Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

Adapting the Chudak-Shmoys approximation algorithm to the k-level uncapacitated facility location problem

(Nederlandse titel: Het Chudak-Shmoys benaderingsalgoritme aangepast voor het k-niveau ongecapaciteerde facility location probleem)

Verslag ten behoeve van het
Delft Institute for Applied Mathematics
als onderdeel ter verkrijging

van de graad van

BACHELOR OF SCIENCE
in
TECHNISCHE WISKUNDE

door

Judith Mulder

Delft, Nederland
Augustus 2011

BSc verslag TECHNISCHE WISKUNDE

"Adapting the Chudak-Shmoys approximation algorithm to the k-level uncapacitated facility location problem "

(Nederlandse titel: "Het Chudak-Shmoys benaderingsalgoritme aangepast voor het k-niveau ongecapaciteerde facility location probleem")

Judith Mulder

Technische Universiteit Delft

Begeleider

Prof.dr.ir. K.I. Aardal

Overige commissieleden

Dr.ir. R.J. Fokkink

Dr. J.G. Spandaw

Augustus, 2011

Delft

Abstract

Chudak and Shmoys have proposed an $(1 + 2/e)$-approximation algorithm to the 1-level uncapacitated facility location problem. In this thesis, this approximation algorithm is first extended to the 2-level problem. We prove that under a specific assumption on the structure of the solution of the LP-relaxation, a solution of the 2-level problem can be transformed to an equivalent solution of the 1-level problem. The assumption made is that all positive values that occur in a connected component are equal. Then, the algorithm of Chudak and Shmoys can be applied on the new obtained solution. Thereafter, we show in a similar way that a valid extension of the Chudak and Shmoys to the k-level uncapacitated facility location problem exists under this assumption.

For the 1, 2 and 3-level uncapacitated facility location problem, 10,000 small and large problem instances are generated at random and the LP-relaxation is solved. The percentage of fractional solutions that satisfy the assumption made in this thesis decreases when the size of the problem instances increases. However, all the solution to these problem instances have a structure in which all positive values in a connected component have the same denominator.

Table of contents

Chapter 1

Introduction

The uncapacitated facility location problem is one of the most studied problems in the field of operations research (Mirchandani and Francis (1990)). In the uncapacitated facility location problem it has to be determined at which locations facilities should be opened to serve a set of given clients. The locations i at which facilities can be built, and the cost of building a facility at a given location, f_i, are also given. Furthermore, each client j has to be assigned to one facility. When client j is assigned to facility i, a cost of c_{ij} is incurred. The objective of the uncapacitated facility location problem is now to find an allocation in which the costs are minimized.

1.1 Formulation of the 1-level uncapacitated facility location problem

Let $\mathcal{D}$ be the set of demand points and $\mathcal{F}$ the set of all potential facility locations. Furthermore, let $\mathcal{N} = \mathcal{F} \cup \mathcal{D}, n = |\mathcal{N}|$, where it is assumed that the sets $\mathcal{F}$ and $\mathcal{D}$ are disjoint. The cost of setting up a facility at location i is f_i. c_{ij} is the cost of shipping between points $i, j \in \mathcal{N}$. In practice, possible facility locations may be at places where a client is located. Then, a dummy possible facility location is introduced at the same place as the client is located and the service cost between this client and the dummy location equals 0. In this way the sets of possible facility locations and clients can always be made disjoint. It is assumed that:

- $f_i > 0$ for each $i \in \mathcal{F}$;
- $c_{ij} \geq 0$ for each $i, j \in \mathcal{N}$;
- $c_{ij} = c_{ji}$ for each $i, j \in \mathcal{N}$ (service costs are symmetric);
- $c_{ik} \leq c_{ij} + c_{jk}$ for each $i, j, k \in \mathcal{N}$ (service costs satisfy the triangle inequality).

Notice that the costs are only assumed to be positive (fixed facility cost) or nonnegative (service cost), but that they do not have to be integer. Problems in which the triangle inequality is satisfied are called metric problems. Thus, the considered facility location problem in this section is the *metric 1-level uncapacitated facility location problem.*

Let y_i be equal to 1 if facility $i \in \mathcal{F}$ is open and 0 otherwise. Furthermore, let x_{ij} be equal to 1 if client j is assigned to facility i and 0 otherwise. The integer programming formulation of the

1-level uncapacitated facility location problem is now given by:

$$z^*_{IP} = \min \sum_{i \in \mathcal{F}} f_i y_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{D}} c_{ij} x_{ij} \tag{1.1}$$

subject to

$$\sum_{i \in \mathcal{F}} x_{ij} = 1, \qquad \forall j \in \mathcal{D} \tag{1.2}$$

$$x_{ij} \leq y_i, \qquad \forall i \in \mathcal{F}, \quad \forall j \in \mathcal{D} \tag{1.3}$$

$$x_{ij} \in \mathbb{B}, \qquad \forall i \in \mathcal{F}, \quad \forall j \in \mathcal{D} \tag{1.4}$$

$$y_i \in \mathbb{B}, \qquad \forall i \in \mathcal{F} \qquad . \tag{1.5}$$

In this formulation, the objective function (1.1) is to minimize the total cost needed to serve the demand of all clients. The total cost can be divided in the facility cost and the service cost. The facility cost consists of all costs related to the opening of facilities at certain locations and the service cost consists of all costs related to the transportation of the demand from the open facilities to the clients. Constraints (1.2) ensure that the total demand of each client $j \in \mathcal{D}$ is satisfied by exactly one facility location. Furthermore, constraints (1.3) make sure that clients can only be served from open facility locations. Finally, constraints (1.4) and (1.5) ensure that the variables of the problem are binary, such that a facility location is fully opened or not and clients are fully allocated to a facility location or not.

1.2 Extension to the 2-level uncapacitated facility location problem

The 2-level uncapacitated facility location problem is a natural extension of the 1-level uncapacitated facility location problem. The formulation of the 2-level facility location problem can be derived from the formulation of the k-level facility location problem given in Aardal et al. (1999). The following formulation is then obtained. Let $\mathcal{D}$ be the set of demand points and $\mathcal{F} = \mathcal{F}^1 \cup \mathcal{F}^2$ the set of all potential facility locations. F^l is the set of all possible locations for the facilities of level $l, l = 1, 2$. It is assumed that the sets $\mathcal{F}^1$ and $\mathcal{F}^2$ are disjoint. Furthermore, let $\mathcal{N} = \mathcal{F} \cup \mathcal{D}, n = |\mathcal{N}|$, where it is assumed that the sets $\mathcal{F}$ and $\mathcal{D}$ are disjoint. The cost of setting up a facility at location i is f_i. c_{qr} is the cost of shipping between points $q, r \in \mathcal{N}$. Again, the sets of possible facility locations at the first and second level and the sets of possible facility locations and clients can always be made disjoint by introducing dummy locations as is explained in the previous section. It is assumed that:

- $f_i > 0$ for each $i \in \mathcal{F}$;
- $c_{qr} \geq 0$ for each $q, r \in \mathcal{N}$;
- $c_{qr} = c_{rq}$ for each $q, r \in \mathcal{N}$ (service costs are symmetric);
- $c_{qs} \leq c_{qr} + c_{rs}$ for each $q, r, s \in \mathcal{N}$ (service costs satisfy the triangle inequality).

A path $p \in \mathcal{P}$ is defined as a sequence of two facilities (i_1, i_2), with $i_1 \in \mathcal{F}^1$ and $i_2 \in \mathcal{F}^2$. Each client $j \in \mathcal{D}$ must be assigned to exactly one path $p \in \mathcal{P}$. The total service cost incurred by assigning client j to path (i_1, i_2) is equal to $c_{pj} = c_{i_1,i_2} + c_{i_2,j}$.

Let y_{i_l} be equal to 1 if facility $i_l \in \mathcal{F}^l$ is open at level $l, l = 1, 2$ and 0 otherwise. Furthermore, let x_{pj} be equal to 1 if client j is assigned to path p and 0 otherwise. The integer programming formulation of the 2-level uncapacitated facility location problem is now given by:

$$z^*_{IP} = \min \sum_{l=1}^{2} \sum_{i_l \in \mathcal{F}^l} f_{i_l} y_{i_l} + \sum_{p \in \mathcal{P}} \sum_{j \in \mathcal{D}} c_{pj} x_{pj} \tag{1.6}$$

subject to

$$\sum_{p \in \mathcal{P}} x_{pj} = 1, \qquad \forall j \in \mathcal{D} \tag{1.7}$$

$$\sum_{p: p \ni i_l} x_{pj} - y_{i_l} \leq 0, \qquad \forall i_l \in \mathcal{F}^l, \quad l = 1, 2, \quad \forall j \in \mathcal{D} \tag{1.8}$$

$$x_{pj} \in \mathbb{B}, \qquad \forall p \in \mathcal{P}, \qquad \forall j \in \mathcal{D} \tag{1.9}$$

$$y_{i_l} \in \mathbb{B}, \qquad \forall i_l \in \mathcal{F}^l, \quad l = 1, 2. \tag{1.10}$$

Again, the objective (1.6) of the model is to minimize the total cost, where the cost consists of facility and service cost. The costs can be calculated in a similar way as in the 1-level problem, only both levels of facilities have to be considered now and service costs are also incurred by transporting the demand between two facilities of different levels. Furthermore, the constraints of the 2-level problem are very similar to those of the 1-level problem. Constraints (1.7) ensure that each client is served by exactly one path, constraints (1.8) make sure that both facilities of each used path are opened and constraints (1.9) and (1.10) ensure that the variables are binary.

1.3 Complexity

The uncapacitated facility location problem is NP-hard, which means that no algorithm exists that is guaranteed to find the optimal solution of the problem in polynomial time, unless $P = NP$. Therefore, it is useful to develop an approximation algorithm that quickly finds high-quality feasible solutions (Hochbaum (1997)). A ρ-approximation algorithm for a minimization problem is a polynomial-time algorithm that is guaranteed to find a feasible solution to the considered problem with an objective value that is within a factor ρ of the optimum value. We refer to ρ as the *approximation guarantee.*

1.4 LP-relaxation and duality

Here we give two definitions that are used frequently in the subsequent chapters. For more details on these topics we refer to Chvátal (1983) and Wolsey (1998).

Consider the following integer programming formulation:

$$z^*_{IP} = \min \mathbf{c}^T \mathbf{x}$$

subject to

$$
\begin{aligned}
A\mathbf{x} \geq \ & \mathbf{b} \\
\mathbf{x} \geq \ & \mathbf{0} \\
\mathbf{x} \in & \mathbb{Z}^n
\end{aligned}
$$

In this formulation $\mathbf{c}$ and $\mathbf{x}$ are $n \times 1$ vectors, $\mathbf{b}$ and $\mathbf{y}$ are $m \times 1$ vectors and A is an $m \times n$ matrix. The linear programming relaxation (LP-relaxation) of the original integer formulation can be obtained by relaxing the integrality constraints $\mathbf{x} \in \mathbb{Z}^n$. The linear programming relaxation is then given by:

$$z^*_{LP} = \min \mathbf{c}^T\mathbf{x}$$

subject to

$$
\begin{aligned}
A\mathbf{x} \geq & \mathbf{b} \\
\mathbf{x} \geq & \mathbf{0}
\end{aligned}
$$

The optimal value of the objective function z^*_{LP} satisfies $z^*_{LP} \leq z^*_{IP}$. This problem in original form is called the *primal* problem. The *dual* of the problem is then given by:

$$z^*_D = \max \mathbf{b}^T\mathbf{y}$$

subject to

$$
\begin{aligned}
A^T\mathbf{y} \leq & \mathbf{c} \\
\mathbf{y} \geq & 0
\end{aligned}
$$

Now, we can introduce the duality theorem:

Theorem 1.1 *If the primal problem has an optimal solution* $\mathbf{x}^*$, *then the dual problem has an optimal solution* y^* *such that*

$$z^*_{IP} = \mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^* = z^*_D.$$

1.5 Problem definition

In this thesis, approximation algorithms for the uncapacitated facility location problem are considered. As will be seen in the next chapter, much research on approximation algorithms for the 1-level problem has been performed. However, for higher level facility location problems less approximation algorithms are known. Furthermore, the current algorithms for the 2-level problem seem too simplistic (see for example Aardal et al. (1999)) or decompose the problem into a sequence of 1-level problems (see for example Zhang (2006)). Therefore, in this thesis we try to develop a more elaborate algorithm that works directly on the 2-level problem. In Chudak and Shmoys (2003) a $(1 + 2/e)$-approximation algorithm for the metric 1-level uncapacitated facility location problem is developed. Furthermore, some useful properties are derived and proved. In this thesis we investigate whether we can extend the algorithm of Chudak and Shmoys (2003), possibly under some assumptions, to the multi-level uncapacitated facility location problem. In particular we show that the Chudak-Shmoys algorithm can be extended to the k-level uncapacitated facility location problem, when all variables x_{pj}, which are larger than 0 in the optimal LP-solution, are equal.

1.6 Structure

The remaining chapters of this thesis are organized in the following way. Chapter 2 gives an review on the most important literature concerning approximation algorithms of the uncapacitated facility location problem. Thereafter, the $(1+2/e)$-approximation algorithm for the 1-level uncapacitated facility location problem proposed in Chudak and Shmoys (2003) is discussed in Chapter 3. In Chapter 4, an extension of the $(1+2/e)$-approximation algorithm to the 2-level problem is discussed. In this chapter, we first discuss an extension to the 2-level uncapacitated facility location problem under different assumptions on the structure of the optimal solutions of the LP-relaxations. Thereafter, the extension to higher level problems is discussed. Next, some computational results are given in Chapter 5. Thereafter, in Chapter 6 some conclusions will be formulated. Finally, in Chapter 7 a discussion will be given on the extended approximation algorithm.

Chapter 2

Literature review

In this chapter an overview of the most important literature on approximation algorithms for the uncapacitated facility location problem is given. First, literature on the one-level facility location problem is discussed. Thereafter, also literature on higher level problems is reviewed.

2.1 The 1-level uncapacitated facility location problem

Hochbaum (1982) provides heuristic algorithms with approximation guarantees for three important hard problems. One of the considered problems is the discrete fixed cost median problem, which is also known as the simple plant location problem or 1-level uncapacitated facility location problem. The author uses the set covering problem to obtain a $\log |\mathcal{D}|$-approximation algorithm for the one-level facility location problem, with $|\mathcal{D}|$ the number of clients. The service costs do not have to satisfy the triangle inequality for this algorithm.

In the algorithm described by Hochbaum (1982), the approximation guarantee depends on the number of clients included in the problem. If, however, the connection costs are required to satisfy the triangle inequality, then constant-factor approximation algorithms have been found. In Shmoys et al. (1997), the first constant-factor approximation algorithm for the metric uncapacitated facility location problem is presented. This algorithm is based on solving the linear programming relaxation of the integer uncapacitated facility location problem and rounding the obtained fractional solution to integer values. The algorithm consists of two steps. In the first step a filtering technique is used to obtain a new fractional solution. The filtering technique is used to ensure that the new solution satisfies certain requirements that are useful in rounding the solution. In this new solution, a client j is only allocated to a (partially opened) facility i if the service cost c_{ij} is not too high. In the second step of the algorithm, the fractional solution obtained in the first step will then be rounded to a near-optimal integer solution. First, Shmoys et al. (1997) describe a 4-approximation algorithm for the problem. Then, the filtering technique is improved to obtain an algorithm with a better performance guarantee. This new algorithm has an approximation guarantee of 3.16 for the one-level metric uncapacitated facility location problem.

After the publication of the 3.16-approximation algorithm in Shmoys et al. (1997), many improvements have been made, see for example Chudak (1998), Charikar and Guha (1999), Sviridenko (2002). In Chudak and Shmoys (2003), first, again, a 4-approximation algorithm for the one-level uncapacitated facility location problem is described. Next, this algorithm is improved to a $(1+2/e) \approx 1.736$-approximation algorithm. This algorithm will be discussed in more detail in Chapter 3. Thereafter again numerous improvements are made (examples are Mahdian et al. (2006), Charikar and Guha (2005), Byrka and Aardal (2010)).

The best known approximation algorithm for the metric uncapacitated facility location problem is described in Li (2011). The approximation guarantee is obtained by combining two bifactor approximation algorithms. A bifactor approximation algorithm for the uncapacitated facility location problem is an algorithm that produces a solution for which the total cost is bounded by $\lambda_f F^* + \lambda_c C^*$, where F^* and C^* are the facility and connection cost of an optimal solution and λ_f and λ_c are the two approximation factors. Then, it is proved that the approximation guarantee of the algorithm is 1.488. This result is already very close to the lower bound of 1.463 obtained in Guha and Khuller (1998).

2.2 The k-level uncapacitated facility location problem

In Aardal et al. (1999) a 3-approximation algorithm for the k-level facility location problem is derived. In this algorithm, an optimal solution of the linear programming relaxation of the k-level problem is used in a randomized rounding procedure. In this procedure, a sequence of facilities that partially service a client is opened at random, where the probability equals the fraction of the demand served by this sequence in an optimal solution of the linear programming relaxation. This results in an algorithm with expected costs at most 3 times the optimal costs. Finally, a derandomization technique is provided that can be used to obtain a 3-approximation algorithm for the k-level uncapacitated facility location problem.

Zhang (2006) proposes an algorithm that combines a randomized rounding technique with a dual fitting technique to obtain a better approximation algorithm for the 2-level uncapacitated facility location problem. Both these techniques have been used earlier to solve facility location problems, but they are never combined before. In this way, an approximation algorithm of 1.77 is obtained.

Chapter 3

The $(1+2/e)$-approximation algorithm by Chudak and Shmoys

In Chudak and Shmoys (2003) a $(1+2/e)$-approximation algorithm for the 1-level uncapacitated facility location problem is given. The main idea of this algorithm will be described in this chapter. The definitions, lemmas, corollaries and theorems described in this chapter are all from Chudak and Shmoys. Here, the lemmas, corollaries and theorems are described and some intuition is provided. For the technical proofs we refer to Chudak and Shmoys.

First, the linear programming relaxation and its dual formulation of the 1-level uncapacitated facility location problem are given. The integer programming formulation of the problem is already given in the introduction as formulation (1.1)-(1.5). The linear programming relaxation of the 1-level facility location problem can now be obtained by relaxing the constraints that $x_{ij} \in \mathbb{B}$ for each $i \in \mathcal{F}, j \in \mathcal{D}$ and $y_i \in \mathbb{B}$ for each $i \in \mathcal{F}$. Relaxing the first constraints would result in: $0 \leq x_{ij} \leq 1$ for each $i \in \mathcal{F}, j \in \mathcal{D}$. However, constraints (1.2) already ensure that $x_{ij} \leq 1$ for each $i \in \mathcal{F}, j \in \mathcal{D}$. Therefore, the relaxed constraints will become $x_{ij} \geq 0$ for each $i \in \mathcal{F}, j \in \mathcal{D}$. Similarly, when relaxing the second constraints, we would obtain $0 \leq y_i \leq 1$ for each $i \in \mathcal{F}$. However, note that constraints (1.3) already make sure that $y_i \geq 0$ for each $i \in \mathcal{F}$, because $x_{ij} \geq 0$ for each $i \in \mathcal{F}, j \in \mathcal{D}$. Furthermore, costs are related to y_i in a linear way in the objective function with $f_i > 0$ for all $i \in \mathcal{F}$. This will ensure that the value of y_i will be as small as possible. Furthermore, y_i only occurs in constraints (1.3). The largest value that x_{ij} can take equals 1, so y_i will never have to take values larger than 1 to satify these constraints. Thus, y_i will always be less than or equal to 1 in an optimal solution, so this does not have to be required explicitly. Therefore, the constraints concerning y_i can be left out. To summarize, we obtain the following formulation for the linear programming relaxation of the 1-level problem:

$$z^*_{LP} = \min \sum_{i \in \mathcal{F}} f_i y_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{D}} c_{ij} x_{ij} \tag{3.1}$$

subject to

$$\sum_{i\in\mathcal{F}} x_{ij} = 1, \qquad \forall j \in \mathcal{D} \tag{3.2}$$

$$x_{ij} \leq y_i, \qquad \forall i \in \mathcal{F}, \quad \forall j \in \mathcal{D} \tag{3.3}$$

$$x_{ij} \geq 0, \qquad \forall i \in \mathcal{F}, \quad \forall j \in \mathcal{D} \tag{3.4}$$

$$\tag{3.5}$$

Given a feasible fractional solution $(\overline{x}, \overline{y})$, the fractional facility and service cost are defined as respectively $\overline{C}_f := \sum_{i\in\mathcal{F}} f_i \overline{y}_i$ and $\overline{C}_j := \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{D}} c_{ij}\overline{x}_{ij}$.

Let v_j and w_{ij} be the dual variables corresponding to the primal constraints (3.2) and (3.3). The dual problem corresponding to the linear programming relaxation is given by:

$$z^*_{LP} = \max \sum_{j\in\mathcal{D}} v_j \tag{3.6}$$

subject to

$$v_j - w_{ij} \leq c_{ij} \qquad \forall i \in \mathcal{F}, \quad \forall j \in \mathcal{D} \tag{3.7}$$

$$\sum_{j\in\mathcal{D}} w_{ij} \leq f_i \qquad \forall i \in \mathcal{F} \tag{3.8}$$

$$w_{ij} \geq 0 \qquad \forall i \in \mathcal{F}, \quad \forall j \in \mathcal{D}. \tag{3.9}$$

Now, introduce the following definitions:

Definition 3.1 *If $(\overline{x}, \overline{y})$ is a feasible solution to the linear programming relaxation and $j \in \mathcal{D}$ is any demand point, the* **`neighborhood`** *of j, $\mathrm{N}(j)$, is the set of facilities that fractionally service j, that is $\mathrm{N}(j) = \{i \in \mathcal{F} : \overline{x}_{ij} > 0\}$.*

Note that the following fact is a simple consequence of Definition 3.1:

Fact 3.2 *For each demand point $j \in \mathcal{D}$ it has to hold that $\sum_{i\in\mathrm{N}(j)} \overline{x}_{ij} = 1$.*

Definition 3.3 *Suppose that $(\overline{x}, \overline{y})$ is a feasible solution to the linear programming relaxation, and let $g_j \geq 0$ for each $j \in \mathcal{D}$. Then $(\overline{x}, \overline{y})$ is* **`g-close`** *if $\overline{x}_{ij} > 0$ implies that $c_{ij} \leq g_j$ $(j \in \mathcal{D},\ i \in \mathcal{F})$.*

Notice that if $(\overline{x}, \overline{y})$ is g-close and $j \in \mathcal{D}$ is a demand point, then all neighbors $\mathrm{N}(j)$ of j, are inside the ball of radius g_j centered at j. Thus, the service cost from each facility that fractionally service j to j can be bounded by g_j. In Figure 3.1, the neighborhoods of two clients j and k are shown. The clients are shown by a circle and the facility locations by a square. A line between a client and a facility location denotes that the client is (partially) serviced by the facility location in the optimal LP-solution. The bounds on the service cost are also illustrated in the figure. Facility locations i_2 and i_3 both partially service clients j and k, so these facility locations are both in the neighborhood of j and k. Client j is further partially serviced by facility location i_1, which indicates that i_1 is also in the neighborhood of j. Finally, facility locations i_4 and i_5 partially service client k and thus are part of the neighborhood of k.

Now, consider constraints (3.7) of the dual problem. Due to complementary slackness it has to hold that $v^*_j - w^*_{ij} = c_{ij}$ when $x_{ij} > 0$. Combining this with constraints (3.9) gives us the following lemma.

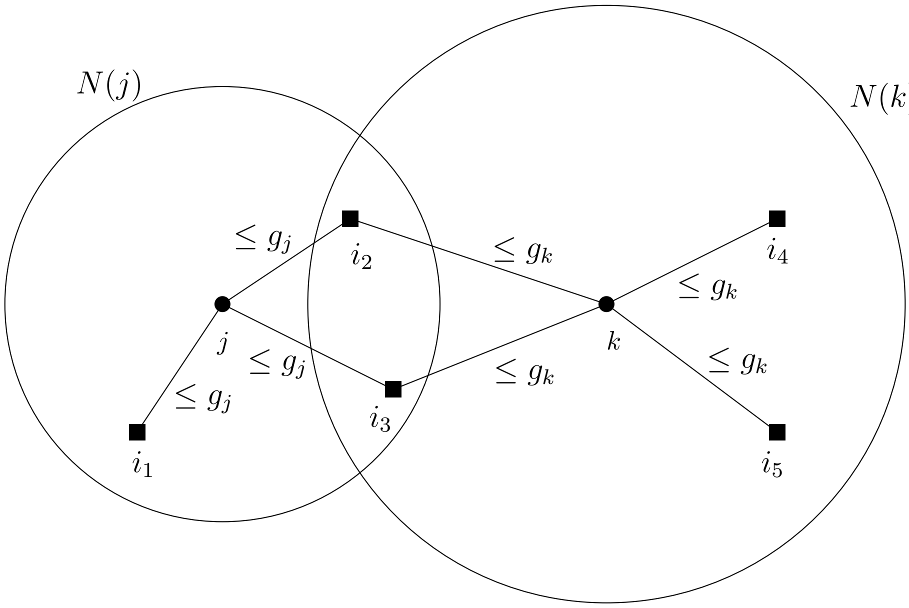

Figure 3.1: Example of the neighborhood structure of two clients j and k.

Lemma 3.4 *If (x^*, y^*) is an optimal solution to the primal linear programming relaxation and (v^*, w^*) is an optimal solution to the dual problem, then (x^*, y^*) is v^*-close.*

Definition 3.5 *A feasible solution $(\overline{x}, \overline{y})$ to the linear programming relaxation is complete if $\overline{x}_{ij} > 0$ implies that $\overline{x}_{ij} = \overline{y}_i$ for every $i \in \mathcal{F},\ j \in \mathcal{D}$.*

It can now be shown that the optimal solution (x^*, y^*) of the linear programming relaxation is "almost" complete, which means that for each demand point $j \in \mathcal{D}$ for at most one facility $i \in \mathrm{N}(j)$ it may hold that $x^*_{ij} < y^*_i$. Using this observation, the solution to the linear programming relaxation can be made complete for an equivalent instance of the problem. This is described in the following lemma.

Lemma 3.6 *Suppose that $(\overline{x}, \overline{y})$ is a feasible solution to the linear programming relaxation for a given instance of the uncapacitated facility location problem $\mathcal{I}$. Then we can find, in polynomial time, an equivalent instance $\widetilde{\mathcal{I}}$ and a complete feasible solution $(\widetilde{x}, \widetilde{y})$ to its linear programming relaxation with the same fractional facility and service costs as $(\overline{x}, \overline{y})$. The new instance $\widetilde{\mathcal{I}}$ differs only by replacing each facility location by at most $|\mathcal{D}|+1$ copies of the same location; furthermore, if $(\overline{x}, \overline{y})$ is g-close, then so is $(\widetilde{x}, \widetilde{y})$.*

Now, suppose that we have an optimal solution (x^*, y^*) to the primal LP-relaxation and an optimal solution (v^*, w^*) to the dual problem. We assume that the fractional service cost of j of this optimal solution is $\overline{C}_j = \sum_{i \in \mathcal{F}} c_{ij} x^*_{ij}$. Then, perform the following clustering algorithm:

1. Initialize the set of unallocated clients $\mathcal{S} = \mathcal{D}$ and the set of cluster centers $\mathcal{C} = \emptyset$.
2. Repeat the following as long as $\mathcal{S} \neq \emptyset$.
   (a) Choose $j_\circ \in \mathcal{S}$ with the smallest $v^*_j + \overline{C}_j$ value, where $j \in \mathcal{S}$.
   (b) Create a new cluster $\mathcal{Q}$ "centered" at $j_\circ$ and add $j_\circ$ to the set of cluster centers: $\mathcal{C} = \mathcal{C} \cup j_\circ$.
   (c) Add all unassigned clients k that share at least one neighbor with $j_\circ$ to cluster $\mathcal{Q}$,

that is $\mathcal{Q} = \{k \in \mathcal{S} : \mathrm{N}(k) \cap \mathrm{N}(j_\circ) \neq \emptyset\}$.

(d) Remove the clients allocated to cluster $\mathcal{Q}$ from the set of unallocated clients, so set $\mathcal{S} = \mathcal{S} - \mathcal{Q}$.

In Figure 3.2, a cluster centered at client $j_\circ$ is illustrated. Again, clients are denoted by circles and facility locations by squares. Lines indicate that a facility location (partially) service a client in the optimal LP-solution. From the figure, it can be seen that client $j_\circ$ is partially serviced by facility locations i_2 and i_3. These facility locations also partially service clients k_1, k_2 and k_3. Therefore, these clients are also part of cluster Q that has its center at client $j_\circ$. However, client l is only serviced by facility location i_1. Since this location is not in the neighborhood of client $j_\circ$, client l does not belong to cluster Q as can be seen in the figure.

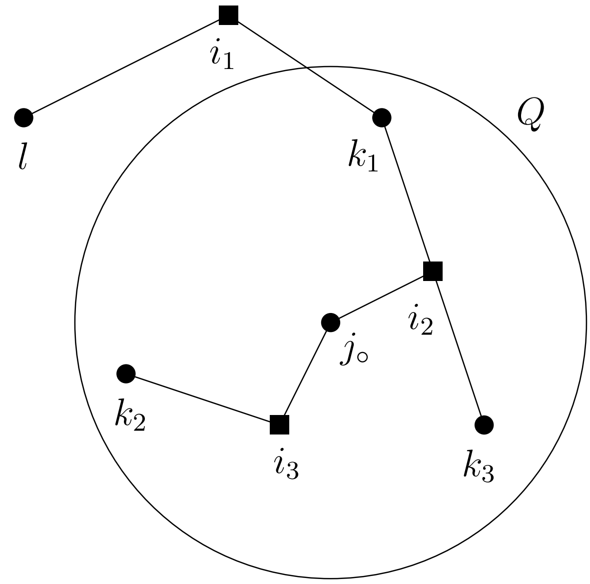

Figure 3.2: Example of a cluster centered at client $j_\circ$.

Next, we can divide the facility locations into two different groups.

Definition 3.7 *The set of* `central facility locations` $\mathcal{L}$ *is the set of facility locations that are in the neighborhood of some cluster center, that is* $\mathcal{L} = \cup_{j \in \mathcal{C}} \mathrm{N}(j)$; *the remaining set of facility locations* $\mathcal{R} = \mathcal{F} - \mathcal{L}$ *are* `noncentral facility locations`.

In Figure 3.3 the sets of central and noncentral facility locations are shown. Clients $j_\circ$ and $m_\circ$ are the centers of the two clusters in the figure. Only facility location i_4 is not in the neighborhood of one of the cluster centers. Therefore, i_4 is the only facility location in the set of noncentral facilities. All other facility locations are in the set of central facility locations.

Now, facilities can be opened and demand points allocated to the open facilities as follows. First, exactly one facility per cluster is opened in the following way. Neighboring facility $i \in \mathrm{N}(j)$ is opened at random with probability x^*_{ij}, (notice that $x^*_{ij} = y^*_i$ since the solution is complete), independently for each center $j \in \mathcal{C}$. Next, each noncentral facility $i \in \mathcal{R}$ is opened independently with probability y^*_i. Finally, each demand point $j \in \mathcal{D}$ is assigned to its closest open facility $i \in \mathcal{F}$. Using this algorithm, we can find the expected total facility and service cost as is defined in the following lemmas and corollaries.

Lemma 3.8 *For each facility location* $i \in \mathcal{F}$, *the probability that a facility at location* i *is open is* y^*_i.

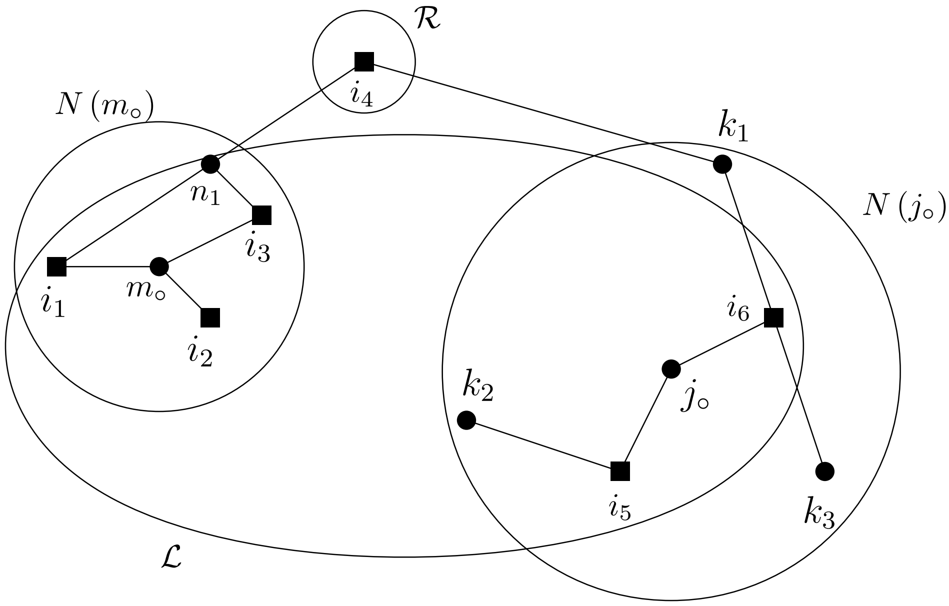

Figure 3.3: Example of the sets of central and noncentral facility locations.

Using this lemma, the expected total facility cost can be found by summing the expected facility cost per facility. This will lead to the following corollary.

Corollary 3.9 *The expected total facility cost is* $\sum_{i \in \mathcal{F}} f_i y_i^*$.

The expected service cost of a demand point can be determined in the following way. Consider a demand point $k \in \mathcal{D}$. Then, two situations can occur: at least one neighboring facility of k is opened or all neighboring facilities of k are closed. In the first case the expected service cost of k is equal to $\overline{C}_k = \sum_{i \in \mathrm{N}(k)} c_{ik} x_{ik}^*$. In the second case, we first determine the probability that all neighboring facilities of k are closed. For notational simplicity suppose that $N(k) = \{1, \ldots, d\}$. When each cluster center in $\mathcal{C}$ shares at most one neighbour with k, each neighbor $i \in N(k)$ is opened with probability $y_i^* = x_{ik}^*$ independently. Thus, the probability q that all facilities in $N(k)$ are closed is $q = \prod_{i=1}^{d} (1 - y_i^*) = \prod_{i=1}^{d} (1 - x_{ik}^*)$. Now, we can use Fact 3.2 together with the fact that $1 - x \leq e^{-x}$ for $x > 0$ to obtain:

$$q = \prod_{i=1}^{d} (1 - x_{ik}^*) \leq \prod_{i=1}^{d} e^{-x_{ik}^*} = e^{-\sum_{i=1}^{d} x_{ik}^*} = \frac{1}{e}.$$

When a cluster center in $\mathcal{C}$ can share more than one neighbor with k, the events of opening facilities in $N(k)$ are no longer independent for two neighboring facilities of k that are neighbors of the same cluster center. However, if one of the two facilities is closed, the probability that the other is opened incereases, thus the dependencies are favorable for the analysis. Thus, the probability that all neighboring facilities of k are closed is at most $\frac{1}{e}$. Assume now that demand point k belongs to the cluster centered at $j_\circ$. Since exactly one facility in each cluster is opened, $j_\circ$ always has a neighboring facility $i_\circ$ that is opened. Now, select a facility $l \in \mathrm{N}(k) \cap \mathrm{N}(j_\circ)$. The expected service cost of the open facility $i_\circ$ ($c_{i_\circ k}$) to k can now be bounded by

$$c_{i_\circ k} \leq c_{i_\circ j_\circ} + c_{j_\circ l} + c_{lk}.$$

It is known that $c_{i_\circ j_\circ} \leq v_{j_\circ}^*$, $c_{j_\circ l} \leq v_{j_\circ}^*$ and $c_{lk} \leq v_k^*$ (see Figure 3.4). Furthermore, the expected service cost between $i_\circ$ and $j_\circ$ or between $j_\circ$ and l ($c_{j_\circ l}$) is at most $\overline{C}_{j_\circ}$. Thus, the expected service cost between $i_\circ$ and k is at most

$$c_{i_\circ k} \leq v_{j_\circ}^* + \overline{C}_{j_\circ} + v_k^*.$$

Since in the clustering step the client with smallest $v_j^* + \overline{C}_j$ value is chosen, it is also known that

$$v_{j_\circ}^* + \overline{C}_{j_\circ} \leq v_k^* + \overline{C}_k.$$

Thus, the expected service cost between $i_\circ$ and k is at most

$$c_{i_\circ k} \leq v_k^* + \overline{C}_k + v_k^* = \overline{C}_k + 2v_k^*.$$

The expected service cost for demand point k can now be determined by combining the two possibilities. In the first case, an expected cost of $\overline{C}_k$ is incurred and in the second case an expected cost of $\overline{C}_k + 2v_k^*$ is incurred. Thus, always at least an expected cost of $\overline{C}_k$ is incurred. When all neigboring facilities of k are closed, which happens with probability at most $\frac{1}{e}$, an additional expected cost of $2v_k^*$ is incurred. This results in a total expected service cost for demand point k of at most $\overline{C}_k + \frac{2}{e}v_k^*$.

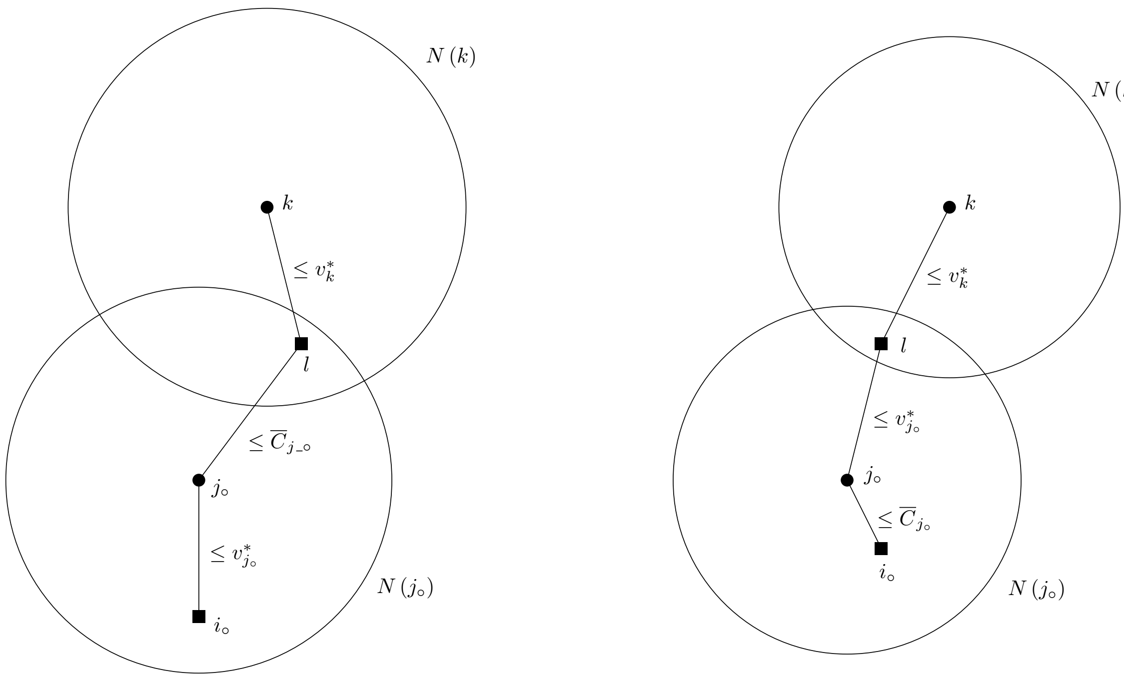

Figure 3.4: Bounding the service cost of k.

Lemma 3.10 *For each demand point $k \in \mathcal{D}$, the expected service cost of k is at most $\overline{C}_k + \frac{2}{e}v_k^*$, with $\overline{C}_k = \sum_{i \in \mathcal{F}} c_{ik} x_{ik}^*$.*

The expected total service cost can be found by summing the expected service costs per demand point.

Corollary 3.11 *The expected total service cost is at most $\sum_{k \in \mathcal{D}} \overline{C}_k + \frac{2}{e} z_{LP}^*$.*

Next, the expected total costs can be found by combining Corollaries 3.9 and 3.11.

Theorem 3.12 *There is a polynomial-time randomized algorithm that finds a feasible solution to the uncapacitated facility location problem with expected cost at most $(1 + \frac{2}{e}) z_{LP}^*$.*

Finally, a derandomization method is proposed that derandomize the algorithm in such a way that the total costs are less than or equal to the expected total costs. This derandomization method completes the proof of the following theorem:

Theorem 3.13 *There is a polynomial-time algorithm that rounds an optimal solution to the linear programming relaxation to a feasible integer solution whose value is within* $(1 + 2/e) \approx 1.736$ *of the optimal value of the linear programming relaxation.*

Chapter 4

Extension of the $(1 + 2/e)$-approximation algorithm

In Chudak and Shmoys (2003) some useful properties are derived for the 1-level uncapacitated facility location problem. A natural question is whether these properties can be extended to the 2-level (and probably even to the k-level) facility location problem.

In Chudak and Shmoys (2003), first the 4-approximation algorithm, first introduced by Shmoys et al. (1997), for the 1-level uncapacitated facility location problem is presented. Furthermore, the authors introduce some useful properties of the 1-level problem. In Aardal et al. (1999) the ideas behind the 4-approximation algorithm are used to design a 3-approximation algorithm for the k-level facility location problem, with $k \geq 1$. Here, we investigate first how to extend the properties described in Chapter 3 to the 2-level facility location problem. Thereafter, we investigate whether these properties can also be extended to higher level facility location problems.

4.1 Extension to the 2-level facility location problem

Again, we first introduce the linear programming relaxation and its dual formulation of the 2-level uncapacitated facility location problem. The integer programming formulation of the 2-level uncapacitated facility location problem is given in Chapter 1. Now, the linear programming relaxation of the problem can be obtained in a similar way as for the 1-level problem. The constraints that have to be relaxed are again $x_{pj} \in \{0,1\}$ for each $p \in \mathcal{P},\ j \in \mathcal{D}$ and $y_{i_l} \in \{0,1\}$ for each $i_l \in \mathcal{F}^l,\ l = 1,2$. When these constraints are relaxed, we allow the variables x_{pj} and y_{i_l} to take values between 0 and 1. However, as already seen for the 1-level problem in Chapter 3, constraints (4.2) will ensure that $x_{pj} \leq 1$ for each $p \in \mathcal{P},\ j \in \mathcal{D}$ and for $y_{i_l} \in \{0,1\}$, the objective function in combination with constraints (4.3) will make sure that $y_{i_l} \leq 1$ for each $i_l \in \mathcal{F}^l,\ l = 1,2$ (since the sum $\sum_{p:p \ni i_l} x_{pj}$ is bounded above by 1 by constraints (4.2)) and constraints (4.3) ensure that $y_{i_l} \geq 0$ for each $i_l \in \mathcal{F}^l,\ l = 1,2$. In this way, we obtain the following formulation:

$$z^*_{LP} = \min \sum_{l=1}^{2} \sum_{i_l \in \mathcal{F}^l} f_{i_l} y_{i_l} + \sum_{p \in \mathcal{P}} \sum_{j \in \mathcal{D}} c_{pj} x_{pj} \tag{4.1}$$

subject to

$$\sum_{p \in \mathcal{P}} x_{pj} = 1, \qquad \forall j \in \mathcal{D} \tag{4.2}$$

$$\sum_{p: p \ni i_l} x_{pj} - y_{i_l} \leq 0, \qquad \forall j \in \mathcal{D}, \quad \forall i_l \in \mathcal{F}^l, \quad l = 1, 2 \tag{4.3}$$

$$x_{pj} \geq 0, \qquad \forall p \in \mathcal{P}, \quad \forall j \in \mathcal{D} \tag{4.4}$$

$$\tag{4.5}$$

Given a feasible fractional solution $(\overline{x}, \overline{y})$, the fractional facility and service cost are defined as respectively $\overline{C}_f := \sum_{l=1}^{2} \sum_{i_l \in \mathcal{F}^l} f_{i_l} \overline{y}_{i_l}$ and $\overline{C}_j := \sum_{p \in \mathcal{P}} \sum_{j \in \mathcal{D}} c_{pj} \overline{x}_{pj}$.

Let v_j and $w_{i_l,j}$ be the dual variables corresponding to the primal constraints (4.2) and (4.3). The dual problem corresponding to the linear programming relaxation is given by

$$z^*_{LP} = \max \sum_{j \in \mathcal{D}} v_j \tag{4.6}$$

subject to

$$v_j - \sum_{i_l \in p} w_{i_l,j} \leq c_{pj} \qquad \forall p \in \mathcal{P}, \quad \forall j \in \mathcal{D} \tag{4.7}$$

$$\sum_{j \in \mathcal{D}} w_{i_l,j} \leq f_{i_l} \qquad \forall i_l \in \mathcal{F}^l, \quad l = 1, 2 \tag{4.8}$$

$$w_{i_l,j} \geq 0 \qquad \forall j \in \mathcal{D}, \quad \forall i_l \in \mathcal{F}^l, \quad l = 1, 2 \tag{4.9}$$

Before we propose an extension of the method desribed in Chudak and Shmoys (2003), an assumption on the structure of the solution of the LP-relaxation is made. Therefore, the algorithmic result will only be valid for cases in which the structure of the LP-relaxation solution corresponds to the assumed structure. A solution of the linear programming relaxation of the 2-level facility location problem consists of one or more connected components. In a *connected component*, each facility and client can be reached from every other facility and client in the same component using only paths including clients, ignoring directions, for which $x^*_{pj} > 0$ where (x^*, y^*) is the optimal solution of the LP-relaxation. On the other hand, a facility or client that lies in another connected component cannot be reached using only paths including clients for which $x^*_{pj} > 0$. In Figure 4.1 an example of a solution with three connected components A, B and C is given. In this figure, the facility locations at level 1 are denoted with a cross, the facilities at level 2 with a square and the clients with a circle. In Figure 4.2 an example of a fractional solution to a problem is given. The facilities and clients are denoted in the same way as in Figure 4.1. The solution given in Figure 4.2 consists of two connected components. All variables in the right connected component take the value 0 or 1. However, in the left connected component, fractional values occur in the solution. Note that the connections between facilities a_1 and b_1 and between a_1 and b_2 are both part of two paths that both are used with fraction $\frac{1}{2}$. Thus, these connections are used twice in the solution.

In the solution of the linear programming relaxation of the 2-level facility location problem, the x^*_{pj} variables can take different values. However, it is assumed that for all paths and clients that belong to the same connected component, the value of the corresponding x^*_{pj} variables with

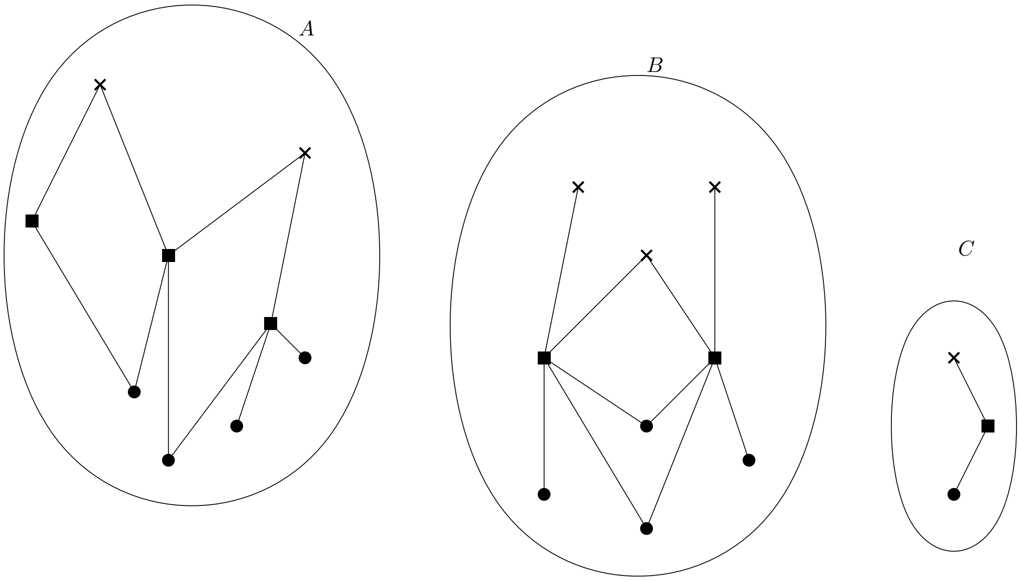

Figure 4.1: Example of a solution with three connected components.

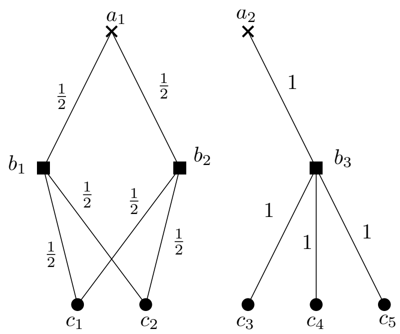

Figure 4.2: Example of a fractional solution that consists of two connected components.

$x^*_{pj} > 0$ are equal. Thus, the following assumption on the solution of the LP-relaxation of the 2-level problem is introduced:

Assumption 4.1 *If x^*_{pj} is noninteger for at least one path p and client j in a connected component, then $x^*_{pj} = \begin{cases} r \\ 0 \end{cases}$ for all $p \in \mathcal{P}$ and $j \in \mathcal{D}$ that belong to the same connected component and with $r \in \mathbb{Q}$ a constant between 0 and 1.*

The value of r is thus the same within a connected component, but can differ between components. In the computations we will investigate how realistic this assumption is. The following lemma can be used to extend the algorithm for the 1-level facility location problem proposed in Chudak and Shmoys (2003) to the 2-level problem.

Lemma 4.2 *For each connected component, the value of r is of the form $\frac{1}{b}$ with $b \in \mathbb{N}$.*

Proof: First, the LP-relaxation solution of integer connected components, or in other words connected components for which $r = 1$ are already optimal, so these solutions do not have to be changed. In this case, r is indeed of the form $\frac{1}{b}$ with $b = 1 \in \mathbb{N}$. Furthermore, assume for fractional components that $r = \frac{a}{b}$ with $a, b \in \mathbb{N},\ a \neq 1,\ \gcd(a, b) = 1$ with $\gcd(a, b)$ the greatest common divisor of a and b. The solution of the LP-relaxation has to satisfy

$$\sum_{p\in\mathcal{P}} x_{pj} = q\cdot\frac{a}{b} = 1$$

for each client j, with $q=\sum_{p\in\mathcal{P}} I_{x^*_{pj}>0}\in\mathbb{N}$ and $I_{x>0}$ the indicator function that takes the value 1 when $x>0$ and 0 otherwise. Thus, it has to hold that $qa=b$ or $q=\frac{b}{a}$. It is already known that $\gcd(a,b)=1$, so b is not a multiple of a. Therefore, $q=\frac{b}{a}\notin\mathbb{N}$ when $a\neq 1$. However, this leads to a contradiction, so a has to be equal to 1 and r is of the form $\frac{1}{b}$ with $b\in\mathbb{N}$. ∎

Next, we will introduce a lemma that states that an optimal solution of the linear programming relaxation of the 2-level uncapacitated facility location problem satisfying Assumption 4.1 can be transformed into a complete solution of the linear programming relaxation of a related 1-level uncapacitated facility location problem. Using this lemma, we can thus use the properties found for the 1-level uncapacitated facility location problem described in Chapter 3.

Lemma 4.3 *Suppose that $(\overline{x},\overline{y})$ is a feasible solution to the linear programming relaxation for a given instance of the 2-level uncapacitated facility location problem $\mathcal{I}$ that satisfies Assumption 4.1. Then we can find, in polynomial time, an equivalent instance $\widetilde{\mathcal{I}}$ and a complete feasible solution $(\widetilde{x},\widetilde{y})$ to the linear programming relaxation of the 1-level uncapacitated facility location problem with the same fractional facility and service costs as $(\overline{x},\overline{y})$.*

Proof: First, we describe an algorithm that transforms the solution $(\overline{x},\overline{y})$ of the 2-level problem to a complete feasible solution $(\widetilde{x},\widetilde{y})$ of the 1-level problem. Then, we show that the fractional facility and service cost does not change when performing the algorithm. Finally, we show that the algorithm can be executed in polynomial time.

The following algorithm can be used to transform a feasible solution $(\overline{x},\overline{y})$ of th LP-relaxation of the 2-level uncapacitated facility location problem to a complete feasible solution $(\widetilde{x},\widetilde{y})$ of the LP-relaxation of the 1-level uncapacitated facility location problem. The steps in the algorithm will be explained below.

1. Initialize a feasible LP-solution to the 2-level problem $(\widehat{x},\widehat{y})=(\overline{x},\overline{y})$.
2. Repeat the following as long as not all connected components are considered:
    (a) Consider a connected component in which $\widehat{x}_{pj}=\frac{1}{b}$ for all $p\in\mathcal{P},\ j\in\mathcal{D}$ for which $\widehat{x}_{pj}>0$ with $b\in\mathbb{N}_{>0}$.
    (b) Repeat the following as long as at least one facility $i\in\mathcal{F}^1$ that is part of the connected component exists for which $\widehat{y}_i>\frac{1}{b}$:
        i. Select a facility $i\in\mathcal{F}^1$ for which $\widehat{y}_i>\frac{1}{b}$.
        ii. Create a new facility $i^c,\ c\in\mathbb{N}_{>0}$ which is an exact copy of facility i. This means that facilities i and i^c have the same fixed and service cost. c will take the smallest possible value, which means that c will equal 1 if no copy of facility i is yet made, while it will equal 2 if one copy is already made, etcetera. Add facility i^c to the sets of facilities $\mathcal{F}^1$ and $\mathcal{F}$.
        iii. Determine which client(s) $j\in\mathcal{D}$ force facility i to open with $\widehat{y}_i>\frac{1}{b}$.
        iv. Repeat the following until all clients j from Step 2(b)iii are considered:
            A. Select a client $j\in\mathcal{D}$ that forces $\widehat{y}_i$ to be larger than $\frac{1}{b}$. If this step is already performed at least once in this iteration and client j is connected to a path

$(i, \bar{l})$ for a facility $\bar{l}$ at level 2 for which it holds that there exists a client $\bar{j}$ in such a way that $\widehat{x}_{(i^c,\bar{l})\bar{j}} > 0$, then this path is selected. Set $\widehat{x}_{(i^c,\bar{l})j} = \frac{1}{b}$ and $\widehat{x}_{(i\bar{l})j} = 0$. Otherwise, select a path $(i, l) \in \mathcal{P}$ for which $\widehat{x}_{(i,l)j} > 0$. Add the path (i^c, l) to the set of paths $\mathcal{P}$ and set $\widehat{x}_{(i^c,l)j} = \frac{1}{b}$ and $\widehat{x}_{(i,l)j} = 0$.

B. Remove client j from the set of clients that force facility i to open with $\widehat{y}_1 > \frac{1}{b}$.

v. Set $\widehat{y}_{i^c} = \frac{1}{b}$ and $\widehat{y}_i = \widehat{y}_i - \frac{1}{b}$.

(c) Repeat the following as long as at least one facility $i \in \mathcal{F}^2$ that is part of the connected component exists for which $\widehat{y}_i > \frac{1}{b}$:

i. Select a facility $i \in \mathcal{F}^2$ for which $\widehat{y}_i > \frac{1}{b}$.

ii. Create a new facility i^c, $c \in \mathbb{N}_{>0}$ which is an exact copy of facility i in the same ways as in step 2(b)ii. Add the new facility i^c to the sets of facilities $\mathcal{F}^2$ and $\mathcal{F}$.

iii. Determine which client(s) $j \in \mathcal{D}$ force facility i to open with $\widehat{y}_i > \frac{1}{b}$.

iv. Repeat the following until all clients j from Step 2(c)iii are considered:

A. Select a client $j \in \mathcal{D}$ that forces $\widehat{y}_i$ to be larger than $\frac{1}{b}$. If this step is already performed at least once in this iteration and client j is connected to a path $(\bar{l}, i)$ for a facility $\bar{l}$ at level 1 for which it holds that there exists a client $\bar{j}$ in such a way that $\widehat{x}_{(\bar{l},i^c)\bar{j}} > 0$, then this path is selected. Set $\widehat{x}_{(\bar{l},i^c)j} = \frac{1}{b}$ and $\widehat{x}_{(\bar{l},i)j} = 0$. Otherwise, select a path $(l, i) \in \mathcal{P}$ for which $\widehat{x}_{(l,i)j} > 0$. Add the path (l, i^c) to the set of paths $\mathcal{P}$ and set $\widehat{x}_{(l,i^c)j} = \frac{1}{b}$ and $\widehat{x}_{(l,i)j} = 0$.

B. Remove client j from the set of clients that force facility i to open with $\widehat{y}_1 > \frac{1}{b}$.

v. Set $\widehat{y}_{i^c} = \frac{1}{b}$ and $\widehat{y}_i = \widehat{y}_i - \frac{1}{b}$.

3. Repeat the following until all paths and clients are considered:

(a) Select a path $p \in \mathcal{P}$ for which $\widehat{x}_{pj} > 0$ for at least one client j.

(b) Create a new "superfacility" s that consists of the two facilities $i_1 \in \mathcal{F}^1$, $i_2 \in \mathcal{F}^2$ that are part of path p ($p = (i_1, i_2)$). The fixed cost of the "superfacility" equals the sum of the fixed cost of facilities i_1 and i_2. For each client j, the service cost from "superfacility" s to client j equals the sum of the service cost from facility i_1 to facility i_2 and the service cost from facility i_2 to client j.

(c) Set $\widetilde{x}_{sj} = \widehat{x}_{pj}$ and $\widetilde{y}_s = \widehat{y}_{i_1} = \widehat{y}_{i_2}$.

First, note that a solution $(\overline{x}, \overline{y})$ that satisfies Assumption 4.1 does not have to be complete. For example, the solution shown in Figure 4.2 is a solution that satisfies Assumption 4.1, but is not complete. The solution consists of two connected components. In the left component only values of $\frac{1}{2}$ and in the right component only 1, so the assumption is satisfied. However, the probability that a facility will be opened does not equal x_{pj} for each facility on path p, because clients c_1 and c_2 receive fraction $\frac{1}{2}$ from facility a_1 using path (a_1, b_1) and fraction $\frac{1}{2}$ from facility a_1 using path (a_1, b_2). Thus, clients c_1 and c_2 receive all demand from facility a_1, which indicates that facility a_1 is opened with 1 in stead of $\frac{1}{2}$, so the solution is not complete. In general, a feasible solution to the LP-relaxation of the 2-level uncapacitated facility location problem that satisfies Assumption 4.1 can violate the completeness in two ways. First, it can happen that for a certain facility $i_1 \in \mathcal{F}^1$ more paths $p = (i_1, i_2) \in \mathcal{P}$ exist for which $\overline{x}_{pj} = \frac{1}{b}$ for a certain client $j \in \mathcal{D}$. Then, client j receives $\frac{2}{b}$ or even more (dependening on how many paths that include facility i_1 are connected to client j) of its demand from facility i_1 and thus is facility i_1 not opened with fraction $\frac{1}{b}$ but with fraction $\frac{2}{b}$ or more. Second, for a certain facility $i_2 \in \mathcal{F}^2$ more paths $p = (i_1, i_2) \in \mathcal{P}$ exist for which $\overline{x}_{pj} = \frac{1}{b}$ for a certain client j. Then, client j receives $\frac{2}{b}$ or even more (dependening on how many paths that include facility i_2 are connected to client j) of its demand from facility i_2 and thus is facility i_2 not opened with fraction $\frac{1}{b}$ but with fraction $\frac{2}{b}$ or more.

Steps 2b and 2c ensure that a complete solution is obtained. The complete solution is called $(\widehat{x}, \widehat{y})$. In Step 2b the case in which one or more facilities at level 1 exist for which $\widehat{y}_i > \frac{1}{b}$ is considered and in Step 2c the case in which facilities at level 2 exist for which $\widehat{y}_i > \frac{1}{b}$. The idea of Steps 2b and 2c is the same, but they are separated to gain some intuition behind the algorithm. First a facility i for which $\widehat{y}_i > \frac{1}{b}$ is selected in Steps 2(b)i and 2(c)i. Then, in Steps 2(b)ii and 2(c)ii, an exact copy i^c of facility i is made, which means that the fixed and service cost of facilities i and i^c are equal. In this way, the fractional facility and service cost will not change when facility i^c is included somewhere in the solution instead of facility i. In Steps 2(b)iv and 2(c)iv a client j is selected that receives more than $\frac{1}{b}$ of its demand from facility i and thus forces $\widehat{y}_i$ to be larger than $\frac{1}{b}$. For each client, exactly one value $\widehat{x}_{pj}$ is changed to ensure that the client receives at most $\frac{1}{b}$ of its demand from the new facility i^c. In Steps 2(b)ivB and 2(c)ivB client j is connected to a new path (i^c, l) or (l, i^c) respectively. When this is done for each client that forces facility i to open with a fraction higher than $\frac{1}{b}$, facility i^c is opened with fraction $\frac{1}{b}$, since each client is connected to a path including facility i^c with fraction $\frac{1}{b}$ or 0 and at least one client is connected to a path that includes i^c, because otherwise facility i^c would not have been created. Therefore, in Steps 2(b)v and 2(c)v facility i^c is opened with fraction $\frac{1}{b}$. Furthermore, for each client that forces facility i to open with a fraction more than $\frac{1}{b}$, exactly $\frac{1}{b}$ of its demand is reallocated to facility i^c in stead of i. Therefore, for all those clients, the fraction of demand that is delivered from facility i is lowered by $\frac{1}{b}$. Thus, the fraction of facility i that is opened can also be lowered with $\frac{1}{b}$. In the solution obtained after Step 2b, the solution is complete with respect to the first level. Step 2c ensures that the solution is complete with respect to the second level. Since Step 2c does not change anything at level 1, the solution obtained after this step is complete with respect to both levels. The conditions in Steps 2(b)ivA and 2(c)ivA make sure that no unnecessarily additional paths are added to the solution. Without these conditions, a connected component can be splitted in more connected components in the transformed 1-level solution. However, it then becomes possible that two "superfacilities" that occur in two different connected components have a facility at one of the two levels in common. Since each connected component will contain at least one cluster, it will become possible that a facility will be opened twice in the algorithm or, even worse, that two facilities at a certain level will be opened, while in practice one facility would satisfy. This can only happen when two or more clients are both connected to two or more same paths. If for one of the clients the first path is reallocated to the copied facility and for the other client the second path is reallocated to the copied facility, the interdependence between the facilities disappears. However, the conditions in Steps 2(b)ivA and 2(c)ivA ensure that this cannot happen.

In Figure 4.3 the complete solution obtained with the algorithm when the initial solution is as in Figure 4.2 is given. In this figure, a copy a_1^1 of facility a_1 is made, because facility is opened with 1 instead of $\frac{1}{2}$ in the initial solution. Then, one of paths (a_1, b_1) and (a_1, b_2) has to be selected, since these paths satisfy $x_{pj} > 0$. In this case, path (a_1, b_2) is selected. Then, thise path is removed from the solution and replaced by path (a_1^1, b_2). As a result, both facilities a_1

and a_1^1 are opened with fraction $\frac{1}{2}$ and we have obtained a complete solution. The paths (a_1, b_1) and $\left(a_1^1, b_2\right)$ are again used twice in the solution.

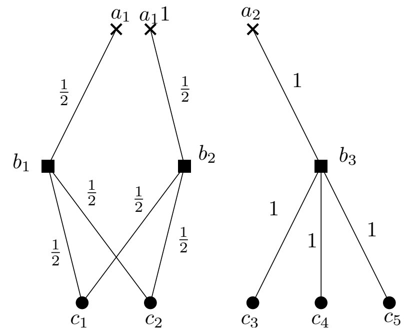

Figure 4.3: Complete solution obtained by copying facility locations.

Thus after Step 2 of the algorithm, a feasible complete solution $(\widehat{x}, \widehat{y})$ is obtained with the same fractional facility and service cost as the original solution $(\overline{x}, \overline{y})$. Then, Step 2a transforms the complete solution $(\widehat{x}, \widehat{y})$ of the 2-level problem to an equivalent solution $(\widetilde{x}, \widetilde{y})$ of the 1-level problem. Since the solution obtained after Step 2 is complete, a path can be seen as a "superfacility". The probability that a client is connected to a "superfacility" is $\widehat{x}_{pj}$, where p is the path that is transformed to "superfacility" and j the client. Next, costs have to be assigned to the "superfacilities" in such a way that the fractional facility and service costs do not change compared to the original costs. A natural idea is to include the cost related to the previous path in the cost of the new "superfacility". Then, one possibility is to add the fixed costs of both facilities that are part of the path together with the service cost between these facilities to obtain the cost of the "superfacility". However, consider the example shown in Figure 4.4. The paths (a_1, b_1) , $\left(a_1^1, b_2\right)$ and (a_2, b_3) are transformed to "superfacilities" s_1, s_2 and s_3 respectively. Now, for example, the service cost of path (a_1, b_1) is included in "superfacility" s_1. From Figure 4.2 we can observe that the service cost of this path is incurred with fraction $\frac{1}{2}$ by client c_1 and with fraction $\frac{1}{2}$ by client c_2. In total, the service cost is thus incurred with factor 1. However, "superfacility" s_1 is only opened with fraction $\frac{1}{2}$ in the transformed solution. Thus, the service cost then have to be included with factor 2 in the "superfacility". However, when more clients are connected to the "superfacility" this factor has to be adapted. Thus, this method will not work. Therefore, we propose another method in which only the fixed costs of the facilities are added to obtain the fixed cost of the "superfacility". The service cost of the path that is transformed to "superfacility" is added to the service cost of the client to the "superfacility". In this way, all service costs that are incurred from the facility at level 1 to the client are included in the service cost from the "superfacility" to the client and are thus multiplied by the correct factor. The fixed cost are also incurred in the right way, because both facilities are open with the same fraction as the "superfacility".

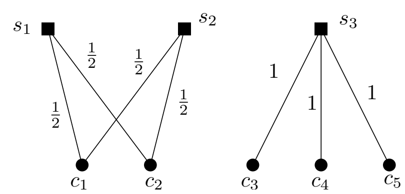

Figure 4.4: Transformed solution to the 1-level problem.

When the conditions in Steps 2(b)ivA and 2(c)ivA are not implied, then it would also be possible to construct the complete solution in Figure 4.5. Transforming this solution will result in the solution shown in Figure 4.6. In this solution, "superfacility" s_1 consists of facilities a_1 and b_1, s_2 of facilities a_1^1 and b_2, s_3 of a_1 and b_2 and s_4 of a_1^1 and b_1. The left connected component in the solution shown in Figures 4.2 and 4.5 is now split into two different connected components. After clustering either s_1 or s_2 will be opened. The same holds for s_3 or s_4. Comparing the possibilities show that always three of the four facilities a_1, a_1^1, b_1 and b_2 are opened. When facilities a_1, b_1 and b_2 are opened too many costs are incurred, because both facilities b_1 and b_2 are opened with only $\frac{1}{2}$ in the optimal solution to the LP-relaxation.

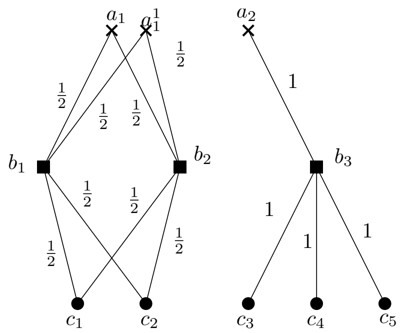

Figure 4.5: Complete solution obtained by copying facility locations without conditions.

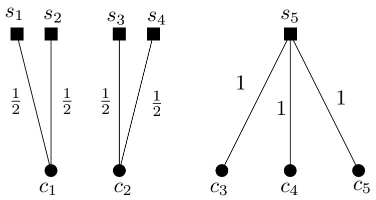

Figure 4.6: Transformed solution obtained by copying facility locations without conditions.

Finally, it has to be shown that this procedure runs in polynomial time. This can be checked by determining the maximum number of copies of facilities that have to be made. Because it is assumed that all paths that are used in a connected component of the optimal solution of the linear program are used with the same fraction, it can be seen that this fraction is at least $\frac{1}{|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2|}$ (as $|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2|$ is an upper bound on the number of possible paths in a 2-level facility location problem). Therefore, for each facility at most $|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2|$ copies have to be made to obtain a complete solution in which all paths and facilities are used with the same fraction. The total number of facilities is $|\mathcal{F}^1| + |\mathcal{F}^2| = |\mathcal{F}|$, so at most $|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2||\mathcal{F}| \leq n^4$ iterations are needed to obtain a complete solution. Therefore, the procedure that constructs a complete solution can be performed in polynomial time. ∎

Using the same approach as in Chudak and Shmoys (2003) it can now be shown that the optimal value of the linear programming relaxation is within a factor of $1 + \frac{2}{e} \approx 1.736$ of the optimal cost.

Theorem 4.4 *When Assumption 4.1 is satisfied, the* $(1+2/e)$*-approximation algorithm of Chudak and Shmoys (2003) can be extended to the 2-level uncapacitated facility location problem.*

Proof: We have already shown that a feasible solution of the transformed 1-level problem can be found that is within $(1 + 2/e)$ of the optimal solution. Now, it still has to be shown that this solution can be extracted to a feasible solution to the 2-level problem, that the costs do not change when extracting this solution and that the algorithm runs in polynomial time for the 2-level problem.

First, note that we have obtained an integer solution of the transformed 1-level problem. In this solution, all facilities at level 1 are "superfacilities", which consists of a facility i_1 at level 1 and a facility i_2 at level 2 of the 2-level problem. Furthermore, the path (i_1, i_2) is included in the "superfacility". Now, we can extract the transformed 1-level solution by extracting the "superfacilities". Thus, each facility at level 1 of the transformed problem is extracted into two facilities (facility i_1 at level 1 and facility i_2 at level 2) and the path (i_1, i_2). If the "superfacility" is opened, both facilities i_1 and i_2 are also opened. Furthermore, if the demand of client j is delivered from the "superfacility", this demand will now be delivered using the path (i_1, i_2). In this way, we obtain an integer (feasible) solution to the original 2-level problem.

Next, we show that the costs of the new solution to the 2-level problem equals that of the solution to the transformed 1-level problem. If a "superfacility" is opened in the transformed 1-level problem, fixed cost will be incurred. These fixed cost equals the sum of the fixed cost of facilities i_1 and i_2 that are included in the "superfacility". Thus, the cost of opening the "superfacility" is $f_{i_1} + f_{i_2}$. After extracting the solution, both facilities i_1 and i_2 are opened. In the extracted solution the fixed cost equals also $f_{i_1} + f_{i_2}$. Furthermore, in the transformed 1-level problem, the service cost incurred when transporting the demand of a client j to a superfacility s is c_{sj}. This service cost consists of the sum of the service cost from facility i_1 to facility i_2 (where facilities i_1 and i_2 are again part of "superfacility" s) and the service cost from facility i_2 to client j. Thus, $c_{sj} = c_{i_1 i_2} + c_{i_2 j}$. In the extracted 2-level problem, service costs are incurred when transporting the demand from facility i_1 to facility i_2 and from facility i_2 to client j. Thus, in the extracted 2-level problem the service cost equals also $c_{i_1 i_2} + c_{i_2 j}$. Since both the facility and service cost do not change when extracting the transformed 1-level problem to the original 2-level problem, the total cost also remain the same.

Finally, we have to show that the algorithm can be executed in polynomial time. Thereto, first note that the ellipsoid algorithm can be used to solve the dual linear program in polynomial time since the dual has a polynomial number of variables (Aardal et al. (1999)). Furthermore, it can be assumed that the algorithm finds a basic optimal solution of the primal linear problem in polynomial time (Aardal et al. (1999)). We have already seen that the time needed to construct a complete solution of the linear programming problem is polynomial. Thereafter, we have obtained a solution to an equivalent instance of the 1-level uncapacitated facility location problem and thus the same methods as in Chudak and Shmoys (2003) can be used. Because the algorithm of Chudak and Shmoys (2003) is a polynomial time algorithm, these methods also run in polynomial time. ∎

4.2 Extension to higher level facility location problems

In this section, we investigate whether the extension to the 2-level facility location problem developed in the previous section can be further extended to higher level facility location problems. Therefore, we first introduce the linear programming relaxation and its dual problem for the k-level uncapacitated facility location problem as given in Aardal et al. (1999). In the k-level problem, a path $p \in \mathcal{P}$ is defined as a sequence of k facilities $p = (i_1, i_2, \ldots, i_k)$ with $i_l \in \mathcal{F}^l$.

$$z_{LP} = \min \sum_{l=1}^{k} \sum_{i_l \in \mathcal{F}^l} f_{i_l} y_{i_l} + \sum_{p \in \mathcal{P}} \sum_{j \in \mathcal{D}} c_{pj} x_{pj} \tag{4.10}$$

subject to

$$\sum_{p\in\mathcal{P}} x_{pj} = 1, \qquad \forall j \in \mathcal{D} \tag{4.11}$$

$$\sum_{p:p\ni i_l} x_{pj} - y_{i_l} \leq 0, \qquad \forall j \in \mathcal{D}, \quad \forall i_l \in \mathcal{F}^l, \quad l = 1, 2, \ldots, k \tag{4.12}$$

$$x_{pj} \geq 0, \qquad \forall p \in \mathcal{P}, \quad \forall j \in \mathcal{D} \tag{4.13}$$

$$\tag{4.14}$$

Again, we define the fractional facility and service cost given a feasible fractional solution $(\overline{x}, \overline{y})$ as respectively $\overline{C}_f := \sum_{l=1}^{2} \sum_{i_l \in \mathcal{F}^l} f_{i_l} \overline{y}_{i_l}$ and $\overline{C}_j := \sum_{p\in\mathcal{P}} \sum_{j\in\mathcal{D}} c_{pj} \overline{x}_{pj}$.

Let v_j and $w_{i_l,j}$ be the dual variables corresponding to the primal constraints (4.11) and (4.12). The dual problem corresponding to the linear programming relaxation is given by

$$z_{LP} = \max \sum_{j\in\mathcal{D}} v_j \tag{4.15}$$

subject to

$$v_j - \sum_{i_l \in p} w_{i_l,j} \leq c_{pj} \qquad \forall p \in \mathcal{P}, \quad \forall j \in \mathcal{D} \tag{4.16}$$

$$\sum_{j\in\mathcal{D}} w_{i_l,j} \leq f_{i_l} \qquad \forall i_l \in \mathcal{F}^l, \quad l = 1, 2, \ldots, k \tag{4.17}$$

$$w_{i_l,j} \geq 0 \qquad \forall j \in \mathcal{D}, \quad \forall i_l \in \mathcal{F}^l, \quad l = 1, 2, \ldots, k \tag{4.18}$$

We assume that the optimal solution of the LP-relaxation of the k-level uncapacitated facility location problem satisfies Assumption 4.1. Then, Lemma 4.2 is still valid for the k-level problem, because its proof does not depend on the number of levels considered in the problem. Next, we prove that Lemma 4.3 can be extended to the k-level problem.

Lemma 4.5 *Suppose that $(\overline{x}, \overline{y})$ is a feasible solution to the linear programming relaxation for a given instance of the k-level uncapacitated facility location problem $\mathcal{I}$ that satisfies Assumption 4.1. Then we can find, in polynomial time, an equivalent instance $\widetilde{\mathcal{I}}$ and a complete feasible solution $(\widetilde{x}, \widetilde{y})$ to the linear programming relaxation of the 1-level uncapacitated facility location problem with the same fractional facility and service costs as $(\overline{x}, \overline{y})$.*

Proof: We can use the algorithm proposed in the proof of Lemma 4.3 to proof this lemma. In the algorithm in the previous section, only facilities at level 1 and level 2 are considered. However, the idea in the corresponding steps of the algorithm (Steps 2b and 2c) is exactly the same. For the other levels, a similar step can be included in the algorithm without changing the results. Furthermore, a path can again be seen as a "superfacility" when the solution is complete. Thus, using the algorithm described in the proof of Lemma 4.3 (extended with additional steps for the other levels) a feasible complete solution to the 1-level uncapacitated facility location problem with the same fractional facility and service costs can be obtained.

Finally, it has to be shown that this procedure runs in polynomial time. This can again be checked by determining the maximum number of copies of facilities that have to be made. Because it is assumed that all paths that are used in a connected component of the optimal solution of the linear program are used with the same fraction, it can be seen that this fraction is at least $\frac{1}{|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2|\cdot\ldots\cdot|\mathcal{F}^k|}$ (as $|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2|\cdot\ldots\cdot|\mathcal{F}^k|$ is an upper bound on the number of possible paths in a k-level facility location problem). Therefore, for each facility at most $|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2|\cdot\ldots\cdot|\mathcal{F}^k|$ copies have to be made to obtain a complete solution in which all paths and facilities are used with the same fraction. The total number of facilities is $|\mathcal{F}^1|+|\mathcal{F}^2|+\ldots+|\mathcal{F}^k| = |\mathcal{F}|$, so at most $|\mathcal{D}||\mathcal{F}^1||\mathcal{F}^2|\cdot\ldots\cdot|\mathcal{F}^k||\mathcal{F}| \leq n^{k+2}$ iterations are needed to obtain a complete solution. Therefore, the procedure that constructs a complete solution can be performed in polynomial time. ∎

Using the same approach as in Chudak and Shmoys (2003) it can now be shown that the optimal value of the linear programming relaxation is within a factor of $1+\frac{2}{e}\approx 1.736$ of the optimal cost.

Theorem 4.6 *When Assumption 4.1 is satisfied, the* $(1+2/e)$*-approximation algorithm of Chudak and Shmoys (2003) can be extended to the* k*-level uncapacitated facility location problem.*

The proof that the algorithm runs in polynomial time for the k-level problem is similar to that of the 2-level problem.

Chapter 5

Computational results

In this chapter, we present some results on the number of times that the solution of the LP-relaxation of the k-level uncapacitated facility location problem satisfies a certain structure. In the previous chapter we showed that the $(1 + 2/e)$-approximation algorithm for the 1-level uncapacitated facility location problem of Chudak and Shmoys (2003) can be extended to the k-level problem when all values of x^*_{pj}, which are larger than 0 in a connected component, are equal. A natural question is whether this assumption can be extended in such a way that all values of x^*_{pj} that are larger than 0 have the same denominator. In Chapter 7 this question will be further discussed. In this chapter, we will already investigate the occurance of such solutions.

We distinguish between integer solutions (integer), fractional solutions that satisfy Assumption 4.1 (assumption), fractional solutions in which all values of x^*_{pj} for which $x^*_{pj} > 0$ have equal denominator z with $z < n$, but that do not satisfy Assumption 4.1 (equal denominator) and all other solutions (other). In solutions that satisfy Assumption 4.1 only one value larger than 0 can be taken in a connected component, so for example only $\frac{1}{2}$ or $\frac{1}{3}$. Examples of equal denominator solutions are solutions in which only the values $\frac{1}{2}$ and 1 appear in a connected component, or only $\frac{1}{3}$, $\frac{2}{3}$ and 1 (1 is seen as $\frac{2}{2}$ and $\frac{3}{3}$ respectively in these cases). Furthermore, connected components in which both values of $\frac{1}{4}$ and $\frac{1}{2}$ occur, are also included in this structure because $\frac{1}{2} = \frac{2}{4}$. Notice, that in this way every two fractions $\frac{1}{p}$ and $\frac{1}{q}$ can be written as fractions with equal denominator, because $\frac{1}{p} = \frac{q}{pq}$ and $\frac{1}{q} = \frac{p}{pq}$. However, when the value of pq becomes larger than n, the solution does not satisfy the equal denominator structure anymore and will be categorized in the structure with other solutions.

5.1 Results for 1-level uncapacitated facility location problems

In this section, we determine how many times the solution of the LP-relaxation satisfies a certain structure. A comparison is made between small and large examples. Each small example consists of 5 facilities and 10 clients, where each large example consists of 25 facilities and 100 clients. In Table 5.1, for each structure the percentage of solutions of the LP-relaxation that satisfies this structure is given. Furthermore, the average duality gap is given for the small and large instances. The duality gap is calculated as $\frac{z_{IP}-z_{LP}}{z_{IP}} \times 100\%$. The duality gap is only calculated for noninteger solutions (since the duality gap will be 0 for integer solutions) and averaged over the solutions for which the duality gap is calculated. Since the average duality gap for the different structures are almost the same, only the average over all fractional solutions is given. As can be seen from the table, for small instances almost all problem instances result in an integer solution of the linear programming relaxation. Notice that an integer solution of the LP-relaxation means that the solution is optimal according to the integer uncapacitated facility location problem. However, when the problem instances increase, the number of integer solutions decrease. For small problems, almost 35% of the fractional solutions satisfy Assumption 4.1. When the size of the problems increase, this percentage drops to almost 29%. Furthermore, the table shows that all problem instances give a solution that satisfy the equal denominator structure (since solutions that are integer or satisfy Assumption 4.1 also satisfy the equal denominator structure). Remarkably, the average duality gap is larger for small problem instances than for large instances. An explanation for this observation can be that in large instances it is more probable that the solution of the LP-relaxation is almost integer (with only a few fractional values). Then, the integer optimal solution differs only a little from the optimal fractional solution, because only the variables with fractional values have to be changed. Thus, the duality gap is small for these problem instances.

Table 5.1: Percentage of solutions that satisfy a structure for the 1-level problem

Structure	Small	Large
Integer	96.8	54.8
Assumption	1.1	13.1
Equal denominator	2.1	32.1
Other	0.0	0.0
Average duality gap	8.3	0.5

5.2 Results for 2-level uncapacitated facility location problems

First, 10,000 small examples with 3 facilities at level 1, 5 facilities at level 2 and 10 clients are generated and solved. Thereafter, 10,000 large examples with 15 facilities at level 1, 25 at level 2 and 200 clients are generated and solved. In Table 5.2 the percentage of solutions that satisfy a certain structure and the average duality gaps are given.

Table 5.2: Percentage of solutions that satisfy a structure for the 2-level problem

Structure	Small	Large
Integer	98.2	63.0
Assumption	1.3	2.3
Equal denominator	0.5	34.7
Other	0.0	0.0
Average duality gap	13.9	0.8

Table 5.2 shows that for small instances almost all solutions to the linear programming relaxation are integer (98.2%). However, when the size of the instances increase, the probability that the LP-relaxation gives an integer solution decreases to almost 63%. This is a similar observation as for the 1-level problem. Now, more than 72% of the fractional solutions of small instances satisfy the assumption made in this thesis. After increasing the size of the problem instances this percentage drops to only 6%. However, again all solutions satisfy the equal denominator structure. The duality gaps for both small and large instances are increased compared to the 1-level problem. The gap for the small instances is still higher than that of large instances

5.3 Results for 3-level uncapacitated facility location problems

For the 3-level uncapacitated facility location problem, again 10,000 small and large instances are randomly generated and the LP-relaxation is solved. This time 2 facilities at the first level, 3 at the second, 5 at the third level and 10 clients are generated in the small problem instances. For the large instances, 10 facilities at level 1, 15 at level 2, 25 at level 3 and 200 clients are generated. The results are given in Table 5.3. Again, the small instances give almost always integer and thus optimal results. When the problem instances are increased, the number of optimal solutions again decrease. The other observations are also similar to the 1 and 2-level problems: the percentage of fractional solutions that satisfy Assumption 4.1 decreases when the problem size increases, all solutions satisfy the equal denominator structure and the average duality gap is larger for small problem instances than for large instances. Again, the duality gaps are increased compared to the 2-level problem.

Table 5.3: Percentage of solutions that satisfy a structure for the 3-level problem

Structure	Small	Large
Integer	98.4	60.2
Assumption	1.3	3.0
Equal denominator	3.0	36.8
Other	0.0	0.0
Average duality gap	15.2	1.4

5.4 Evaluation

For small problem instances, almost all solutions of the linear programming relaxation are integer. However, when a noninteger solution is obtained, the duality gap between the optimal integer solution and the optimal solution of the LP-relaxation is relatively high and increases with the number of included levels. For large problem instances the number of times a noninteger solution is obtained increases, but the duality gap decreases compared to the small problems. The average duality gap still increases when the number of included levels increases, but for low levels, the LP-relaxation gives a good estimate of the optimal integer solution.

The percentage of fractional solutions that satisfy the assumption made in this thesis is more than 30% for small problems with 1, 2 and 3 levels. However, when the problem size increases, this percentage drops to less than 10% for the 2 and 3-level problems. Since all solutions satisfy the equal denominator structure, it would be very useful to investigate whether the $(1 + 2/e)$-approximation algorithm has also a valid extension for solutions that satisfy this structure.

Chapter 6

Conclusion

In Chudak and Shmoys (2003) a $(1+2/e)$-approximation algorithm for the 1-level uncapacitated facility location problem is provided. In this thesis, we try to extend this algorithm to an approximation algorithm for higher level uncapacitated facility location problems. Thereto, an assumption on the structure of the linear programming relaxation of the facility location problem is made. We assumed that all variables x_{pj} that are larger than 0 have the same value in a connected component. We proved that under this assumption, the solution of the LP-relaxation of the k-level uncapacitated facility location problem can be transformed into a solution of a 1-level uncapacitated facility location problem. Furthermore, we proved that this transformation can be done in polynomial time. Therefore, the algorithm of Chudak and Shmoys (2003) can be extended to a $(1 + 2/e)$-approximation algorithm for the k-level uncapacitated facility location problem under this assumption.

Furthermore, we investigated how many times solutions of the LP-relaxation satisfied certain structure requirements for 1,2 and 3-level facility location problems. We distinguished between integer solutions, fractional solutions that satisfy Assumption 4.1, fractional solutions for which all values of $x^*_{pj} > 0$ have equal denominator z with $z < n$, but do not satisfy Assumption 4.1, and all other solutions. For small problems almost all instances resulted in an integer solution of the linear programming relaxation. For large problems, the number of integer solutions decreased. Furthermore, the number of fractional solutions that satisfy Assumption 4.1 also decrease when the size of the instances increases. However, all solutions of the LP-relaxation satisfied the structure with equal denominators. Thus, the approximation algorithm developed in this thesis is mostly useful for uncapacitated facility location problems with only a few levels.

The average duality gap, the relative difference of the LP-relaxation compared to the optimal integer solution, is very small for large problem instances. Thus, for large instances the solution to the linear programming relaxation is a good estimate of the optimal integer solution. However, the average duality gap increases when the number of considered levels in the facility location problem increases. For small problems the average duality gap is larger. Furthermore, it also increases when the number of included levels increases.

Chapter 7

Discussion

In this thesis, we extended the $(1 + 2/e)$-algorithm for the 1-level uncapacitated facility location problem as described in Chudak and Shmoys (2003) to the k-level problem under an assumption on the structure of the optimal solution to the LP-relaxation. More precisely, in Chapter 4 we developed an approximation algorithm for the k-level problem with the same performance garantuee as for the 1-level under the assumption that all variables x^*_{pj} for which $x^*_{pj} > 0$ within a connected component of the optimal solution of the linear programming relaxation, have equal value. In Chapter 5, we presented some computational results on the fraction of LP-solutions that satisfy this assumption for different level uncapacitated facility location problem. Because all generated problems for the 1,2 and 3-level uncapacitated facility location problems satisfy the equal denominator structure, it would be very useful, especially for higher level facility location problems, to develop an approximation algorithm that is also valid for this kind of solutions. This will be discussed in this chapter.

The assumption made in this thesis is used in the proof of Lemmas 4.3 and 4.5. The assumption provides for every facility location an upper bound on the number of copies that have to be made to obtain a complete solution. This upper bound is needed to prove that the algorithm can be performed in polynomial time. This suggests that the approximation algorithm is still valid for solutions of the linear programming relaxation that does not satisfy Assumption 4.1 if for every facility location an upper bound on the number of copies needed to obtain a complete solution can be determined and the sum of all upper bounds is polynomial in n.

First, we consider the solutions for which all variables in a connected component have the same denominator, but that do no satisfy Assumption 4.1. We can adjust this type of solutions in such a way that it satisfies Assumption 4.1. Thereto, we have to make copies of the paths for which the variables are larger than the minimum value of a variable in the same connected component. A copy of a path consists of a copy of each facility that is part of the path.

Consider an instance for which the solution to the LP-relaxation contains only variables with the same denominator, which is less than or equal to n, in a connected component. First, notice that connected component for which all values of $x^*_{pj} > 0$ are the same, already satisfy Assumption 4.1 and do not have to be changed. Next, consider a connected component for which different values of x^*_{pj} exist. Since the solution of the LP-relaxation satisfies the equal denominator structure, all these values of x^*_{pj} are of the form $\frac{t}{z}$ with $t = 0, 1, \ldots, z$ for a certain $z \leq n$. We want to obtain an equivalent solution in which all values of $x^*_{pj} > 0$ are equal. The most logical value that all these x^*_{pj} will take in the equivalent solution is $\frac{1}{z}$. Thus all combinations of paths and clients for which $x^*_{pj} = \frac{1}{z}$ do not have to be changed. Consider a combination of path and client (p, j) for which $x^*_{pj} = \frac{t}{z}$ with $t = 2, 3, \ldots, z$. Then, we make $t-1$ copies of the path p $(p^1, p^2, \ldots, p^{t-1})$ and reallocate $\frac{1}{z}$ to all the combinations of a copied path $p^u, u = 1, \ldots, t-1$ and client j. A copy of a path contains a copy of all facilities that are part of the path. The facility and service costs of a copied facility are the same as that of the original facility. In this way, we have created t paths with $x^*_{pj} = \frac{1}{z}$ in stead of the original path with $x^*_{pj} = \frac{t}{z}$ without changing the costs of the optimal LP-solution. This can be repeated for each combination of path p and client j for which $x^*_{pj} > \frac{1}{z}$. This method seems effective at a first sight, but some problems can occur. For example, a connected component of the k-level problem can be split into more connected components in the transformed 1-level problem. Then, the performance guarantee of $(1 + 2/e)$ cannot be given anymore, because clusters can also become smaller in this case. Since only one facility per cluster is opened, smaller clusters (and thus more clusters) result in higher facility costs. Due to time constraints, we did not succeed in answering these questions.

However, we can already prove that the algorithm described above is a polynomial time algorithm. First, note that the number of possible combinations between a path and a client is $|\mathcal{F}_1||\mathcal{F}_2||\mathcal{D}| \leq n \cdot n \cdot n = n^3$. Furthermore, for each combination of a path and a client at most $z \leq n$ copies of the path have to be made. Each copy of a path contains of a copy of all facilities of the path, so in the 2-level problem, 2 copies have to be made. The total number of copies that have to be made can thus be bounded by $2 \cdot n \cdot n^3 = 2n^4 (\leq n^5)$. This expression is polynomial in n, so the above algorithm is a polynomial time algorithm. Since the algorithm is polynomial and (almost) all solutions of the 1, 2 and 3-level problems satisfy this structure, it would be very useful to investigate whether this proof can be finished.

Bibliography

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

J. Byrka and K. Aardal. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM Journal on Computing*, 39(6):2212–2231, 2010.

M. Charikar and S. Guha. Improved combinatorial algorithms for facility location and k-median problems. *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 378–388, 1999.

M. Charikar and S. Guha. Improved combinatorial algorithms for facility location problems. *SIAM Journal on Computing*, 34(4):803–824, 2005.

F.A. Chudak. Improved approximation algorithms for uncapacitated facility location. *Proceedings of the 6th Integer Programming and Combinatorial Optimization (IPCO)*, 1412:180–194, 1998.

F.A. Chudak and D.B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM Journal on Computing*, 33(1):1–25, 2003.

V. Chvátal. *Linear programming.* W.H. Freeman and Company, New York, seventeenth edition, 1983. ISBN 0-7167-1587-2.

S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 228–248, 1998.

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

D.S. Hochbaum (editor). *Approximation algorithms.* PWS Publishing Company, 1997.

S. Li. A 1.488-approximation algorithm for the uncapacitated facility location problem, 2011. ICALP.

M. Mahdian, Y. Ye, and J. Zhang. Approximation algorithms for metric facility location problems. *SIAM Journal on Computing*, 36(2):411–432, 2006.

P.B. Mirchandani and R.L. Francis (editors). *Discrete location theory*, pages 119–172. John Wiley and Sons, Inc., New York, second edition, 1990. ISBN 0-471-89233-5.

D.B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 265–274, 1997.

M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. *Proceedings of the 9th Integer Programming and Combinatorial Optimization (IPCO)*, 2337:240–257, 2002.

L.A. Wolsey. *Integer programming.* John Wiley and Sons, Inc., New York, first edition, 1998. ISBN 978-0-471-28366-9.

J. Zhang. Approximating the two-level facility location problem via a quasi-greedy approach. *Mathematical Programming*, 108:159–176, 2006.