiswaps

We have shown that local search with single swaps yields a $(5 + \epsilon)$-factor approximation algorithm for the metric k-Median Problem for any $\epsilon > 0$. Using multiswaps the approximation guarantee can be improved significantly:

Theorem 6.3 *(Arya et al. [2004]) Consider an instance of the metric k-*Median Problem*, and let $p \in \mathbb{N}$. Let X be a feasible solution and X^* an optimum solution. If $c_S((X \setminus \{A\}) \cup \{B\}) \geq c_S(X)$ for all $A \subseteq X$ and $B \subseteq X^*$ with $|A| = |B| \leq p$, then $c_S(X) \leq (3 + \frac{2}{p})c_S(X^*)$.*

Proof: Let σ and σ^* be again optimum assignments of the customers to the k facilities in X and X^*, respectively. For each $A \subseteq X$, let $C(A)$ be the set of facilities in X^* that are captured by A, i.e.

$$C(A) := \left\{ y \in X^* : |\{j \in \mathcal{D} : \sigma(j) \in A, \sigma^*(j) = y\}| > \frac{1}{2}|\{j \in \mathcal{D} : \sigma^*(j) = y\}| \right\}.$$

We partition $X = A_1 \,\dot\cup \cdots \dot\cup\, A_r$ and $X^* = B_1 \,\dot\cup \cdots \dot\cup\, B_r$ as follows:

```
Let $\{x \in X : C(\{x\}) \neq \emptyset\} = \{x_1, \dots, x_r\} =: \bar{X}$.
For $i = 1$ to $r - 1$ do:
    Set $A_i := \{x_i\}$.
    While $|A_i| < |C(A_i)|$ do:
        Add an element $x \in X \setminus (A_1 \cup \cdots \cup A_i \cup \bar{X})$ to $A_i$.
    Set $B_i := C(A_i)$.
Set $A_r := X \setminus (A_1 \cup \cdots \cup A_{r-1})$ and $B_r := X^* \setminus (B_1 \cup \cdots \cup B_{r-1})$.
```

It is clear that this algorithm guarantees $|A_i| = |B_i|$ for $i = 1, \dots, r$, and that the sets $A_1, \dots, A_r$ are pairwise disjoint and $B_1, \dots, B_r$ are pairwise disjoint. Note that adding an element is always possible if $|A_i| < |C(A_i)|$, because then $|X \setminus (A_1 \cup \dots \cup A_{i-1} \cup \bar{X})| = |X| - |A_1| - \dots - |A_i| - |\{x_{i+1}, \dots, x_r\}| > |X^*| - |C(A_1)| - \dots - |C(A_i)| - |C(\{x_{i+1}\})| - \dots - |C(\{x_r\})| = |X^* \setminus (C(A_1) \cup \dots \cup C(A_i) \cup C(\{x_{i+1}\}) \cup \dots \cup C(\{x_r\}))| \geq 0$.

Let $\pi : \mathcal{D} \to \mathcal{D}$ be a bijection such that for all $j \in \mathcal{D}$:

- $\sigma^*(\pi(j)) = \sigma^*(j)$;
- if $\sigma(\pi(j)) = \sigma(j)$ then $\sigma(j)$ captures $\sigma^*(j)$; and
- if $\sigma(j) \in A_i$ and $\sigma(\pi(j)) \in A_i$ for some $i \in \{1, \dots, r\}$, then A_i captures $\sigma^*(j)$.

Such a mapping π can be obtained almost identically as above.

We now define a set of swaps (A, B) with $|A| = |B| \leq p$, $A \subseteq X$ and $B \subseteq X^*$. Each swap will be associated with a positive weight. The swap (A, B) means that X is replaced by $X' := (X \setminus A) \cup B$; we say that A is the source set and B is the target set. If $|X'| < |X| = k$, we can extend X' by adding arbitrary $k - |X'|$ facilities without increasing the cost.

For each $i \in \{1, \dots, r\}$ with $|A_i| \leq p$, we consider the swap (A_i, B_i) with weight 1. For each $i \in \{1, \dots, r\}$ with $|A_i| = q > p$, we consider the swap $(\{x\}, \{y\})$ with weight $\frac{1}{q-1}$ for each $x \in A_i \setminus \{x_i\}$ and $y \in B_i$. Each $y \in X^*$ appears in the target set of swaps of total weight 1, and each $x \in X$ appears in the source set of swaps of total weight at most $\frac{p+1}{p}$.

We reassign customers as with the single swaps. More precisely, for a swap (A, B) we reassign all $j \in \mathcal{D}$ with $\sigma^*(j) \in B$ to $\sigma^*(j)$ and all $j \in \mathcal{D}$ with $\sigma^*(j) \notin B$ and $\sigma(j) \in A$ to $\sigma(\pi(j))$. Note that we have $B \supseteq C(A)$ for each of the considered swaps (A, B). Thus, for all $j \in \mathcal{D}$ with $\sigma(j) \in A$ and $\sigma^*(j) \notin B$ we have $\sigma(\pi(j)) \notin A$. Therefore we can bound the increase of the cost due to the swap as follows:

$$\begin{aligned}
0 &\leq c_S(X') - c_S(X) \\
&\leq \sum_{j \in \mathcal{D}: \sigma^*(j) \in B} (c_{\sigma^*(j)j} - c_{\sigma(j)j}) + \sum_{j \in \mathcal{D}: \sigma(j) \in A, \sigma^*(j) \notin B} (c_{\sigma(\pi(j))j} - c_{\sigma(j)j}) \\
&\leq \sum_{j \in \mathcal{D}: \sigma^*(j) \in B} (c_{\sigma^*(j)j} - c_{\sigma(j)j}) + \sum_{j \in \mathcal{D}: \sigma(j) \in A} (c_{\sigma(\pi(j))j} - c_{\sigma(j)j})
\end{aligned}$$

as $c_{\sigma(\pi(j))j} \geq c_{\sigma(j)j}$ by definition of σ. Hence taking the weighted sum over all swaps yields

$$0 \leq \sum_{j \in \mathcal{D}} (c_{\sigma^*(j)j} - c_{\sigma(j)j}) + \frac{p+1}{p} \sum_{j \in \mathcal{D}} (c_{\sigma(\pi(j))j} - c_{\sigma(j)j})$$

$$
\begin{aligned}
&\le\ c_S(X^*) - c_S(X) + \frac{p+1}{p}\sum_{j\in\mathcal{D}}\big(c_{\sigma^*(j)j} + c_{\sigma^*(j)\pi(j)} + c_{\sigma(\pi(j))\pi(j)} - c_{\sigma(j)j}\big)\\
&=\ c_S(X^*) - c_S(X) + \frac{p+1}{p}\sum_{j\in\mathcal{D}}\big(c_{\sigma^*(j)j} + c_{\sigma^*(\pi(j))\pi(j)}\big)\\
&=\ c_S(X^*) - c_S(X) + 2\,\frac{p+1}{p}c_S(X^*),
\end{aligned}
$$

because π is a bijection. □

Arya et al. [2004] also showed that this performance guarantee is tight. We have a $(3+\epsilon)$-factor approximation algorithm for any $\epsilon > 0$, which is the currently best known approximation guarantee for the metric k-Median Problem.

6.3 Local Search for Uncapacitated Facility Location

We apply similar techniques to the Uncapacitated Facility Location Problem to obtain a simple approximation algorithm based on local search:

Theorem 6.4 *(Arya et al. [2004]) Consider an instance of the metric* Uncapacitated Facility Location Problem. *Let X and X^* be any feasible solutions. If neither $X\setminus\{x\}$ nor $X\cup\{y\}$ nor $(X\setminus\{x\})\cup\{y\}$ is better than X for any $x\in X$ and $y\in\mathcal{F}\setminus X$, then $c_S(X)\le c_F(X^*)+c_S(X^*)$ and $c_F(X)\le c_F(X^*)+2c_S(X^*)$.*

Proof: We use the same notation as in the previous proofs. In particular, let σ and σ^* be optimum assignments of the customers to X and X^*, respectively.

The first inequality is easily proved by considering the operations of adding an $y\in X^*$ to X, which increases the cost by at most $f_y+\sum_{j\in\mathcal{D}:\sigma^*(j)=y}(c_{\sigma^*(j)j}-c_{\sigma(j)j})$. Summing these values up yields that $c_F(X^*)+c_S(X^*)-c_S(X)$ is nonnegative.

Let again $\pi:\mathcal{D}\to\mathcal{D}$ be a bijection such that for all $j\in\mathcal{D}$:

- $\sigma^*(\pi(j)) = \sigma^*(j)$;

- if $\sigma(\pi(j)) = \sigma(j)$ then $\sigma(j)$ captures $\sigma^*(j)$ and $\pi(j)=j$.

Such a mapping π can be obtained identically as above after fixing $\pi(j):=j$ for $|\{j\in\mathcal{D}:\sigma^*(j)=y,\sigma(j)=x\}|-|\{j\in\mathcal{D}:\sigma^*(j)=y,\sigma(j)\neq x\}|$ elements $j\in\mathcal{D}$ with $\sigma^*(j)=y$ and $\sigma(j)=x$ for any pair $x\in X$, $y\in X^*$ where x captures y.

To bound the facility cost of X, let $x\in X$, and let $\mathcal{D}_x:=\{j\in\mathcal{D}:\sigma(j)=x\}$. If x does not capture any $y\in X^*$, we consider dropping x and reassigning each $j\in\mathcal{D}_x$ to $\sigma(\pi(j))\in X\setminus\{x\}$. Hence

$$0\le -f_x+\sum_{j\in\mathcal{D}_x}\big(c_{\sigma(\pi(j))j}-c_{xj}\big). \tag{18}$$

If the set $C(\{x\})$ of facilities captured by x is nonempty, let $y \in C(\{x\})$ be the nearest facility in $C(\{x\})$ (i.e. $\min_{j\in\mathcal{D}}(c_{xj}+c_{yj})$ is minimum). We consider the addition of each facility $y' \in C(\{x\}) \setminus \{y\}$, which increases the cost by at most

$$f_{y'} + \sum_{j\in\mathcal{D}_x:\sigma^*(j)=y',\pi(j)=j} (c_{\sigma^*(j)j} - c_{xj}). \tag{19}$$

Moreover, we consider the swap $(\{x\},\{y\})$. For $j \in \mathcal{D}_x$ we reassign j to $\sigma(\pi(j))$ if $\pi(j) \neq j$, and to y otherwise.

The new service cost for $j \in \mathcal{D}_x$ is at most $c_{\sigma(\pi(j))j}$ in the first case, $c_{\sigma^*(j)j}$ if $\pi(j) = j$ and $\sigma^*(j) = y$, and

$$c_{yj} \leq c_{xj} + \min_{k\in\mathcal{D}}(c_{xk}+c_{yk}) \leq c_{xj} + \min_{k\in\mathcal{D}}(c_{xk}+c_{\sigma^*(j)k}) \leq 2c_{xj} + c_{\sigma^*(j)j}$$

otherwise, as x captures $\sigma^*(j)$ in the latter case.

Altogether, the swap from x to y increases the cost by at most

$$\begin{aligned} & f_y - f_x - \sum_{j\in\mathcal{D}_x} c_{xj} + \sum_{j\in\mathcal{D}_x:\pi(j)\neq j} c_{\sigma(\pi(j))j} \\ & + \sum_{j\in\mathcal{D}_x:\pi(j)=j,\sigma^*(j)=y} c_{\sigma^*(j)j} + \sum_{j\in\mathcal{D}_x:\pi(j)=j,\sigma^*(j)\neq y} (2c_{xj} + c_{\sigma^*(j)j}). \end{aligned} \tag{20}$$

Adding the nonnegative terms (19) and (20) yields

$$\begin{aligned} 0 \quad \leq \quad & \sum_{y'\in C(x)} f_{y'} - f_x + \sum_{j\in\mathcal{D}_x:\pi(j)\neq j} (c_{\sigma(\pi(j))j} - c_{xj}) \\ & + \sum_{j\in\mathcal{D}_x:\pi(j)=j,\sigma^*(j)=y} (c_{\sigma^*(j)j} - c_{xj}) + \sum_{j\in\mathcal{D}_x:\pi(j)=j,\sigma^*(j)\neq y} 2c_{\sigma^*(j)j} \\ \leq \quad & \sum_{y'\in C(x)} f_{y'} - f_x + \sum_{j\in\mathcal{D}_x:\pi(j)\neq j} (c_{\sigma(\pi(j))j} - c_{xj}) + \ 2 \sum_{j\in\mathcal{D}_x:\pi(j)=j} c_{\sigma^*(j)j} \end{aligned} \tag{21}$$

Summing (18) and (21), respectively, over all $x \in X$ yields

$$\begin{aligned} 0 \quad \leq \quad & \sum_{x\in X}\sum_{y'\in C(x)} f_{y'} - c_F(X) + \sum_{j\in\mathcal{D}:\pi(j)\neq j} (c_{\sigma(\pi(j))j} - c_{\sigma(j)j}) + 2\sum_{j\in\mathcal{D}:\pi(j)=j} c_{\sigma^*(j)j} \\ \leq \quad & c_F(X^*) - c_F(X) + \sum_{j\in\mathcal{D}:\pi(j)\neq j} (c_{\sigma^*(j)j} + c_{\sigma^*(j)\pi(j)} + c_{\sigma(\pi(j))\pi(j)} - c_{\sigma(j)j}) \\ & +2\sum_{j\in\mathcal{D}:\pi(j)=j} c_{\sigma^*(j)j} \\ = \quad & c_F(X^*) - c_F(X) + 2c_S(X^*). \end{aligned}$$

□

This directly implies that we have a $(3+\epsilon)$-approximation for any $\epsilon > 0$. However, by Theorem 4.7, we get a 2.375-approximation algorithm by multiplying facility costs by $\delta = 1.4547$, applying the above local search technique, scaling facility costs back, and applying greedy augmentation.

Instead, for a simpler result, we could multiply facility costs by $\sqrt{2}$ before applying local search, and we obtain a solution X with $c_S(X) \leq \sqrt{2}c_F(X^*) + c_S(X^*)$ and $\sqrt{2}c_F(X) \leq \sqrt{2}c_F(X^*) + 2c_S(X^*)$. Hence $c_F(X) + c_S(X) \leq (1 + \sqrt{2})(c_F(X^*) + c_S(X^*))$, and we get the slightly worse approximation guarantee of Arya et al. [2004], namely $(1 + \sqrt{2} + \epsilon) \approx 2.415$, without greedy augmentation. (This can be viewed as an application of Proposition 4.9.)

Charikar and Guha [1999] proved the same approximation guarantee for a very similar local serach algorithm.

7 Capacitated Facility Location

In this section we consider hard capacities. Local search is the only technique known to lead to an approximation guarantee for hard capacities. In fact, we show that an even more general problem, called the Universal Facility Location Problem, can be solved up to a factor of 6.702 by a local search algorithm. This includes the metric Capacitated Facility Location Problem (with hard capacities), for which local search is also the only technique that is known to lead to an approximation algorithm. But it also includes simpler problems like the metric Soft-Capacitated Facility Location Problem.

When dealing with hard capacities, we have to allow the demand of customers to be split, i.e. assigned to multiple open facilities. However, the total demand assigned to a facility must not exceed its capacity. If we do not allow splitting, we cannot expect any result as even deciding whether a feasible solution exists at all is *NP*-complete (it contains the well-known Partition problem).

7.1 Capacitated and Universal Facility Location

The (metric) Capacitated Facility Location Problem is defined as follows:

Given:

- a finite set V and distances $c_{ij} \geq 0$ $(i, j \in V)$ with $c_{ij} + c_{jk} \geq c_{ik}$ for all $i, j, k \in V$.
- a set $\mathcal{D} \subseteq V$ of customers (or clients);
- a set $\mathcal{F} \subseteq V$ of potential facilities;
- a fixed cost $f_i \geq 0$ for opening each facility $i \in \mathcal{F}$;
- a capacity $u_i \geq 0$ for each facility;
- a demand $d_j \geq 0$ for each client.

we look for:

- a subset S of facilities (called *open*) and
- an assignment $x : S \times \mathcal{D} \to \mathbb{R}_+$ of customers' demand to open facilities, where $\sum_{i \in S} x_{ij} = d_j$ for all $j \in \mathcal{D}$,

such that the sum of facility costs and service costs

$$\sum_{i \in S} f_i + \sum_{i \in S} \sum_{j \in \mathcal{D}} c_{ij} x_{ij}$$

is minimum.

Note that for a fixed set S of opened facilities, an optimum assignment x can be found be solving a transshipment problem. If all u_i and d_j are integral, there is an integral optimum assignment. In particular, if capacities are integral and all demands are 1, the demand will not be split.

The first approximation algorithm for the general case is due to Pál, Tardos and Wexler [2001], extending an earlier result for a special case by Korupolo, Plaxton and Rajamaran [2000]. The approximation guarantee was then improved to 5.83 by Zhang, Chen and Ye [2004]. For the special case of uniform facility opening costs, Levi, Shmoys and Swamy [2004] obtained a 5-factor approximation algorithm by rounding an LP relaxation.

The work by Pál, Tardos and Wexler has been generalized to the so-called Universal Facility Location Problem by Mahdian and Pál [2003]. In the Universal Facility Location Problem we are given $V, c, \mathcal{F}, \mathcal{D}, d$ as above, and a non-decreasing, left-continuous function $f_i : \mathbb{R}_+ \to \mathbb{R}_+ \cup \{\infty\}$, where $f_i(u)$ is the cost to install capacity u at facility i. The task is to find numbers $x_{ij} \geq 0$ ($i \in \mathcal{F}$, $j \in \mathcal{D}$) with $\sum_{i \in \mathcal{F}} x_{ij} = d_j$ for all $j \in \mathcal{D}$ (i.e. a fractional assignment) such that $c(x) := c_F(x) + c_S(x)$ is minimum, where

$$c_F(x) := \sum_{i \in \mathcal{F}} f_i \left(\sum_{j \in \mathcal{D}} x_{ij} \right) \qquad \text{and} \qquad c_S(x) := \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{D}} c_{ij} x_{ij}.$$

Lemma 7.1 *(Mahdian and Pál [2003]) Every instance of the* Universal Facility Location Problem *has an optimum solution.*

Proof: If there is no solution with finite cost, any solution is optimum. Otherwise let $(x^i)_{i \in \mathbb{N}}$ be a sequence of solutions whose costs approach the infimum c^* of the set of costs of feasible solutions. As this sequence is bounded, there is a subsequence $(x^{i_j})_{j \in \mathbb{N}}$ converging to some x^*. As all f_i are left-continuous and non-decreasing, we have $c(x^*) = c(\lim_{j \to \infty} x^{i_j}) \leq \lim_{j \to \infty} c(x^{i_j}) = c^*$, i.e. x^* is optimum. $\square$

For an algorithm, we have to specify how the functions f_i are given. We need an oracle that, for each $i \in \mathcal{F}$, $u, c \in \mathbb{R}_+$ and $t \in \mathbb{R}$, computes $f_i(u)$ and $\max\{\delta \in \mathbb{R} : u+\delta \geq 0,\ f_i(u+\delta) - f_i(u) + c|\delta| \leq t\}$. This is a natural assumption as this oracle can be implemented trivially for all special cases of the Universal Facility Location Problem considered before.

7.2 Add Operation

In the following sections, based on Vygen [2005], we will present a local search algorithm for the Universal Facility Location Problem. It uses two operations. First, for $t \in \mathcal{F}$ and $\delta \in \mathbb{R}_+$ we consider the operation Add(t, δ), which means replacing the current feasible solution x by an optimum solution y of the transshipment problem

$$\min\Bigg\{c_S(y) \ \ : \ \ y : \mathcal{F} \times \mathcal{D} \to \mathbb{R}_+,\ \sum_{i \in \mathcal{F}} y_{ij} = d_j \ (j \in \mathcal{D}), \\ \sum_{j \in \mathcal{D}} y_{ij} \leq \sum_{j \in \mathcal{D}} x_{ij} \ (i \in \mathcal{F} \setminus \{t\}),\ \sum_{j \in \mathcal{D}} y_{tj} \leq \sum_{j \in \mathcal{D}} x_{tj} + \delta\Bigg\}.$$

We denote by $c^x(t, \delta) := c_S(y) - c_S(x) + f_t(\sum_{j \in \mathcal{D}} x_{tj} + \delta) - f_t(\sum_{j \in \mathcal{D}} x_{tj})$ the estimated cost of this operation; this is an upper bound on $c(y) - c(x)$.

Lemma 7.2 *(Mahdian and Pál [2003]) Let $\epsilon > 0$ and $t \in \mathcal{F}$. Let x be a feasible solution to a given instance. Then there is an algorithm with running time $O(|V|^3 \log |V| \epsilon^{-1})$ to find a $\delta \in \mathbb{R}_+$ with $c^x(t, \delta) \leq -\epsilon c(x)$ or decide that no $\delta \in \mathbb{R}_+$ exists for which $c^x(t, \delta) \leq -2\epsilon c(x)$.*

Proof: Let $C := \{\nu \epsilon c(x) : \nu \in \mathbb{Z}_+, \nu \leq \frac{1}{\epsilon}\}$. For each $\gamma \in C$ let δ_γ be the maximum $\delta \in \mathbb{R}_+$ for which $f_t(\sum_{j \in \mathcal{D}} x_{tj} + \delta) - f_t(\sum_{j \in \mathcal{D}} x_{tj}) \leq \gamma$. We compute $c^x(t, \delta_\gamma)$ for all $\gamma \in C$.

Suppose there is a $\delta \in \mathbb{R}_+$ with $c^x(t, \delta) \leq -2\epsilon c(x)$. Then consider $\gamma := \epsilon c(x) \lceil \frac{1}{\epsilon c(x)} (f_t(\sum_{j \in \mathcal{D}} x_{tj} + \delta) - f_t(\sum_{j \in \mathcal{D}} x_{tj})) \rceil \in C$. Note that $\delta_\gamma \geq \delta$ and hence $c^x(t, \delta_\gamma) < c^x(t, \delta) + \epsilon c(x) \leq -\epsilon c(x)$.

The running time is dominated by solving $|C|$ transshipment problems in a digraph with $|V|$ vertices. □

If there is no sufficiently profitable Add operation, the service cost can be bounded. The following result is essentially due to Pál, Tardos and Wexler [2001]:

Lemma 7.3 *Let $\epsilon > 0$, and let x, x^* be feasible solutions to a given instance, and let $c^x(t, \delta) \geq -\frac{\epsilon}{|\mathcal{F}|} c(x)$ for all $t \in \mathcal{F}$ and $\delta \in \mathbb{R}_+$. Then $c_S(x) \leq c_F(x^*) + c_S(x^*) + \epsilon c(x)$.*

Proof: Consider the (complete bipartite) digraph $G = (\mathcal{D} \mathbin{\dot\cup} \mathcal{F}, (\mathcal{D} \times \mathcal{F}) \cup (\mathcal{F} \times \mathcal{D}))$ with edge weights $c((j,i)) := c_{ij}$ and $c((i,j)) := -c_{ij}$ for $i \in \mathcal{F}$ and $j \in \mathcal{D}$. Let $b(i) := \sum_{j\in\mathcal{D}}(x_{ij} - x^*_{ij})$ for $i \in \mathcal{F}$, $S := \{i \in \mathcal{F} : b(i) > 0\}$ and $T := \{i \in \mathcal{F} : b(i) < 0\}$.

Define a b-flow $g : E(G) \to \mathbb{R}_+$ by $g(i,j) := \max\{0, x_{ij} - x^*_{ij}\}$ and $g(j,i) := \max\{0, x^*_{ij} - x_{ij}\}$ for $i \in \mathcal{F}$, $j \in \mathcal{D}$.

Write g as the sum of b_t-flows g_t for $t \in T$, where $b_t(t) = b(t)$ and $0 \le b_t(v) \le b(v)$ for $v \in V(G) \setminus T$. (This can be done by standard flow decomposition techniques.)

For each $t \in T$, g_t defines a feasible way to reassign customers to t, i.e. a new solution x^t defined by $x^t_{ij} := x_{ij} + g_t(j,i) - g_t(i,j)$ for $i \in \mathcal{F}$, $j \in \mathcal{D}$. We have $c_S(x^t) = c_S(x) + \sum_{e\in E(G)} c(e)g_t(e)$ and hence

$$c^x(t, b(t)) \le \sum_{e\in E(G)} c(e)g_t(e) + f_t\bigg(\sum_{j\in\mathcal{D}} x^*_{tj}\bigg) - f_t\bigg(\sum_{j\in\mathcal{D}} x_{tj}\bigg).$$

If the left-hand side is at least $-\frac{\epsilon}{|\mathcal{F}|}c(x)$ for each $t \in T$, summation yields

$$\begin{aligned} -\epsilon c(x) &\le \sum_{e\in E(G)} c(e)g(e) + \sum_{t\in T} f_t\bigg(\sum_{j\in\mathcal{D}} x^*_{tj}\bigg) \\ &\le \sum_{e\in E(G)} c(e)g(e) + c_F(x^*) \\ &= c_S(x^*) - c_S(x) + c_F(x^*). \end{aligned}$$

□

7.3 Pivot Operation

Let x be a feasible solution for a given instance of the Universal Facility Location Problem. Let A be an arborescence with $V(A) \subseteq \mathcal{F}$ and $\delta \in \Delta^x_A := \{\delta \in \mathbb{R}^{V(A)} : \sum_{j\in\mathcal{D}} x_{ij} + \delta_i \ge 0 \text{ for all } i \in V(A), \sum_{i\in V(A)} \delta_i = 0\}$.

Then we consider the operation $\textsc{Pivot}(A, \delta)$, which means replacing x by a solution x' with $\sum_{j\in\mathcal{D}} x'_{ij} = \sum_{j\in\mathcal{D}} x_{ij} + \delta_i$ for $i \in V(A)$ and $c(x') \le c(x) + c^x(A, \delta)$, where $c^x(A, \delta) := \sum_{i\in V(A)} c^x_{A,i}(\delta)$ and

$$c^x_{A,i}(\delta) := f_i\bigg(\sum_{j\in\mathcal{D}} x_{ij} + \delta_i\bigg) - f_i\bigg(\sum_{j\in\mathcal{D}} x_{ij}\bigg) + \bigg|\sum_{j\in A^+_i} \delta_j\bigg| c_{ip(i)}$$

for $i \in V(A)$. Here A^+_i denotes the set of vertices reachable from i in A, and $p(i)$ is the predecessor of i in A (and $p(i)$ arbitrary if i is the root). Such an x' can be constructed easily by moving demand along the edges in A in reverse topological order. Note that the orientation of A is irrelevant and used only to simplify the notation.

The operation will be performed if its *estimated cost* $c^x(A,\delta)$ is at most $-\epsilon c(x)$. This guarantees that the resulting local search algorithm stops after a polynomial number of improvement steps. We call $\sum_{i\in V(A)}\Big|\sum_{j\in A_i^+}\delta_j\Big|c_{ip(i)}$ the *estimated routing cost* of PIVOT(A,δ).

We now show how to find an improving PIVOT operation unless we are at an approximate local optimum:

Lemma 7.4 *(Vygen [2005]) Let $\epsilon > 0$ and A an arborescence with $V(A) \subseteq \mathcal{F}$. Let x be a feasible solution. Then there is an algorithm with running time $O(|\mathcal{F}|^4\epsilon^{-3})$ to find a $\delta \in \Delta_A^x$ with $c^x(A,\delta) \le -\epsilon c(x)$ or decide that no $\delta \in \Delta_A^x$ exists for which $c^x(A,\delta) \le -2\epsilon c(x)$.*

Proof: Number $V(A) = \{1,\dots,n\}$ in reverse topological order, i.e. for all $(i,j) \in A$ we have $i > j$. For $k \in V(A)$ with $(p(k),k) \in E(A)$ let $B(k) := \{i < k : (p(k),i) \in E(A)\}$ be the set of smaller siblings of k, and let $B(k) := \emptyset$ if k is the root of A. Let $I_k := \bigcup_{j\in B(k)\cup\{k\}} A_j^+$, $b(k) := \max(\{0\}\cup B(k))$ and $s(k) := \max(\{0\}\cup(A_k^+\setminus\{k\}))$.

Let $C := \{\nu\frac{\epsilon}{n}c(x) : \nu \in \mathbb{Z}, -\frac{n}{\epsilon} \le \nu \le \frac{n}{\epsilon}\}$. We compute the table $T_A^x : \{0,\dots,n\}\times C \to \Delta_A^x \cup \{\emptyset\}$ defined as follows. Let $T_A^x(0,0) := 0$, $T_A^x(0,\gamma) := \emptyset$ for all $\gamma \in C\setminus\{0\}$, and for $k = 1,\dots,n$ let $T_A^x(k,\gamma)$ be an optimum solution δ of

$$\max\Bigg\{\sum_{i\in I_k}\delta_i \;:\; \begin{array}{l}\delta_i = (T_A^x(b(k),\gamma'))_i \text{ for } i \in \bigcup_{j\in B(k)} A_j^+,\ \gamma' \in C,\\ \delta_i = (T_A^x(s(k),\gamma''))_i \text{ for } i \in A_k^+\setminus\{k\},\ \gamma'' \in C,\\ \sum_{j\in\mathcal{D}} x_{kj} + \delta_k \ge 0,\ \gamma' + \gamma'' + c_{A,k}^x(\delta) \le \gamma\end{array}\Bigg\}$$

if the set over which the maximum is taken is nonempty, and $T_A^x(k,\gamma) := \emptyset$ otherwise.

Roughly, $T_A^x(k,\gamma)$ is the minimum excess we get at the predecessor $p(k)$ of k when moving demand from each vertex in A_j^+ for $j \in B(k)\cup\{k\}$ to its predecessor or vice versa, at total rounded estimated cost at most γ.

Finally we choose the minimum $\gamma \in C$ such that $T_A^x(n,\gamma) \ne \emptyset$ and $\sum_{i=1}^n (T_A^x(n,\gamma))_i \ge 0$. Then we choose $\delta \in \Delta_A^x$ such that $\delta_i = (T_A^x(n,\gamma))_i$ or $\max\{0, -\sum_{j\in A_i^+\setminus\{i\}}\delta_j\} \le \delta_i \le (T_A^x(n,\gamma))_i$ for all $i = 1,\dots,n$. This corresponds to the operation PIVOT(A,δ) with $c^x(A,\delta) \le \gamma$.

Suppose there exists an operation PIVOT(A,δ) with $c^x(A,\delta) \le -2\epsilon c(x)$. As $c_{A,i}^x(\delta_i) \ge -f_i(\sum_{j\in\mathcal{D}} x_{ij}) \ge -c(x)$, this also implies $c_{A,i}^x(\delta_i) < c_F(x) \le c(x)$. Hence $\gamma_i := \lceil c_{A,i}^x(\delta_i)\frac{n}{\epsilon c(x)}\rceil\frac{\epsilon c(x)}{n} \in C$ for $i = 1,\dots,n$, and $\sum_{i\in I}\gamma_i \in C$ for all $I \subseteq \{1,\dots,n\}$. Then $\sum_{i\in I_k}(T_A^x(k,\sum_{j\in I_k}\gamma_j))_i \ge \sum_{i\in I_k}\delta_i$ for $k = 1,\dots,n$. Hence we find a pivot operation with estimated cost at most $\sum_{i=1}^n\gamma_i < c^x(A,\delta)+\epsilon c(x) \le -\epsilon c(x)$.

The running time can be estimated as follows. We have to compute $n|C|$ table entries, and for each entry $T^x_A(k,\gamma)$ we try all values of $\gamma', \gamma'' \in C$. This yields values δ_i for $i \in I_k \setminus \{k\}$, and the main step is to compute the maximum δ_k for which $\gamma' + \gamma'' + c^x_{A,k}(\delta) \leq \gamma$. This can be done directly with the oracle that we assumed for the functions f_i, $i \in \mathcal{F}$. The final computation of δ from $T^x_A(n,\gamma)$, $\gamma \in C$, is easily done in linear time. Hence the overall running time is $O(n|C|^3) = O(|\mathcal{F}|^4\epsilon^{-3})$. □

We consider $\textsc{Pivot}(A,\delta)$ for special arborescences: stars and comets. A is called a *star centered at* v if $A = (\mathcal{F}, \{(v,w) : w \in \mathcal{F} \setminus \{v\}\})$ and a *comet with center* v *and tail* (t,s) if $A = (\mathcal{F}, \{(t,s)\} \cup \{(v,w) : w \in \mathcal{F} \setminus \{v,s\}\})$ and v,t,s are distinct elements of $\mathcal{F}$. Note that there are less than $|\mathcal{F}|^3$ stars and comets. We remark that our Pivot operation for stars is identical to the pivot operation proposed by Mahdian and Pál [2003].

7.4 Bounding the Facility Cost

We will now show that an (approximate) local optimum has low facility cost. The first part of the following proof is identical to the one of Mahdian and Pál [2003]. The second part uses ideas of Zhang, Chen and Ye [2004], and exploits the new Pivot operation with comets.

Lemma 7.5 *Let x, x^* be feasible solutions to a given instance, and let $c^x(A,\delta) \geq -\frac{\epsilon}{|\mathcal{F}|}c(x)$ for all stars and comets A and $\delta \in \Delta^x_A$. Then $c_F(x) \leq 4c_F(x^*) + 2c_S(x^*) + 2c_S(x) + \epsilon c(x)$.*

Proof: We use the notation of Lemma 7.3 and consider the transshipment problem

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{s\in S, t\in T} c_{st}\, y(s,t) & \\ \text{subject to} & \displaystyle\sum_{t\in T} y(s,t) = b(s) & \text{for all } s \in S, \\ & \displaystyle\sum_{s\in S} y(s,t) = -b(t) & \text{for all } t \in T, \\ & y(s,t) \geq 0 & \text{for all } s \in S, t \in T. \end{array} \tag{22}$$

It is well-known from min-cost flow theory that there exists an optimum solution $y : S \times T \to \mathbb{R}_+$ of (22) such that $F := (S \cup T, \{\{s,t\} : y(s,t) > 0\})$ is a forest.

As $(b_t(s))_{s\in S,t\in T}$ is a feasible solution of (22), we have

$$\begin{aligned}\sum_{s\in S,t\in T} c_{st}y(s,t) &\le \sum_{s\in S,t\in T} c_{st}b_t(s)\\ &= \sum_{s\in S,t\in T} c_{st}(g_t(\delta^+(s)) - g_t(\delta^-(s)))\\ &\le \sum_{e\in E(G)} |c(e)|g(e)\\ &\le c_S(x^*) + c_S(x).\end{aligned} \tag{23}$$

We will now define at most $|\mathcal{F}|$ PIVOT operations. We say that an operation PIVOT(A,δ) *closes* $s\in S$ if $\sum_{j\in\mathcal{D}} x_{sj} > \sum_{j\in\mathcal{D}} x_{sj} + \delta_s = \sum_{j\in\mathcal{D}} x^*_{sj}$. We say that it *opens* $t\in T$ if $\sum_{j\in\mathcal{D}} x_{tj} < \sum_{j\in\mathcal{D}} x_{tj} + \delta_t \le \sum_{j\in\mathcal{D}} x^*_{tj}$. Over all operations that we are going to define, each $s\in S$ will be closed once, and each $t\in T$ will be opened at most four times. Moreover, the total estimated routing cost will be at most $2\sum_{s\in S,t\in T} c_{st}y(s,t)$. Thus the total estimated cost of the operations will be at most $4c_F(x^*) + 2c_S(x^*) + 2c_S(x) - c_F(x)$. This will prove the lemma.

To define the operations, consider a connected component of F, and reorient it as an arborescence B rooted at an element of T. Write $y(e) := y(s,t)$ if $e\in E(B)$ has endpoints $s\in S$ and $t\in T$. A vertex $v\in V(B)$ is called *weak* if $y(\delta^+_B(v)) > y(\delta^-_B(v))$ and *strong* otherwise. We denote by $\Gamma^+_s(v)$, $\Gamma^+_w(v)$ and $\Gamma^+(v)$ the set of strong, weak, and all children of $v\in V(B)$ in B, respectively. Let $t\in T$, and let $\Gamma^+_w(t) = \{w_1,\dots,w_k\}$ be the weak children of t ordered such that $r(w_1)\le\dots\le r(w_k)$, where $r(w_i) := \max\left\{0, y(w_i,t) - \sum_{t'\in\Gamma^+_w(w_i)} y(w_i,t')\right\}$. Moreover, order $\Gamma^+_s(t) = \{s_1,\dots,s_r\}$ such that $y(s_1,t)\ge\dots\ge y(s_r,t)$.

For $i = 1,\dots,k-1$ consider a PIVOT with a star centered at w_i, routing

- at most $2y(w_i,t')$ units of demand from w_i to each weak child t' of w_i,
- $y(w_i,t')$ units from w_i to each strong child t' of w_i, and
- $r(w_i)$ units from w_i to $\Gamma^+_s(w_{i+1})$,

closing w_i and opening $\Gamma^+(w_i)\cup\Gamma^+_s(w_{i+1})$. All customers j with $\sigma(j) = w_i$ are reassigned, and the estimated routing cost is at most

$$\sum_{t'\in\Gamma^+_w(w_i)} c_{w_it'}2y(w_i,t') + \sum_{t'\in\Gamma^+_s(w_i)} c_{w_it'}y(w_i,t') + c_{tw_i}r(w_i)$$
$$+c_{tw_{i+1}}r(w_{i+1}) + \sum_{t'\in\Gamma^+_s(w_{i+1})} c_{w_{i+1}t'}y(w_{i+1},t'),$$

as $r(w_i)\le r(w_{i+1})\le\sum_{t'\in\Gamma^+_s(w_{i+1})} y(w_{i+1},t')$.

Case 1: t is strong. Then consider PIVOT(w_k,δ) and PIVOT(t,δ), routing

- $y(w_k, t')$ units of demand from w_k to each child t' of w_k,
- $y(w_k, t)$ units from w_k to t, and
- at most $2y(s, t)$ units from each strong child s of t to t,

closing w_k and the strong children of t, opening the children of w_k, and opening t twice.

Case 2: t is weak and $y(w_k, t) + y(s_1, t) \geq \sum_{i=2}^{r} y(s_i, t)$. Then consider Pivot operations with the stars centered w_k, s_1, and t, routing

- $y(w_k, t')$ units of demand from w_k to each child t' of w_k,
- $y(w_k, t)$ units from w_k to t,
- $y(s_1, t')$ units from s_1 to each child t' of s_1,
- $y(s_1, t)$ units from s_1 to t, and
- at most $2y(s_i, t)$ units from s_i to t for $i = 2, \ldots, r$,

closing w_k and the strong children of t, opening the children of w_k and the children of s_1, and opening t three times.

Case 3: t is weak and $y(w_k, t) + y(s_1, t) < \sum_{i=2}^{r} y(s_i, t)$. Then consider a Pivot operation with the comet with center w_k and tail (t, s_1), routing

- $y(w_k, t')$ units of demand from w_k to each child t' of w_k,
- $y(w_k, t)$ units from w_k to t, and
- at most $2y(s_1, t)$ units from s_1 to t,

closing w_k and s_1 and opening t and the children of w_k.

Moreover, consider two Pivot operations with the star centered at t, where the first (second) one routes at most $2y(s_i, t)$ units of demand from s_i to t for each odd (even) element i of $\{2, \ldots, r\}$. This closes $s_2, \ldots, s_r$, and opens t twice.

We collect all these Pivot operations for all $t \in T$. Then, altogether, we have closed each $s \in S$ once and opened each $t \in T$ at most four times, with a total estimated routing cost of at most $2 \sum_{\{s,t\} \in E(F)} c_{st} y(s, t)$, which is at most $2c_S(x^*) + 2c_S(x)$ by (23). If none of the operations has estimated cost less than $-\frac{\epsilon}{|\mathcal{F}|} c(x)$, we have $-\epsilon c(x) \leq -c_F(x) + 4c_F(x^*) + 2c_S(x^*) + 2c_S(x)$, as required. □

7.5 The Performance Guarantee

From the previous results we can conclude:

Theorem 7.6 *Let $0 < \epsilon \le 1$, and let x, x^* be feasible solutions to a given instance, and let $c^x(t,\delta) > -\frac{\epsilon}{8|\mathcal{F}|}c(x)$ for $t \in \mathcal{F}$ and $\delta \in \mathbb{R}_+$ and $c^x(A,\delta) > -\frac{\epsilon}{8|\mathcal{F}|}c(x)$ for all stars and comets A and $\delta \in \Delta_A^x$. Then $c(x) \le (1+\epsilon)(7c_F(x^*) + 5c_S(x^*))$.*

Proof: By Lemma 7.3 we have $c_S(x) \le c_F(x^*) + c_S(x^*) + \frac{\epsilon}{8}c(x)$, and by Lemma 7.5 we have $c_F(x) \le 4c_F(x^*) + 2c_S(x^*) + 2c_S(x) + \frac{\epsilon}{8}c(x)$. Hence $c(x) = c_F(x) + c_S(x) \le 7c_F(x^*) + 5c_S(x^*) + \frac{\epsilon}{2}c(x)$, implying $c(x) \le (1+\epsilon)(7c_F(x^*) + 5c_S(x^*))$. □

We improve the approximation guarantee of $7+\epsilon$ by a standard scaling technique (cf. Proposition 4.9) and obtain the main result of this section:

Theorem 7.7 *(Vygen [2005]) For every $\epsilon > 0$ there is a polynomial-time $(\frac{\sqrt{41}+7}{2}+\epsilon)$-approximation algorithm for the* Universal Facility Location Problem.

Proof: We may assume $\epsilon \le \frac{1}{3}$. Let $\beta := \frac{\sqrt{41}-5}{2} \approx 0.7016$. Set $f_i'(z) := \beta f_i(z)$ for all $z \in \mathbb{R}_+$ and $i \in \mathcal{F}$, and consider the modified instance.

Let x be any initial feasible solution. Apply the algorithms of Lemma 7.2 and Lemma 7.4 with $\epsilon' := \frac{\epsilon}{16|\mathcal{F}|}$. They either find an Add or Pivot operation that reduces the cost of the current solution x by at least $\frac{\epsilon}{16|\mathcal{F}|}c(x)$, or they conclude that the prerequisites of Theorem 7.6 are fulfilled. If x is the resulting solution, c_F' and c_F denote the facility cost of the modified and original instance, respectively, and x^* is any feasible solution, then $c_F(x)+c_S(x) = \frac{1}{\beta}c_F'(x)+c_S(x) \le \frac{1}{\beta}(6c_F'(x^*)+4c_S(x^*)+\frac{3\epsilon}{8}c(x))+c_F'(x^*)+c_S(x^*)+\frac{\epsilon}{8}c(x) \le (6+\beta)c_F(x^*)+(1+\frac{4}{\beta})c_S(x^*)+\frac{3\epsilon}{4}c(x) = (6+\beta)(c_F(x^*)+c_S(x^*)) + \frac{3\epsilon}{4}c(x)$. Hence $c(x) \le (1+\epsilon)(6+\beta)c(x^*)$.

Each iteration reduces the cost by a factor of at least $\frac{1}{1-\frac{\epsilon}{16|\mathcal{F}|}}$, hence after $\frac{1}{-\log(1-\frac{\epsilon}{16|\mathcal{F}|})} < \frac{16|\mathcal{F}|}{\epsilon}$ iterations the cost reduces at least by a factor of 2 (note that $\log x < x - 1$ for $0 < x < 1$). This implies a weakly polynomial running time. □

In particular, as $\frac{\sqrt{41}+7}{2} < 6.702$, we have a 6.702-approximation. This is the best known approximation guarantee known today. We do not know whether the performance guarantee is tight. It is an open problem to improve it (maybe by using our new Pivot operation with other forests than stars and comets), and to obtain a strongly polynomial approximation algorithm.

For the Capacitated Facility Location Problem, the performance guarantee can be improved to 5.83, as Zhang, Chen and Ye [2004] showed. In this case an additional operation, which corresponds to a Pivot on forests that are the disjoint union of two stars, can be implemented in polynomial time, although there is an exponential number of such forests.

With this operation it is quite easy to modify the proof of Theorem 7.5 and open each $t \in T$ three instead of four times: the two Pivot operations in Case 1 can be replaced by one (with centers w_k and t), and the three Pivot operations in Case 2 and 3 can be replaced by two, the first one with centers at w_k and t (moving demand away from w_k and from the s_i with i even to t), and the second one with centers s_1 and t (moving demand away from s_1 and from the s_i with $i \geq 3$ odd to t). The rest of the proof is analogous. We omit the details and refer to Zhang, Chen and Ye [2004].

8 Conclusions

In many practical applications the problems do not occur in the simple form discussed here, but with additional constraints or different objectives. Extensions that have received much interest recently include multilevel facility location problems (Aardal, Chudak and Shmoys [1999], Bumb and Kern [2001], Ageev [2002], Ageev, Ye and Zhang [2005], Plaxton [2003], Zhang [2004]), hierarchical cache placement problems (Guha, Meyerson and Munagala [2000]), the single-sink buy-at-bulk network design problem (Guha, Meyerson and Munagala [2001], Talwar [2002], Goel and Estrin [2003], Gupta, Kumar and Roughgarden [2003]) and similar problems (Chekuri, Khanna and Naor [2001], Ravi and Sinha [2002], Swamy and Kumar [2004], Maßberg and Vygen [2005]). Most of these are motivated from various practical applications.

In many cases ideas developped for the basic problems (on which we concentrated in this paper) proved fruitful also for more complicated variants. In fact, the main techniques that we used for the discrete facility location problems, LP rounding, greedy and primal-dual algorithms, and local search, have been applied successfully to many other combinatorial optimization problems (see Korte and Vygen [2000]). However, the only technique that is currently known to yield constant-factor approximations for capacitated facility location problems, local search, is still a pure heuristic – without any reasonable performance guarantee – for most other problems (cf. Aarts and Lenstra [2003]). Nevertheless it is widely applied in practice.

It is fascinating that almost all results in this paper are less than ten years old, although the problems have been studied long before. Research is still very active in this area. Hopefully, this survey can help to stimulate further interesting results.

Acknowledgement: This paper has arisen from lecture notes of a course given at the University of Bonn in the winter term 2004/2005. I thank the students who attended this course for many helpful remarks.

References

Aardal, K., Chudak, F.A., and Shmoys, D.B. [1999]: A 3-approximation algorithm for the k-level uncapacitated facility location problem. Information Processing Letters 72 (1999), 161–167

Aarts, E.L., and Lenstra, J.K., eds. [2003]: Local Search in Combinatorial Optimization. Princeton University Press, Princeton 2003

Ageev, A.A. [2002]: Improved approximation algorithms for multilevel facility location problems. Operations Research Letters 30 (2002), 327–332

Ageev, A., Ye, Y., and Zhang, J. [2005]: Improved combinatorial approximation algorithms for the k-level facility location problem. SIAM Journal on Discrete Mathematics 18 (2005), 207–217

Archer, A., Rajagopalan, R., and Shmoys, D.B. [2003]: Lagrangian relaxation for the k-median problem: new insights and continuity properties. Algorithms – Proceedings of the 11th Annual European Symposium on Algorithms, Springer, Berlin 2003, pp. 31-42.

Arora, S., Raghavan, P., and Rao, S. [1998]: Approximation schemes for Euclidean k-medians and related problems. Proceedings of the 30th Annual ACM Symposium on Theory of Computing (1998), 106–113

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., and Pandit, V. [2004]: Local search heuristics for k-median and facility location problems. SIAM Journal on Computing 33 (2004), 544–562

Bajaj, C.L. [1988]: The algebraic degree of geometric optimization problems. Discrete and Computational Geometry 3 (1988), 177–191

Balinski, M.L. [1965]: Integer programming: methods, uses, computation. Management Science 12 (1965), 253–313

Balinski, M.L., and Wolfe, P. [1963]: On Benders decomposition and a plant location problem. Working paper ARO-27. Mathematica, Princeton 1963

Bumb, A.F., and Kern, W. [2001]: A simple dual ascent algorithm for the multilevel facility location problem. In: Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques; LNCS 2129 (M. Goemans, K. Jansen, J.D.P. Rolim, L. Trevisan, eds.), Spinger, Berlin 2001

Charikar, M., Guha, S., Tardos, É., and Shmoys, D.B. [2002]: A constant-factor approximation algorithm for the k-median problem. Journal of Computer and System Sciences 65 (2002), 129–149

Charikar, M., and Guha, S. [1999]: Improved combinatorial algorithms for the facility location and k-median problems. Proceedings of the 40th Annual IEEE Conference on Foundations of Computer Science (1999), 378–388

Charikar, M., Guha, S., Tardos, É., and Shmoys, D.B. [2002]: A constant-factor approximation algorithm for the k-median problem. Journal of Computer and System Sciences 65 (2002), 129–149

Chekuri, C., Khanna, S., and Naor, J. [2001]: A deterministic algorithm for the cost-distance problem. Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (2001), 232–233

Chudak, F.A., and Shmoys, D.B. [1998]: Improved approximation algorithms for uncapacitated facility location. In: Integer Programming and Combinatorial Optimization; Proceedings of the 6th International IPCO Conference; LNCS 1412 (R.E. Bixby, E.A. Boyd, R.Z. Rios-Mercado, eds.) Springer, Berlin 1998, pp. 180-194; to appear in SIAM Journal on Computing

Chvátal, V. [1979]: A greedy heuristic for the set cover problem. Mathematics of Operations Research 4 (1979), 233–235

Cornuéjols, G., Nemhauser, G.L., and Wolsey, L.A. [1990]: The uncapacitated facility location problem. In: Discrete Location Theory (P. Mirchandani, R. Francis, eds.), Wiley, New York 1990, pp. 119–171

Drezner, Z., Klamroth, K., Schöbel, A., and Wesolowsky, G.O. [2002]: The Weber problem. In: Facility Location: Applications and Theory (Z. Drezner and H.W. Hamacher, eds.), Springer, Berlin 2002, pp. 1–36

Feige, U. [1998]: A threshold of $\ln n$ for the approximating set cover. Journal of the ACM 45 (1998), 634–652

Feige, U., Lovász, L., and Tetali, P. [2004]: Approximating min sum set cover. Algorithmica 40 (2004), 219–234

Garg, N., Khandekar, R., and Pandit, V. [2005]: Improved approximation for universal facility location. Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (2005), 959–960

Goel, A., and Estrin, D. [2003]: Simultaneous optimization for concave costs: single sink aggegration or single source buy-at-bulk. Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (2003), 499–507

Guha, S., and Khuller, S. [1999]: Greedy strikes back: improved facility location algorithms. Journal of Algorithms 31 (1999), 228–248

Guha, S., Meyerson, A., and Munagala, K. [2000]: Hierarchical placement and network design problems. Proceedings of 41st IEEE Symposium on Foundations of Computer Science, 2000, 603–612

Guha, S., Meyerson, A., and Munagala, K. [2001]: A constant factor approximation for the single sink edge installation problem. Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (2001), 383–388

Gupta, A., Kumar, A., and Roughgarden, T. [2003]: Simpler and better approximation algorithms for network design. Proceedings of the 35nd Annual ACM Symposium on the Theory of Computing (2003), 365–372

Hardy, G.H., Littlewood, J.E., and Pólya, G. [1964]: Inequalities. Cambridge University Press, Cambridge 1964

Hochbaum, D.S. [1982]: Heuristics for the fixed cost median problem. Mathematical Programming 22 (1982), 148–162

Jain, K., Mahdian, M., Markakis, E., Saberi, A., and Vazirani, V.V. [2003]: Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. Journal of the ACM 50 (2003), 795–824

Jain, K., and Vazirani, V.V. [2001]: Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. Journal of the ACM 48 (2001), 274–296

Kolliopoulos, S.G., and Rao, S. [1999]: A nearly linear-time approximation scheme for the Euclidean k-median problem. Algorithms – Proceedings of the 7th European Symposium on Algorithms (ESA); LNCS 1643 (J. Nesetril, ed.), Springer, Berlin 1999, pp. 378–389

Korte, B., and Vygen, J. [2000]: Combinatorial Optimization: Theory and Algorithms. Springer, Berlin 2000 (Second edition 2002)

Korupolu, M., Plaxton, C., and Rajaraman, R. [2000]: Analysis of a local search heuristic for facility location problems. Journal of Algorithms 37 (2000), 146–188

Kuehn, A.A., and Hamburger, M.J. [1963]: A heuristic program for locating warehouses. Management Science 9 (1963), 643–666

Kuhn, H.W. [1973]: A note on Fermat's problem. Mathematical Programming 4 (1973), 98–107

Levi, R., Shmoys, D.B., and Swamy, C. [2004]: LP-based approximation algorithms for capacitated facility location. In: Integer Programming and Combinatorial Optimization; Proceedings of the 10th International IPCO Conference; LNCS 3064 (G. Nemhauser, D. Bienstock, eds.), Springer, Berlin 2004, pp. 206–218

Mahdian, M., and Pál, M. [2003]: Universal facility location. In: Algorithms – Proceedings of the 11th European Symposium on Algorithms (ESA); LNCS 2832 (G. di Battista, U. Zwick, eds.), Springer, Berlin 2003, pp. 409–421

Mahdian, M., Ye, Y., and Zhang, J. [2002]: Improved approximation algorithms for metric facility location problems. Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization; LNCS 2462 (K. Jansen, S. Leonardi, V. Vazirani, eds.) Springer, Berlin 2002, pp. 229–242

Mahdian, M., Ye, Y., and Zhang, J. [2003]: A 2-approximation algorithm for the soft-capacitated facility location problem. Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques; LNCS 2764 (S. Arora, K. Jansen, J.D.P. Rolim, A. Sahai, eds.), Springer, Berlin 2003, pp. 129–140

Manne, A.S. [1964]: Plant location under economies-of-scale-decentralization and computation. Management Science 11 (1964), 213–235

Maßberg, J., and Vygen, J. [2005]: Approximation algorithms for network design and facility location with service capacities. Report No. 05949-OR, Research Institute for Discrete Mathematics, University of Bonn, 2005

Oriln, J.B., Punnen, A.P., and Schulz, A.S. [2004]: Approximate local search in combinatorial optimization. SIAM Journal on Computing 33 (2004), 1201–1214

Pál, M., Tardos, É., and Wexler, T. [2001]: Facility location with hard capacities. Proceedings of the 42nd Annual IEEE Symposium on the Foundations of Computer Science (2001), 329–338

Plaxton, C.G. [2003]: Approximation algorithms for hierarchical location problems. Proceedings of the 35nd Annual ACM Symposium on the Theory of Computing (2003), 40–49

Rautenbach, D., Struzyna, M., Szegedy, C., and Vygen, J. [2004]: Weiszfeld's algorithm revisited once again. Report No. 04946-OR, Research Institute for Discrete Mathematics, University of Bonn, 2004

Ravi, R. and Sinha, A. [2002]: Integrated logistics: Approximation algorithms combining facility location and network design. In: Integer Programming and Combinatorial Optimization; Proceedings of the 9th International IPCO Conference; LNCS 2337 (W.J. Cook, A.S. Schulz, eds.), Springer, Berlin 2002, pp. 212–229

Raz, R., and Safra, S. [1997]: A sub constant error probability low degree test, and a sub constant error probability *PCP* characterization of *NP*. Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (1997), 475–484

Shmoys, D.B. [2000]: Approximation algorithms for facility location problems. Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization; LNCS 1913 (K. Jansen, S. Khuller, eds.) Springer, Berlin 2000, pp. 27–33

Shmoys, D.B., Tardos, É., and Aardal, K. [1997]: Approximation algorithms for facility location problems. Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (1997), 265–274

Stollsteimer, J.F. [1963]: A working model for plant numbers and locations. Journal of Farm Economics 45 (1963), 631–645

Struzyna, M. [2004]: Analytisches Placement im VLSI-Design. Diploma thesis (in German), University of Bonn, 2004

Sviridenko, M. [2002]: An improved approximation algorithm for the metric uncapacitated facility location problem. In: Integer Programming and Combinatorial Optimization; Proceedings of the 10th International IPCO Conference; LNCS 2337 (W. Cook, A. Schulz, eds.), Springer, Berlin 2002, pp. 240–257

Swamy, C., and Kumar, A. [2004]: Primal-dual algorithms for connected facility location problems. Algorithmica 40 (2004), 245–269

Szegedy, C. [2005]: Some Applications of the Combinatorial Laplacian. PhD thesis, University of Bonn, 2005

Talwar, K. [2002]: The single sink buy-at-bulk LP has constant integrality gap. In: Integer Programming and Combinatorial Optimization; Proceedings of the 9th International IPCO Conference; LNCS 2337 (W.J. Cook, A.S. Schulz, eds.), Springer, Berlin 2002, pp. 475–486

Vardi, Y., and Zhang, C.-H. [2001]: A modified Weiszfeld algorithm for the Fermat-Weber problem. Mathematical Programming A 90 (2001), 559–566

Vygen, J. [2005]: From stars to comets: improved local search for universal facility location. Report No. 05947-OR, Research Institute for Discrete Mathematics, University of Bonn, 2005

Weiszfeld, E. [1937]: Sur le point pour lequel la somme des distances de n points donnes est minimum. Tohoku Mathematical Journal 43 (1937), 355–386

Zhang, J. [2004]: Approximating the two-level facility location problem via a quasi-greedy approach. Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (2004), 808–817

Zhang, J., Chen, B., and Ye, Y. [2004]: Multi-exchange local search algorithm for the capacitated facility location problem. In: Integer Programming and Combinatorial Optimization; Proceedings of the 10th International IPCO Conference; LNCS 3064 (G. Nemhauser, D. Bienstock, eds.), Springer, Berlin 2004, pp. 219–233