Universidade Estadual de Campinas
Instituto de Computação

Lucas Prado Melo

k-Level Stochastic Facility Location Problems

Problemas de Localização de Instalação Estocásticos k-Níveis

CAMPINAS
2016

Lucas Prado Melo

k-Level Stochastic Facility Location Problems

Problemas de Localização de Instalação Estocásticos k-Níveis

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientador: Prof. Lehilton Lelis Chaves Pedrosa
Co-supervisor/Coorientador: Prof. Dr. Flávio Miyazawa e Prof. Dr. Rafael Schouery

Este exemplar corresponde à versão final da Dissertação defendida por Lucas Prado Melo e orientada pelo Prof. Lehilton Lelis Chaves Pedrosa.

CAMPINAS
2016

Agência(s) de fomento e nº(s) de processo(s): CAPES, 1364110; CAPES, 1406909; CNPq, 154329/2015-0

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

M491k

Melo, Lucas Prado, 1989-
k-level stochastic facility location problems / Lucas Prado Melo. – Campinas, SP : [s.n.], 2016.

Orientador: Lehilton Lelis Chaves Pedrosa.
Coorientadores: Flávio Keidi Miyazawa e Rafael Crivellari Saliba Schouery.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Algoritmos de aproximação. 2. Localização de instalações (Pesquisa operacional). 3. Modelos estocásticos. 4. Otimização combinatória. I. Pedrosa, Lehilton Lelis Chaves,1985-. II. Miyazawa, Flávio Keidi,1970-. III. Schouery, Rafael Crivellari Saliba. IV. Universidade Estadual de Campinas. Instituto de Computação. V. Título.

Informações para Biblioteca Digital

Título em outro idioma: Problemas de localização de instalação estocásticos k-níveis
Palavras-chave em inglês:
Approximation algorithms
Facility location(Operational research)
Stochastic models
Combinatorial optimization
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Lehilton Lelis Chaves Pedrosa [Orientador]
Uéverton dos Santos Souza
Eduardo Candido Xavier
Data de defesa: 12-08-2016
Programa de Pós-Graduação: Ciência da Computação

Universidade Estadual de Campinas
Instituto de Computação

Lucas Prado Melo

k-Level Stochastic Facility Location Problems

Problemas de Localização de Instalação Estocásticos k-Níveis

Banca Examinadora:

- Prof. Dr. Lehilton Lelis Chaves Pedrosa
  Universidade Estadual de Campinas
- Prof. Dr. Uéverton dos Santos Souza
  Universidade Federal Fluminense
- Prof. Dr. Eduardo Candido Xavier
  Universidade Estadual de Campinas

A ata da defesa com as respectivas assinaturas dos membros da banca encontra-se no processo de vida acadêmica do aluno.

Campinas, 12 de agosto de 2016

Resumo

O problema de Localização de Instalações (FLP, do inglês *Facility Location Problem*) é um problema de Otimização Combinatória no qual instalações precisam ser construídas para satisfazer a demanda de um conjunto de clientes. Esse problema possui aplicação em muitas áreas, como serviços de saúde, transporte, logística e produção. Nesta dissertação, discutiremos algoritmos de aproximação para algumas generalizações do FLP: o k-LFLP (FLP em k níveis), no qual instalações são organizadas hierarquicamente e clientes são atendidos por cadeias de instalações de todos os níveis hierárquicos; o FLP Dinâmico e Estocástico (respectivamente DFLP e SFLP), onde custos de abertura e a demanda de clientes muda através do tempo; e variações do FLP Estocástico em k-Níveis (k-LSFLP) que combinam elementos de ambos os problemas anteriores.

Inicialmente, revisamos resultados para o k-LFLP, DFLP e SFLP que utilizamos posteriormente para obter algoritmos de aproximação para o k-LSFLP e sua variação de "Restrição de Estágio" (k-LSSCFLP). Para o k-LSFLP, obtivemos uma $(2k - 1 + o(1))$-aproximação cujo fator de aproximação é 3.495, 5.157 e 7.034 para, respectivamente, $k = 2, 3$ e 4. E, para o k-LSSCFLP, obtivemos uma $(4 - o(1))$-aproximação que possui fator de 2.56, 2.68 e 2.95 para $k = 2, 3$ e 4 respectivamente.

Abstract

The Facility Location Problem (FLP) is a Combinatorial Optimization problem in which some facilities must be built to satisfy demands of a set of clients. It has applications in many areas such as health services, transportation, logistics and production. In this thesis, we will discuss approximation algorithms for some generalizations of the FLP: the k-Level FLP (k-LFLP), where facilities are organized hierarchically and clients are satisfied by chains of facilities spanning all levels; the Dynamic and Stochastic FLP (DFLP and SFLP, respectively), where opening costs and client demands change over time; and variations of the k-Level Stochastic FLP (k-LSFLP) that combines elements of both previous problems.

We first review results for the k-LFLP, DFLP and SFLP that we later utilize to provide approximation algorithms for the k-LSFLP and the "Stage Constrained" k-LSFLP (k-LSSCFLP). For the k-LSFLP, we obtain a $(2k-1+o(1))$-approximation whose approximation factor is 3.495, 5.157 and 7.034 for, respectively, $k = 2, 3$ and 4. And, for the k-LSSCFLP, we obtain a $(4-o(1))$-approximation that has approximation factor of 2.56, 2.68 and 2.95 for $k = 2, 3$ and 4 respectively.

Contents

Chapter 1

Introduction

In the Facility Location Problem (FLP), we look to open facilities to serve clients in a way that minimizes total cost, including costs for opening the facilities and costs associated with the distance from clients to their closest open facilities. This problem along with its variants compose an important class of network design problems with applicantion in many areas, such as health services, transportation, logistics and production (see [16, 12, 35]).

Each variation of the FLP consider aspects of certain real-world applications. For example, facilities might have a maximum capacity of clients it can serve, or an adversary might destroy a number of facilities after they were built. In this thesis, we will be concerned specifically with the k-level FLP (Chapter 2), the Dynamic and Stochastic FLP (Chapter 3) and the k-Level Stochastic FLP (Chapter 4).

The k-level FLP assumes that facilities form a hierarchy and every client must be served by a chain of facilities of all hierarchical levels. For instance, in a 3-level FLP, we might interpret third-level facilities as factories, second-level facilities as warehouses and first-level facilities as stores, thus modelling a simplistic version of a supply chain.

The Dynamic FLP considers change of facility costs and client demands over time. There is a set of discrete time slots, each with its own client demands and facility opening costs. During a time slot, clients might have their demand satisfied by facilities opened at that time slot or earlier, but choosing facilities opened earlier might incur in additional costs.

Stochastic variants of the FLP model uncertainty over final facility opening costs and client demands. Decisions are taken in two time periods known as stages. In the first one, some facilities might be open for a first-stage cost and, in the second stage, one scenario is realized determining opening costs and the set of clients that need to be satisfied.

For more details about those problems, including a brief summary of previous works, please refer to their specific chapters.

FLP is an NP-hard problem, even without the additional difficulties introduced by its generalizations, therefore we cannot hope to find a polynomial-time algorithm that solves it optimally if $P \neq NP$. We can, however, relax some of the constraints imposed by the problem to come up with more practical solutions. In this thesis, we will describe so-called *approximation algorithms* for some FLP variations, that is, instead of describing algorithms that obtain optimal solutions, we will describe polynomial-time algorithms that are guaranteed to obtain solutions that are close to optimal in a certain sense.

We combined techniques from approximation algorithms for the k-level FLP and the Dynamic FLP to obtain approximations to two variations of the k-Level Stochastic FLP (k-LSFLP). We obtained a $(4 - o(1))$-approximation algorithm for the metric stage-constrained k-LSFLP (k-LSSCFLP). For $k = 2, 3, 4$, the algorithm has factors 2.56, 2.78 and 2.95, respectively. These bounds improve on the best known factor of 4 by Wang et al. [32] for the k-LSSCFLP, and of $3 + \epsilon$ by Wu et al. [36] for the 2-LSSCFLP. Also, we obtained a 3.495-approximation algorithm for the 2-LSFLP, that can be generalized to a $(2k - 1 + o(1))$-approximation algorithm for the k-LSFLP. This is the first approximation algorithm for the k-LSFLP. For values of $k = 3, 4$, it has factors 5.157 and 7.0334, respectively.

We will provide some preliminary definitions in Section 1.1 and describe a simple variation of the FLP and one approximation algorithm for it in Section 1.2. Chapters 2 and 3 review approximation algorithms for different variations of the FLP, introducing techniques that are used by our algorithms to the k-LSFLP and k-LSSCFLP. Chapter 4 presents our results. Finally, we close the thesis in Chapter 5 with some concluding remarks.

1.1 Preliminary definitions

An *optimization problem* P is a set of instances. Every instance $\mathcal{I} \in P$ is a tuple (F, c, goal) where F is the set of feasible solutions, $c : F \to \mathbb{R}^+$ is the *objective function* and $\text{goal} \in \{\text{minimize}, \text{maximize}\}$ establishes the type of solution sought in this instance. In other words, if $\text{goal} = \text{minimize}$, then $\mathcal{I}$ asks for a feasible solution with minimum objective value, that is, it looks for $f \in F$ such that, for every $x \in F$, $c(f) \leq c(x)$. Otherwise, in case $\text{goal} = \text{maximize}$, it looks for $f \in F$ such that, for every $x \in F$, $c(f) \geq c(x)$.

Combinatorial optimization problems are optimization problems in which instances' feasible solutions are finite or countably infinite. For example, in an instance of the *Set Cover* problem, we have a set U, a finite family $\{S_i\}_{i \in \mathcal{I}}$ of subsets of U, and a cost function $w : \mathcal{I} \to \mathbb{R}^+$ and we wish to choose a subset $\mathcal{I}' \subseteq \mathcal{I}$ such that $\bigcup_{i \in \mathcal{I}'} S_i = U$ and $\sum_{i \in \mathcal{I}'} w_i$ is minimum. In this case, feasible solutions are all subsets $\mathcal{I}' \subseteq \mathcal{I}$ such that $\bigcup_{i \in \mathcal{I}'} S_i = U$ and the objective function is $c(\mathcal{I}') = \sum_{i \in \mathcal{I}'} w_i$. Since the size of the set of feasible solutions is finite in every instance (it is at most $2^{|\mathcal{I}|}$), the Set Cover problem is a combinatorial optimization problem.

If we assume that $P \neq NP$, some combinatorial optimization problems (such as FLP and Set Cover) cannot be solved by polynomial-time algorithms. However, some relaxed versions of those problems have more practical solutions. We will be concerned with polynomial-time algorithms that abandon the optimality requirement but provide certain approximation guarantees. Namely, we will talk about *α-approximations*.

An $\alpha(n)$-approximation, for a function $\alpha : \mathbb{N} \to [1, \infty)$, is a polynomial-time algorithm for a combinatorial optimization problem that, for each instance (F, c) of size* n, obtains a feasible solution $f \in F$ such that, if the optimal solution has cost OPT, then

$c(f) \leq \alpha(n)\mathrm{OPT}$. We also define as $\alpha(n)$-approximations polynomial-time randomized algorithms that obtain solutions whose cost either in expectation, with high probability, etc. is at most $\alpha(n)$ times the optimal. Alternatively, for maximization problems, $\alpha : \mathbb{N} \to (0, 1]$ and the returned feasible solution f meets $c(f) \geq \alpha(n)\mathrm{OPT}$ instead of $c(f) \leq \alpha(n)\mathrm{OPT}$.

For a more comprehensive and detailed introduction to optimization and combinatorial optimization, see, for example, the introductory chapters in [24] or [4]. To learn more about approximation algorithms, see, for example, [34] or [29].

Linear programs (commonly abbreviated *LP*) are optimization problems that look for a point $\mathbf{x} \in \mathbb{R}^n$ meeting certain linear inequalities and/or linear equations and minimizing (or, alternatively, maximizing) a linear expression given by a function $f : \mathbb{R}^n \to \mathbb{R}$ which is its *objective function*. Recall that a *linear function* on a point $\mathbf{x} \in \mathbb{R}^k$ is any function $g : \mathbb{R}^k \to \mathbb{R}$ in the form $g(x_1, x_2, \dots, x_k) = c_0 + c_1x_1 + c_2x_2 + \cdots + c_nx_n$, where $c_0, c_1, \dots, c_k \in \mathbb{R}$, and *linear inequalities/equations* are relations respectively in forms $g(\mathbf{x}) \leq h(\mathbf{x})$ and $g(\mathbf{x}) = h(\mathbf{x})$, where g and h are linear functions. Optimal solutions to linear programs can be found in polynomial time as proven by Khachiyan [18].

Linear programming is a large and well studied field. Here, we only succintly discuss results that are useful to us. To learn more, the interested reader may refer to textbooks on the subject, for example, [28] and [11]

In *standard form*, linear programs can be succintly expressed as:

$$\begin{aligned} \text{maximize} \quad & \mathbf{c}'\mathbf{x} \\ \text{subject to } & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned} \tag{1.1}$$

where $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ is a real matrix with the coefficients of the inequalities, $\mathbf{c} \in \mathbb{R}^n$ is the real vector with the coefficients of the objective function and $\mathbf{b} \in \mathbb{R}^m$ is a vector. Here, the unary operator "$'$" denote transposition, e.g. c' is the c transposed. Notice that any linear program can be transformed into the standard form by applying elementary algebraic manipulations, adding new variables and/or linear constraints, and/or exchanging the signal of the coefficients of the objective function (to transform a minimization problem to a maximization one).

A point $\mathbf{x} \in \mathbb{R}^n$ such that $A\mathbf{x} \leq \mathbf{b}$ is called a *feasible* point or solution to the LP. The set of feasible points is the *feasible region or set* of the LP. In some cases, the feasible region for a linear program is empty because there is no point that satisfies all constraints simultaneously, e.g. there cannot be any point in which inequalities $x_1+x_2 \leq 3$, $x_2-x_1 \geq 1$ and $x_1 \geq 4$ are met. If a linear program has an empty feasible region, it is called *infeasible*, otherwise it is called *feasible*.

Feasible points $\mathbf{x}$ with maximum or minimum objective value for, respectively, a maximization or minimization problem are called *optimal solutions*. However, not all feasible

*The size of an instance is a fundamental concept in Computational Complexity Theory that requires many details to be defined precisely. For simplicity, we assume here that, for every optimization problem, there is an stipulated way of representing each of its instances as a string of characters, and the size of an instance is the length of the string representing it.

linear programs have optimal solutions. In some cases, given any feasible solution $\mathbf{x}$, it is always possible to find another feasible solution $\hat{\mathbf{x}}$ for which the objective value improves, that is, the objective value of $\hat{\mathbf{x}}$ is strictly greater/smaller than the objective value of $\mathbf{x}$ for the maximization/minimization linear program. If a feasible linear program does not have an optimal solution, then it is called *unbounded*, otherwise it is called *bounded*.

Associated with any linear program, there is a *dual program* for which the original problem is called the *primal program*. For a linear program in the standard form as previously defined, its associated dual program is:

$$\begin{aligned} \text{minimize} \quad & \mathbf{b}'\mathbf{y} \\ \text{subject to } A'\mathbf{y} &\geq \mathbf{c} \\ \mathbf{y} &\geq 0 \end{aligned} \tag{1.2}$$

where $\mathbf{y} \in \mathbb{R}^m$ is the vector of dual variables.

Program (1.2) can be written in standard form by multiplying matrix A and vectors $\mathbf{b}$ and $\mathbf{c}$ by -1:

$$\begin{aligned} \text{maximize} \quad & -\mathbf{b}'\mathbf{y} \\ \text{subject to } -A'\mathbf{y} &\leq -\mathbf{c} \\ \mathbf{y} &\geq 0 \end{aligned}$$

From the dual program in standard form, it is easy to show that the dual of (1.2) is (1.1), that is, the dual program of a dual program is the primal program.

Theorem 1.1.1 (Weak Duality)**.** *For each pair of feasible solutions $\boldsymbol{x}$ and $\boldsymbol{y}$ of the primal and dual linear programs* (1.1) *and* (1.2)*, respectively,* $\mathbf{c}'\mathbf{x} \leq \mathbf{b}'\mathbf{y}$.

Proof. Multiply $A\mathbf{x} \leq \mathbf{b}$ by $\mathbf{y}'$ to obtain $\mathbf{y}'\mathbf{b} \geq \mathbf{y}'A\mathbf{x} = (A'\mathbf{y})'\mathbf{x} \geq \mathbf{c}'\mathbf{x}$. □

In some cases, both primal and dual programs are infeasible and, thus, the Weak Duality Theorem holds trivially. In fact, this theorem shows that, whenever either program is unbounded, the other is necessarily infeasible. The next theorem gives a property of the only other possibility, when both programs have optimal solutions:

Theorem 1.1.2 (Strong Duality)**.** *If a linear program has an optimal solution $\boldsymbol{x} \in \mathbb{R}^n$, then the dual program has an optimal solution $\boldsymbol{y} \in \mathbb{R}^m$ such that* $\mathbf{c}'\mathbf{x} = \mathbf{b}'\mathbf{y}$.

For brevity, we omit the proof for the theorem above. The interested reader might see it in textbooks on the subject, for example [24].

The relationship $\mathbf{y}'\mathbf{b} \geq \mathbf{y}'A\mathbf{x} \geq \mathbf{c}'\mathbf{x}$ seen in the proof of Theorem 1.1.1 can be applied to Theorem 1.1.2 to help characterize optimal solutions:

Theorem 1.1.3 (Complementary slackness conditions)**.** *A pair of feasible solutions* $\mathbf{x}$ *and* $\mathbf{y}$ *for, respectively, the primal and dual linear programs* (1.1) *and* (1.2)*, is optimal if and only if:*

$$\forall i \in \{1, 2, \ldots, m\},\ y_i \left(b_i - \sum_{j=1}^{n} a_{ij} x_j \right) = 0$$

and

$$\forall j \in \{1, 2, \ldots, n\},\ x_j \left(c_j - \sum_{i=1}^{m} a_{ij} y_i \right) = 0$$

Proof. Assuming that $\mathbf{x}$ and $\mathbf{y}$ are optimal, we conclude from the expression $\mathbf{y}'\mathbf{b} \geq \mathbf{y}'(A\mathbf{x}) = (A'\mathbf{y})'\mathbf{x} \geq \mathbf{c}'\mathbf{x}$ that both $\mathbf{y}'(\mathbf{b} - A\mathbf{x}) = \mathbf{0}$ and $\mathbf{x}'(\mathbf{c} - A'\mathbf{y}) = \mathbf{0}$.

Conversely, assuming $\mathbf{y}'(\mathbf{c} - A\mathbf{x}) = \mathbf{0}$ and $\mathbf{x}'(\mathbf{b} - A'\mathbf{y}) = \mathbf{0}$, we reach the conclusion that $\mathbf{y}'\mathbf{c} = \mathbf{b}'\mathbf{x}$ which can only occur if both solutions are also optimal due to weak duality (see Theorem 1.1.1). $\square$

An alternative way of stating the complementary slackness conditions that will become useful for us is the following: a pair of feasible solutions $\mathbf{x}$ and $\mathbf{y}$ for, respectively, the primal and dual linear programs (1.1) and (1.2), is optimal if and only if:

$$\forall i \in \{1, 2, \ldots, m\},\ y_i > 0 \text{ implies } b_i = \sum_{j=1}^{n} a_{ij} x_j$$

and

$$\forall j \in \{1, 2, \ldots, n\},\ x_j > 0 \text{ implies } c_j = \sum_{i=1}^{m} a_{ij} y_i$$

that is, whenever a variable of either the primal or dual program is different than zero, its associated inequality in the other program is met with equality.

1.2 The Uncapacitated FLP

Now, we will review an algorithm by Jain and Vazirani [17] for the Uncapacitated FLP to illustrate how Linear Programming can be used to obtain approximation algorithms and to set the tone for the rest of this thesis.

The Uncapacitated FLP is a (minimization) Combinatorial Optimization Problem in which every instance $\mathcal{I}$ has a set of clients $\mathcal{C}$, a set of facilities $\mathcal{F}$, opening costs $\{f_i\}_{i \in \mathcal{F}}$ and a metric function $c : (\mathcal{F} \cup \mathcal{C})^2 \to \mathbb{Q}^+$. A function $c : U^2 \to \mathbb{Q}$ is said to be *metric* if and only if it is symmetric and it satisfies the triangle inequality, that is, $\forall i, j \in U : c_{ij} = c_{ji}$ (symmetry) and $\forall i, j, k \in U : c_{ij} \leq c_{ik} + c_{kj}$ (triangle inequality). A feasible solution of $\mathcal{I}$ is a non-empty subset $O \subseteq \mathcal{F}$ of facilities to be opened whose objective value is:

$$\sum_{i \in O} f_i + \sum_{j \in \mathcal{C}} \min_{i \in O} c_{ij}$$

In [17], the above problem is formulated as an *Integer Linear Program* (ILP), i.e. a

Linear Program in which variables can only assume integer values:

$$\min \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}, j\in\mathcal{C}} c_{ij} x_{ij} \tag{1.3}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{F}} x_{ij} = 1 \qquad \forall j \in \mathcal{C} \tag{1.4}$$

$$x_{ij} \le y_i \qquad \forall i \in \mathcal{F}, j \in \mathcal{C} \tag{1.5}$$

$$x_{ij}, y_i \in \{0,1\} \qquad \forall i \in \mathcal{F}, j \in \mathcal{C} \tag{1.6}$$

where, for each $i \in \mathcal{F}$ and $j \in \mathcal{C}$, y_i indicates whether facility i is open and x_{ij} indicates whether client j is connected to facility i. Expression (1.3) is the objective function, inequalities (1.4) assert that every client must have exactly one connection, inequalities (1.5) assert, that if a client is connected to a facility, that facility must be open, and propositions (1.6) determine that the variables can only assume values in the set $\{0,1\}$.

We can relax propositions (1.6) replacing them by $x_{ij}, y_i \ge 0, \forall i \in \mathcal{F}, j \in \mathcal{C}$ to obtain an LP. Define OPT_{ILP} and OPT_{LP} to be the objective value of an optimal solution for, respectively, the original ILP and the relaxed LP. Note that, since the feasible set for the ILP is contained in the feasible set for the LP, $\mathrm{OPT}_{LP} \le \mathrm{OPT}_{ILP}$.

The following is the dual program for the relaxed LP:

$$\max \quad \sum_{j\in\mathcal{C}} v_j \tag{1.7}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{C}} w_{ij} \le f_i \quad \forall i \in \mathcal{F} \tag{1.8}$$

$$v_j - w_{ij} \le c_{ij} \quad \forall i \in \mathcal{F}, j \in \mathcal{C} \tag{1.9}$$

$$w_{ij}, v_j \ge 0 \quad \forall i \in \mathcal{F}, j \in \mathcal{C}$$

In [17], the authors suggested the following interpretation for the dual variables: for each $j \in \mathcal{C}$, if j is connected to a facility i, v_j is the budget assigned to client j, w_{ij} is the part of its budget that client j uses to open facility i and the rest of the budget is used to pay the connection cost. It is also required that each open facility i is entirely payed by the clients assigned to it, that is $\sum_{j\in D(i)} w_{ij} = f_i$ where $D(i) \subseteq \mathcal{C}$ is the set of all clients assigned to facility i.

If we assume that there is an optimal solution (x, y) to the LP that is also integral, the interpretation is consistent: if client j is connected to facility i, i.e. $x_{ij} = 1$, then we have $y_i = 1$ due to (1.5) and, through complementary slackness, $\sum_{j'\in\mathcal{C}} w_{ij'} = f_i$ and $v_i = c_{ij} + w_{ij}$. Also, if $w_{ij} > 0$ for facility i and client j, then $x_{ij} = y_i$ implying that, if facility i is open, then client j must be connected to it. Thus, we conclude that, for each open facility i, $\sum_{j\in D(i)} w_{ij} = f_i$ in this case.

The algorithm builds a feasible solution to the dual LP, employing a *dual-ascent* strategy: in a simulated timeline, all dual variables start at 0, and, for each client $j \in \mathcal{C}$, variable v_j is increased uniformly until it becomes frozen. The timeline stops when all variables $\{v_j\}$ are frozen. Each facility is unopened at the start and can become tentatively open during the algorithm. The idea is to obtain a feasible dual solution whose value is a lower bound to the optimal integral solution. To guarantee feasibility, we must handle the following events:

1. If, for some client j and unopened facility i, $v_j = c_{ij}$, v_j cannot increase anymore without violating (1.9). For that reason, we start also increasing w_{ij} uniformly;

2. If, for some unopened facility i, $\sum_{j \in \mathcal{C}} w_{ij} = f_i$, we cannot increase variables $\{w_{ij}\}_{j \in \mathcal{C}}$ without violating (1.8). Thus, for each client j such that w_{ij} is increasing uniformly, we freeze v_j, stop increasing variables $\{w_{i'j}\}_{i' \in \mathcal{F}}$ and label facility i as tentatively open;

3. If, for some client j and a tentatively open facility i, $v_j = c_{ij}$, we cannot increase v_j or start increasing w_{ij}, so we freeze j and stop increasing variables $\{w_{i'j}\}_{i' \in \mathcal{F}}$.

We cannot simply open all tentatively open facilities since it could be the case that, for some facilities i and i' and a client j, $w_{ij}, w_{i'j} > 0$, implying that the same client would have to pay for both facilities, while j should only pay for the facility to which it is connected. Therefore, we will find and open a *maximal independent set* of tentatively opened facilities instead of simply opening all of them.

An *independent set* on a graph $G(V, E)$ is a set of vertices $I \subseteq V$ such that there are no vertices $u, v \in I$ that are adjacent. An independent set I is a *maximal independent set* if and only if there is no vertex $v \in V \setminus I$ such that $I \cup \{v\}$ is independent.

First, we define a graph G whose vertex set is the set of tentatively open facilities and connect facilities i and i' if and only if there is a client j with $w_{ij}, w_{i'j} > 0$. Next, create a set I, initially empty, by iterating over all vertices, and adding a vertex v to I only if there is no vertex u already in I such that v and u are connected.

Since no client contributes to two facilities of I, we can safely open all facilities in I. For each j such that there is $i \in I$ with $w_{ij} > 0$, we connect j to i. Also, for each unconnected client j such that there is $i \in I$ with $v_j = c_{ij}$, we choose one such facility i and connect j to i. If a client j was connected to some facility i at this point, we say that j is directly connected to i. Note that this implies $v_j - w_{ij} = c_{ij}$. Next, we connect all remaining clients to their closest facilities.

For all directly connected clients, we can express their distance to the facilities connected to them as a function of dual variables. The following lemma establishes a similar relationship for indirectly connected clients.

Lemma 1.2.1. *If client j is indirectly connected to facility i, then $c_{ij} \leq 3v_j$.*

Proof. Select the tentatively open facility i' that caused v_j to become frozen. If j was not directly connected, this implies that $i' \notin I$, implying that there is a facility $i \in I$ such that i' and i have an edge in G or, equivalently, there is client j' with $w_{ij'}, w_{i'j'} > 0$.

Let t be the time when i' became tentatively open. Either v_j became frozen at that time, when $v_j = t$, or v_j became frozen later because v_j reached $c_{i'j}$, thus $v_j \geq t$. In the case of client j', if $v_{j'}$ did not become frozen earlier than t, then $v_{j'}$ would have become frozen at t as i' became tentatively open since $w_{i'j'} > 0$, thus $v_{j'} \leq t \leq v_j$.

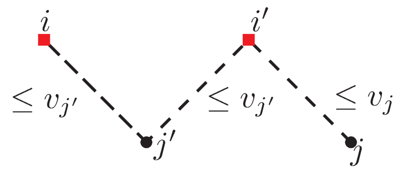

Figure 1.1: Illustration of Lemma 1.2.1

Because of the way we increase dual variables, we also know that:

$$\begin{aligned} v_{j'} - w_{ij'} &= c_{ij'} \leq v_{j'} \\ v_{j'} - w_{i'j'} &= c_{i'j'} \leq v_{j'} \\ v_j - w_{i'j} &= c_{i'j} \leq v_j. \end{aligned}$$

Suppose that j is connected to i, thus, we would have a connection cost of

$$c_{ij} \leq c_{i'j} + c_{i'j'} + c_{ij'} \leq 2v_{j'} + v_j \leq 3v_j,$$

by triangle inequality (see Figure 1.1). Since j will be connected to the closest open facility, its connection cost will be at most $c_{ij} \leq 3v_j$ as desired. $\square$

We can now show the approximation factor:

Theorem 1.2.2. *The algorithm described above is a 3-approximation of the UFLP.*

Proof. Let $\sigma : \mathcal{C} \to \mathcal{F}$ map each client j to the facility that j is connected to. The total cost of our solution is:

$$\sum_{i \in I} f_i + \sum_{j \in \mathcal{C}} c_{\sigma(j)j} \tag{1.10}$$

Let D be the set of directly connected clients. By the observations above, the fact that, for every $i \in I$, $f_i = \sum_{j \in \mathcal{C}} w_{ij}$, and Lemma 1.2.1, the above cost is at most:

$$\begin{aligned} &\sum_{i \in I} \sum_{j \in \mathcal{C}} w_{ij} + \sum_{j \in D} (v_j - w_{\sigma(j)j}) + \sum_{j \in \mathcal{C} \setminus D} 3v_j \\ =&\sum_{j \in D} w_{\sigma(j)j} + \sum_{j \in D} (v_j - w_{\sigma(j)j}) + \sum_{j \in \mathcal{C} \setminus D} 3v_j \\ \leq& 3 \sum_{j \in \mathcal{C}} v_j \leq 3\mathrm{OPT}_{LP} \leq 3\mathrm{OPT}_{ILP} \end{aligned}$$

We know that $\sum_{i \in I} \sum_{j \in \mathcal{C}} w_{ij} = \sum_{j \in D} w_{\sigma(j)j}$ because every client $j \in D$ only contributes to one facility in I, the one j is connected to, and no client in $\mathcal{C} \setminus D$ contributes to facilities in I. Also, we know that $\sum_{j \in \mathcal{C}} v_j \leq \mathrm{OPT}_{LP}$ by Weak Duality (Theorem 1.1.1). $\square$

Chapter 2

The k-Level Facility Location Problem

The metric k-Level Facility Location Problem (k-LFLP) is composed by a set of *clients* $\mathcal{C}$, a set of *facilities* $\mathcal{F} = \bigcup_{i=1}^{k} \mathcal{F}_i$, where $\mathcal{F}_i$ is the set of facilities in the i-th level, a *metric function* $d : (\mathcal{C} \cup \mathcal{F})^2 \to \mathbb{Q}$, and facility *opening costs* $\{f_i\}_{i \in \mathcal{F}}$. The problem consists in selecting a set $S = \bigcup_{l=1}^{k} S_l$ of facilities to *open*, where $S_l \subseteq \mathcal{F}_l$, and defining a map $\phi : \mathcal{C} \to S_1 \times S_2 \times \cdots \times S_k$ that associates every client j to a path $p = (i_1, i_2, \ldots, i_k) \in S_1 \times S_2 \times \cdots \times S_k$, such that the cost, given by the following expression, is minimized:

$$\sum_{i \in S} f_i + \sum_{j \in \mathcal{C}} d(j, \phi(j)),$$

where, for all clients $j \in \mathcal{C}$ and paths $p = (i_1, i_2, \ldots, i_k)$, $d(j, p) = d(j, i_1) + \sum_{l=2}^{k} d(i_{l-1}, i_l)$.

In this chapter, we will briefly review the history of approximation algorithms for the k-LFLP in Section 2.1, and describe, in Section 2.2, the $(3 - o(1))$-approximation for the k-LFLP by Byrka and Rybicki [9] that we will later adapt for an stochastic variation of the k-LFLP in Chapter 4.

2.1 Literature overview

For the UFLP and, as a consequence, for the k-LFLP, Guha and Khuller [14] demonstrated that there is no algorithm with approximation ratio smaller than 1.463 if NP $\not\subseteq$ DTIME$(n^{O(\log \log n)})$,* which Sviridenko (see the notes by Vygen [30]) strengthened by replacing the assumption by $P \neq$ NP. Additionally, Krishnaswamy and Sviridenko [19] proved that there is no algorithm for the k-LFLP with approximation ratio better than 1.61 for arbitrary k and 1.539 for $k = 2$ unless NP $\subseteq$ DTIME$(n^{O(\log \log n)})$.

Shmoys, Tardos and Aardal [27] created the first approximation algorithm for the 2-LFLP which uses a randomized filtering strategy based on results of Lin and Vitter [20, 21], obtaining an approximation ratio of 3.16. First, they introduced an algorithm that, given a parameter $\alpha \in (0, 1)$ obtains a solution of cost at most $\max\left\{\frac{1}{\alpha}, \frac{3}{1-\alpha}\right\}$ times the optimal value. This algorithm filters an optimal fractional solution of the instance,

*Given a function $f(n)$, DTIME$(f(n))$ roughly represent the set of decision problems for which there is an algorithm that takes at most $O(f(n))$ steps for any instance of size n. Decision problems are problems that ask a yes-or-no question.

by removing, for every client j, all fractional connections to paths ikj ($i \in \mathcal{F}_2, k \in \mathcal{F}_1$) whose length is larger than a value c_j that depends on parameter α, and scaling variables to ensure feasibility. The filtered fractional solution is later rounded in a way that total opening cost is at most the opening cost of the fractional solution and, as a result of triangle inequality, total connection cost is at most three times the fractional connection cost. The resulting solution has opening cost at most $\frac{1}{\alpha}F$ and connection cost at most $\frac{3}{1-\alpha}C$, where F and C are the opening and connection costs of the optimal fractional solution, respectively. This is later improved by randomizing parameter α, selecting it uniformly from the interval $(\beta, 1)$, where $\beta \in (0, 1)$ is a parameter of this randomization. Their analysis reveals that such procedure obtains an approximation factor of $\max\left\{\frac{\ln(1/\beta)}{1-\beta}, \frac{3}{1-\beta}\right\}$, which is less than 3.16 for $\beta = 1/e^3$.

Aardal, Chudak and Shmoys [1] gave the first approximation algorithm for general k, achieving an approximation factor of 3 and improving on the result of [27] for the 2-LFLP. Their algorithm rounds an LP relaxation by clustering clients and opening one random path of bounded length for all clients in the same cluster.

There are also algorithms for the k-LFLP that do not require solving linear programs, which is a very time-consuming process in practice especially for large k. Meyerson, Mungala, and Plotkin [23] gave the first of these algorithms, obtaining an approximation factor of $O(\log|\mathcal{C}|)$ based on their approximation algorithm to the COST-DISTANCE problem. Then, Guha, Meyerson, and Mungala [15] obtained the first combinatorial approximation algorithm with a constant approximation factor, namely 9.2, followed by Bumb and Kern's [5] dual-ascent algorithm which was a 6-approximation (the dual-ascent technique will be further discussed in Section 3.2). Ageev [3] observed that any ρ-approximation for the UFLP can be used to produce a 3ρ-approximation for the k-LFLP by a simple reduction which, in combination with Mahdian, Ye, and Zhang [22] 1.517-approximation for the UFLP, yields a 4.56-approximation for the k-LFLP. Finally, Ye and Zhang [2] obtained a parametrizable reduction from the k-LFLP to the UFLP reminiscent of Ageev's reduction that, combined with the algorithm from [22], obtains a 3.27-approximation. They were also able to achieve better approximation factors for small k by combining their original algorithm with a recursive algorithm, obtaining, for example, factors 2.422, 2.845, and 3.057 for $k = 2$, $k = 3$, and $k = 4$. Note that, for $k = 2$ and $k = 3$, their algorithm was an improvement over the 3-approximation by Aardal, Chudak and Shmoys.

Later, Zhang [39] improved the results for $k \leq 4$. First, he obtained a 1.77-approximation for the two-level version of the problem adapted from the dual-ascent method of [22]. In this algorithm, as clients' "budgets" are increased, part of them are "offered" to open second-level facilities. Those offers take into account the cost of opening first-level facilities needed to connect each client to the second-level facility being considered, and are realized as soon as the amount offered to a facility is enough to cover its opening cost, opening all first-level facilities involved in the transaction as well as the second-level facility. However, while in the UFLP version of this algorithm of [22], the amount destined to open a facility is known exactly, in this adaptation for the 2-LFLP, one can only efficiently obtain an estimate. This lack of information reflects on the approximation factor that goes from 1.51 for the UFLP to 1.77 in the 2-LFLP. Also, the estimation mechanism requires rounding an LP relaxation, making the dual-ascent algorithm not fully combinatorial as in the UFLP case. Zhang shows that this improved approximation for the 2-LFLP can be directly combined with results from [2] to yield a 2.51-approximation for the 3-LFLP. Zhang also combines the results from [2] with his ideas to obtain a 2.81-approximation for the 4-LFLP. Later results by Byrka and Aardal [6] for the UFLP in combination with Zhang's algorithm allowed a small decrease in the approximation factor for the 3-LFLP, resulting in a 2.492-approximation.

Byrka and Rybicki [9] obtained an algorithm based on a new ILP formulation. The algorithm scales fractional opening by a parameter $\gamma \geq 1$, and employs a clever LP rounding technique in trees, adapted from the results by Garg, Konjevod and Ravi [13] for the Group Steiner Tree Problem. They improve the approximation factors for all $k > 2$, which became 2.02 for $k = 3$, 2.14, for $k = 4$ and 2.24 for $k = 5$, and converges to 3 as k goes to infinity. This algorithm is covered in detail in Section 2.2.

Finally, Byrka, Li and Rybicki [8] adapted the algorithm from [9] to a generalization of the k-LFLP known as the Prize Collecting k-LFLP where, instead of necessarily connecting every client to a path, every unconnected client in a solution adds a penalty cost. An instance of the k-LFLP can be represented as an instance of that generalization by simply setting penalties to prohibitively high values. By running the algorithm for certain scaling parameters and taking the best solution, they obtained better approximation ratios for all $k > 2$. For example, for $k = 3$, $k = 4$ and $k = 5$, the ratios were 1.97, 2.09 and 2.19 respectively.

2.2 A $(3 - o(1))$-approximation for the k-LFLP

In this section, we review the $(3 - o(1))$-approximation for the k-LFLP by Byrka and Rybicki [9] for constant k. Their algorithm uses an ILP formulation that "creates" many copies of the facilities and organizes them in a forest structure, allowing an adaptation of a procedure described in [13] to be later utilized for rounding the LP relaxation. Although the authors also presented a way to use cost scaling to improve the approximation factor, we will not discuss it here.

2.2.1 ILP formulation

The basic idea of the ILP formulation is to create a forest $\mathcal{R}$ whose nodes represent paths starting at a l-th level facility, ending in a k-th level facility and passing through facilities of all levels in-between in ascending order. Symbolically, let $\mathcal{P}_l$ represent such paths starting at a l-th level facility, that is $\mathcal{P}_l = \mathcal{F}_l \times \mathcal{F}_{l+1} \times \cdots \times \mathcal{F}_k$, also let $\mathcal{P}_F = \bigcup_{l=1}^{k} \mathcal{P}_l$ represent all possible paths/nodes.

All paths $p = (i_l, i_{l+1}, \ldots, i_k) \in \mathcal{P}_F$ are also interpreted as copies of facility i_l with the additional constraint, in the case $l < k$, that all clients served by this copy must also be served by the copy of facility i_{l+1} associated with the path $p' = (i_{l+1}, i_{l+2}, \ldots, i_k)$. As a consequence, there is the natural notion of node p' being considered the *parent* of node p.

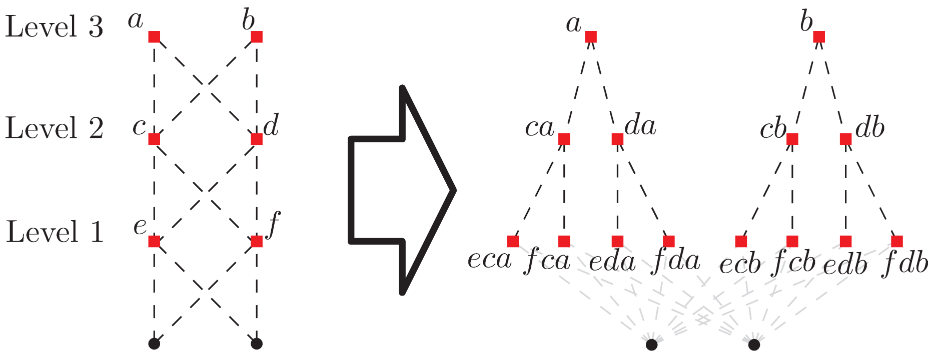

Figure 2.1: An example of an instance with 3 levels. On the left, the original instance, and, on the right, the resulting forest. Red squares represent facilities and black circles represent clients. Dashed lines represent possible connections.

This relationship is captured by the function $\pi : \mathcal{P}_F \setminus \mathcal{P}_k \to \mathcal{P}_F$:

$$\pi(i_l, i_{l+1}, \ldots, i_k) = (i_{l+1}, i_{l+2}, \ldots, i_k)$$

where $(i_l, i_{l+1}, \ldots, i_k) \in \mathcal{P}_l$ with $l < k$. Thus, function π directs every edge in a tree of $\mathcal{R}$ towards the k-th level facility i_k in that tree, making i_k the *root* of that tree. See Figure 2.1.

In the formulation, every node $p = (i_l, i_{l+1}, \ldots, i_k) \in \mathcal{P}_F$ is associated with the *opening cost* c_p, which is the same as the opening cost of i_l, as p is interpreted as a copy of i_l. In addition to the forest $\mathcal{R}$, the formulation also considers paths starting at clients and passing through facilities spanning all levels in ascending order. Those paths represent the assignment of clients to paths in $\mathcal{P}_1$, and the set of all of them is denoted by $\mathcal{P}_C \triangleq \mathcal{C} \times \mathcal{P}_1$. "Opening" one path $(j, p') \in \mathcal{P}_C$, or equivalently assigning j to p', results in opening the node $p' \in \mathcal{P}_1$, which, in turn, implies in opening all p''s ancestors. The cost c_p for paths $p = (j, p') \in \mathcal{P}_C$ is given by the length of p, which is $d(j, p')$.

The resulting ILP for the k-LFLP is:

$$\min \quad \sum_{p \in \mathcal{P}} c_p x_p \tag{2.1}$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}_C : j \in p} x_p \geq 1 \qquad \forall j \in \mathcal{C} \tag{2.2}$$

$$\sum_{p \in \mathcal{P}_C : j \in p, p \sqsupset q} x_p \leq x_q \qquad \forall j \in \mathcal{C},\ \forall q \in \mathcal{P}_F \tag{2.3}$$

$$x_p \leq x_{\pi(p)} \quad \forall p \in \mathcal{P}_F \setminus \mathcal{P}_k \tag{2.4}$$

$$x_p \in \{0, 1\} \quad \forall p \in \mathcal{P} \tag{2.5}$$

where $p \sqsupset q$ denotes that tuple q is a proper suffix of tuple p and $\mathcal{P} = \mathcal{P}_F \cup \mathcal{P}_C$. For each $p = (i, p') \in \mathcal{P}_C$, variable x_p indicates whether client j is served by a path $p' \in \mathcal{P}_1$, that is, whether the elected solution has $\phi(j) = p'$. For each $p \in \mathcal{P}_F$, variable x_p indicate whether the node p is opened. Inequalities (2.2) state that every client $j \in \mathcal{C}$ must be assigned to at least one path (in optimal solutions, clients are assigned to exactly one path). Inequalities (2.3) state that, if the path $q \in \mathcal{P}_F$ is a suffix of an open path $p \in \mathcal{P}_C$ starting at a client $j \in \mathcal{C}$, then q must also be opened. Finally, Inequalities (2.4) state that, if a non-root node is opened, then its parent must also be opened. In the relaxed LP, integrality constraints (2.5) are replaced by $x_p \geq 0$, for all $p \in \mathcal{P}$.

This linear program has $O(mn^k)$ inequalities, where n is the number of facilities per level, m is the number of clients and k is a fixed constant. Therefore solving the LP takes polynomial time.

The dual program of the relaxed LP is:

$$
\begin{aligned}
\max \quad & \sum_{j \in \mathcal{C}} v_j \\
\text{s.t.} \quad & v_j - \sum_{q \in \mathcal{P}_F : q \sqsupset p} w_{j,q} \leq c_p && \forall p = (j, \dots) \in \mathcal{P}_C \\
& -y_p + \sum_{j \in \mathcal{C}} w_{j,p} \leq c_p && \forall p \in \mathcal{P}_1 \\
& -y_p + \sum_{q \in \mathcal{P}_{l-1} : p \sqsupset q} y_q + \sum_{j \in \mathcal{C}} w_{j,p} \leq c_p && \forall l \in \{2, \dots, k-1\}, p \in \mathcal{P}_l \\
& \sum_{q \in \mathcal{P}_{k-1} : p \sqsupset q} y_q + \sum_{j \in \mathcal{C}} w_{j,p} \leq c_p && \forall p \in \mathcal{P}_k \\
& v_j, w_{j,p}, y_p \geq 0 && \forall p, j
\end{aligned} \tag{2.6}
$$

The Strong Duality Theorem (see Theorem 1.1.2) states that the value of an optimal solution to a primal program (in this case, the relaxed LP) is equal to the value of an optimal solution to the corresponding dual program. Thus, we can utilize optimal solutions to both primal and dual programs to bound the cost of the solution obtained by the algorithm. This algorithm does not compute the dual solution directly, it only rounds a primal (fractional) solution. However, the following lemma provides a link to the dual program that will be later utilized in the analysis:

Lemma 2.2.1. *Let* $\mathbf{x}$ *be an optimal solution to the relaxed LP and* $(\mathbf{v}, \mathbf{w}, \mathbf{y})$ *an optimal solution to the dual program. For every path* $p = (j, i_1, i_2, \dots, i_k) \in \mathcal{P}_C$ *such that* $x_p > 0$*, the length of* p *is at most* v_j*.*

Proof. By optimality of both solutions and complementary slackness (see Theorem 1.1.3), we know that, whenever $x_p > 0$, Inequality (2.6) is met with equality. Symbolically, $x_p > 0$ implies in $c_p - v_j + \sum_{q \in \mathcal{P}_F : q \sqsupset p} w_{j,q} = 0$.

Thus, $c_p = v_j - \sum_{q \in \mathcal{P}_F : q \sqsupset p} w_{j,q}$, where c_p is the length of the path p as previously defined. The result follows directly from the fact that, for all $j \in \mathcal{C}$ and $q \in \mathcal{P}_F$, $w_{j,q}$ is non-negative. □

2.2.2 Clustering

The algorithm executes a randomized procedure that rounds the fractional solution to the relaxed LP, selecting facilities to open in the final integral solution. However, before rounding the LP, it executes an auxiliary *clustering* procedure that partitions clients in *clusters*. Later, each cluster receives from the randomized opening procedure a backup path of bounded length (in expectation) that serves as an worst-case path to clients that were "unlucky," that is, clients whose fractionally connected sets were not opened by the algorithm.

Now, we define more precisely the clustering procedure. Let $\mathbf{x}$ be the obtained optimal solution to the relaxed LP and let $N(j) \triangleq \{p \in \mathcal{P}_F : \exists q \in \mathcal{P}_1, x_{jq} > 0 \text{ and } q \sqsupseteq p\}$ be the *neighborhood* of a client $j \in \mathcal{C}$, where "$a \sqsupseteq b$" means that either $a \sqsupset b$ or $a = b$. We will refer to nodes of certain level in the neighborhood by $N_\ell(j) \triangleq N(j) \cap \mathcal{P}_\ell$. Define[†] $d^{av}(j) \triangleq \sum_{p \in \mathcal{P}_1} x_{jp} c_{jp} = \sum_{p \in N_1(j)} x_{jp} c_{jp}$ and $d^{\max}(j) \triangleq \max\{c_{jp} : p \in N_1(j)\}$. The clustering procedure works in the following way:

procedure `Clustering()`:
- $\mathcal{C}' \leftarrow \mathcal{C}$
- **while** $\mathcal{C}' \neq \emptyset$ **do**
  - $j \leftarrow \arg\min_{j' \in \mathcal{C}'} d^{\max}(j') + d^{av}(j')$
  - $C \leftarrow \{j' \in \mathcal{C}' : N(j') \cap N(j) \neq \emptyset\}$
  - Add cluster C and make j its center
  - $\mathcal{C}' \leftarrow \mathcal{C}' \setminus C$
- **end**

After the clustering procedure, each cluster randomly selects exactly one root in the neighborhood of its center and places one *token* in it. That token will be later utilized by the randomized facility opening procedure (see Section 2.2.3) to ensure that a backup path of bounded length is opened for each cluster. Tokens are distributed in the following way: for each cluster C with center j, randomly select one node in $N_k(j)$, assigning to each node $r \in N_k(j)$ probability $\sum_{p \in \mathcal{P}_1 : p \sqsupseteq r} x_{jp}$, and place a token in the selected node. Notice that, for every pair of cluster centers j and j', $N_k(j) \cap N_k(j') = \emptyset$ and, for that reason, it is impossible for a root to receive tokens from two different cluster centers.

2.2.3 Randomized Facility Opening

Now, we describe the randomized facility opening procedure.

Notice that each node $v \in \mathcal{P}_F$ belongs to the neighborhood of at most one cluster center. For every node $v \in \mathcal{P}_F$, let variable x_v represent the fractional opening given by the LP solution, and y_v represent "how much" the cluster center associated with v "uses" v, that is:

$$y_v = \begin{cases} \sum_{p \in \mathcal{P}_C : j \in p, p \sqsupseteq v} x_{jp} & \text{if } v \text{ is in the neighborhood of some cluster center } j \\ 0 & \text{otherwise} \end{cases}$$

[†]Here we denote (j, p) as jp for simplicity.

`Round()` is a recursive random procedure that reads the rational vector $(\mathbf{x}, \mathbf{y})$ and sets a binary vector $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ such that, for every node $v \in \mathcal{P}_F$, $\hat{x}_v = 1$ if and only if facility v is opened and $\hat{y}_v = 1$ if and only if facility v belongs to the backup path of its associated cluster center.

Initially, $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \mathbf{0}$ and `Round()` is executed on every root of forest $\mathcal{R}$. An execution on a node $v \in \mathcal{P}_F$ may recursively call `Round()` on every *child* of v, where u is a child of v if and only if v is the parent of u.

Let $C(v) \triangleq \{u \in \mathcal{P}_F : \pi(u) = v\}$ be the set of children of node v. The pseudocode for the procedure is presented below:

```
1  procedure Round(v):
2      if v has a token then
3          x̂_v ← 1, ŷ_v ← 1
4          if v is not a leaf then
5              randomly select one node u ∈ C(v), assigning to each i ∈ C(v)
               probability y_i/y_v of being chosen
6              give token to node u
7              for i ∈ C(v) do
8                  Round(i)
9              end
10         end
11     else
12         if v is a root node then
13             x_pred ← 1
14         else
15             x_pred ← x_π(v)
16         end
17         rnd ← random number uniformly distributed in [0, 1]
18         if rnd ≤ (x_v − y_v)/(x_pred − y_v) then
19             x̂_v ← 1
20             if v is not a leaf then
21                 for i ∈ C(v) do
22                     Round(i)
23                 end
24             end
25         end
26     end
```

Some observations are in order to ensure that the numerical expressions do not lead to failures in lines 5 and 18.

First, consider the expression in line 5. For each $i \in C(v)$, $y_i \geq 0$ and $\sum_{i \in C(v)} y_i = y_v$, thus assigning probability y_i/y_v to each $i \in C(v)$ defines a proper probability distribution, except, possibly, if $y_v = 0$. However y_v cannot be 0: if v were a root node, its probability of receiving a token would have been $y_v = 0$; similarly, if v were not a root, the probability of receiving a token given that its parent had received it (which is the only possibility) would have been $y_v/y_{\pi(v)} = 0$ if $y_{\pi(v)} \neq 0$. If $y_{\pi(v)}$ were 0, the argument can be made inductively to show that neither $\pi(v)$ nor v would have received a token.

Now, consider the expression in line 18. Inequality (2.4) guarantees that $x_v \leq x_{\pi(v)} = x_{pred}$, when v is not a root, and, whenever v is a root, we may assume that $x_v \leq 1 = x_{pred}$ due to the optimality of $\mathbf{x}$. Consequently, we use this fact together with Inequality (2.3), which, in turn, assures that $y_v \leq x_v$, to conclude that $0 \leq \frac{x_v - y_v}{x_{pred} - y_v} \leq 1$ whenever $x_{pred} \neq y_v$. We can also show that x_{pred} must be different from y_v and, thus, $0 \leq \frac{x_v - y_v}{x_{pred} - y_v} \leq 1$ holds unconditionally. Assume that $x_{pred} = y_v$. Either v is a root with $y_v = 1$, and thus should have had received a token, or $y_v = x_{\pi(v)}$. In the latter case, Inequalities (2.3) and (2.4) imply that $y_v = y_{\pi(v)} = x_{\pi(v)}$ and that the probability of receiving a token is $y_v/y_{\pi(v)} = 1$. Both cases lead to a contradiction since line 18 cannot be executed when v receives a token and, thus, the assumption that $x_{pred} = y_v$ is always false.

Now, we turn to the probabilistic properties of the generated integral solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$.

Lemma 2.2.2. *For every cluster center j, exactly one backup path is opened in $N_1(j)$.*

Proof. Notice that exactly one token is placed in one of the roots in the neighborhood of j. Also, whenever the token is received at some node v, that node is necessarily opened (indicated by setting $\hat{x}_v$ to 1), and it belongs to the backup path of the cluster (indicated by setting $\hat{y}_v$ to 1). Finally, every node v that receives a token sends it to exactly one of its children, opening, in this way, all facilities in a path p from the root to the leaves. The path p thus becomes the backup path of that cluster. □

Lemma 2.2.3. *$E[\hat{y}_v] = y_v$ for all $v \in \mathcal{P}_F$.*

Proof. First, since $\hat{y}_v$ is either 0 or 1, $E[\hat{y}_v] = Pr[\hat{y}_v = 1]$.

Now, let us prove the lemma by induction on the distance from the node to the root of its tree. In the basis, which occurs when v is a root, the probability of $\hat{y}_v$ being 1 is given by the probability of it being chosen during the token distribution, which is either $\sum_{p \in \mathcal{P}_1 : p \sqsupseteq r} x_{jp} = y_v$, when v is associated with a cluster center j, or $0 = y_v$ otherwise.

When v is not a root, we can apply conditional probability to obtain that

$$Pr[\hat{y}_v = 1] = Pr[\hat{y}_v = 1 | \hat{y}_{\pi(v)} = 1] Pr[\hat{y}_{\pi(v)} = 1] + Pr[\hat{y}_v = 1 | \hat{y}_{\pi(v)} = 0] Pr[\hat{y}_{\pi(v)} = 0]$$

We known that $Pr[\hat{y}_v = 1 | \hat{y}_{\pi(v)} = 0] = 0$ and, by induction hypothesis, $Pr[\hat{y}_{\pi(v)} = 1] = y_{\pi(v)}$. Finally, $Pr[\hat{y}_v = 1 | \hat{y}_{\pi(v)} = 1]$ is $y_v/y_{\pi(v)}$ since this is the probability of the token being passed from $\pi(v)$ to v (see line (5) of `Round()` and the lemma follows. □

From the guarantees provided by both Lemmas 2.2.2 and 2.2.3, we fully determine the probability distribution of the selected backup path in a cluster. Let j be a cluster center, and B_j be the random variable that represent the backup path selected by j. We can conclude that, for all $p \in N_1(j)$, $Pr[B_j = p] = y_p$. Thus, we can present the observation in Section 2.2.2 that led to Lemma 2.2.5 in the following lemma:

Lemma 2.2.4. *The expected connection cost of a cluster center j to the backup path is at most $d^{av}(j)$.*

Proof. Due to Inequality 2.3, for all $p \in \mathcal{P}_1$, $y_p \leq x_{jp}$. Thus,

$$E[d(j, B_j)] = \sum_{p \in N_1(j)} d(j,p) y_p \leq \sum_{p \in \mathcal{P}_1} c_{jp} x_{jp} = d^{av}(j) \qquad \square$$

In the case of a non-central client j in cluster C, instead of simply obtaining an upper bound to the expected cost of connecting j to the backup path of C, we obtain an upper bound to the cost of connecting j to the backup path given that no node in $N_1(j)$ was opened. This is necessary because, to analyze the expected connection cost of j, j might use the backup path only if no path in $N_1(j)$ was opened, since using a path of $N_1(j)$ yields a cheaper connection cost in general.

Lemma 2.2.5. *Let j be a non-central client. If no path in $N_1(j)$ was completely opened, then the expected connection cost for j is at most $2d^{\max}(j) + d^{av}(j)$.*

Proof. Let j' be the center of the cluster C to which j belongs, $b \in N_1(j')$ be the backup path opened in C and D be the event of no path in $N_1(j)$ being completely open. We know that $N(j') \cap N(j)$ must be non-empty due to the way C was constructed. There are two possible cases:

1. There is $\ell \in N(j') \cap N(j)$ and a path $p' \in N_1(j')$ such that $p' \sqsupseteq \ell$ and $d(j', p') \leq d^{av}(j')$: in that case, by the triangle inequality, $d(j,b) \leq d(j,p) + d(j',p') + d(j,b) \leq d^{\max}(j) + d^{av}(j') + d^{\max}(j')$, where p is any path in $N_1(j)$ with $p \sqsupseteq \ell$; by construction of C, $d^{av}(j') + d^{\max}(j') \leq d^{av}(j) + d^{\max}(j)$, thus $d(j,b) \leq 2d^{\max}(j) + d^{av}(j)$;

2. For every node $\ell \in N(j') \cap N(j)$ and every path $p' \in N_1(j')$ such that $p' \sqsupseteq \ell$, $d(j',p') > d^{av}(j')$: in that case, $E[d(j',b)|D] \leq E[d(j',b)] = d^{av}(j')$, thus $E[d(j,b)|D] \leq d(j,p) + d(j',p') + E[d(j',b)|D] \leq d^{\max}(j) + d^{\max}(j') + d^{av}(j') \leq 2d^{max}(j) + d^{av}(j)$, where p is any path in $N_1(j)$ such that $p \sqsupseteq \ell$ and p' is any path in $N_1(j')$ such that $p' \sqsupseteq \ell$. $\square$

Next, we obtain the expectation of $\hat{x}_v$ for all $v \in \mathcal{P}_F$ so we can establish the expected total facility opening cost and a lower bound to the probability of an entire path in $\mathcal{P}_1$ in the neighborhood of each non-central client being entirely opened.

Lemma 2.2.6. *$E[\hat{x}_v] = x_v$, for all $v \in \mathcal{P}_F$.*

Proof. It is sufficient to show that $E[\hat{x}_v - \hat{y}_v] = x_v - y_v$ and then utilize this result together with Lemma 2.2.3 to conclude this lemma.

Let us prove it by induction on the distance from v to the root of its tree. In the base case, when v is a root, we conclude that $E[\hat{x}_v - \hat{y}_v] = Pr[\hat{x}_v = 1, \hat{y}_v = 0]$ since $\hat{x}_v \geq \hat{y}_v$ and both variables only assume values 0 and 1. Thus,

$$\begin{aligned} Pr[\hat{x}_v = 1, \hat{y}_v = 0] &= Pr[\hat{x}_v = 1 | \hat{y}_v = 0] Pr(\hat{y}_v = 0) \\ &= \frac{x_v - y_v}{1 - y_v}(1 - y_v) = x_v - y_v \qquad \text{(see Line 18 of } \texttt{Round()}\text{)} \end{aligned}$$

Now, in the induction step, assume that v is a non-root node and that $E[x_{\pi(v)}] = x_{\pi(v)}$. As before, $E[\hat{x}_v - \hat{y}_v] = Pr[\hat{x}_v = 1, \hat{y}_v = 0]$, but we must also relate this expression to $\hat{x}_{\pi(v)}$ so we can use the induction hypothesis. Luckily, v can only be open if $\pi(v)$ is open, so:

$$\begin{aligned} E[\hat{x}_v - \hat{y}_v] &= Pr[\hat{x}_v = 1, \hat{x}_{\pi(v)} = 1, \hat{y}_v = 0] \\ &= Pr[\hat{x}_v = 1 | \hat{x}_{\pi(v)} = 1, \hat{y}_v = 0] Pr[\hat{x}_{\pi(v)} = 1, \hat{y}_v = 0] \end{aligned} \tag{2.7}$$

The event $\hat{y}_v = 1$ implies $\hat{x}_{\pi(v)} = 1$, i.e. the event $\hat{y}_v = 1$ is completely contained in the event $\hat{x}_{\pi(v)} = 1$, that means that $Pr[\hat{x}_{\pi(v)} = 1, \hat{y}_v = 0] = Pr[\hat{x}_{\pi(v)} = 1] - Pr[\hat{y}_v = 1]$, which, due to the induction hypothesis and Lemma 2.2.3, is $x_{\pi(v)} - y_v$.

To obtain $Pr[\hat{x}_v = 1 | \hat{x}_{\pi(v)} = 1, \hat{y}_v = 0]$, we can deduce that v must not have a token if $\hat{y}_v = 0$ and conclude that the only way of $\hat{x}_v$ being set to 1 is if line 19 of `Round()` gets executed, which occurs with probability $\dfrac{x_v - y_v}{x_{\pi(v)} - y_v}$.

Substitute those expressions on (2.7) to obtain the desired result. □

2.2.4 Analysis

We can now obtain the approximation factor of the algorithm.

For any client j, let $F_k(\gamma)$ be the greatest lower bound to the probability of any facility of a path in $N_1(j)$ being opened, where γ is the total fractional utilization of paths by j, given by the expression $\sum_{p \in N_1(j)} x_{jp}$. If there is a path in $N_1(j)$ that is opened, simply connect j to the path in $N_1(j)$ that would result in the smallest connection cost, otherwise connect j to the backup path of its cluster. The following lemma bounds the expected connection cost for j:

Lemma 2.2.7. *For every client j, let P_j be the path for which j is assigned according to the strategy above. Then $E[d(j, P_j)] \leq F_k(\gamma) d^{av}(j) + (1 - F_k(\gamma))(2d^{\max}(j) + d^{av}(j))$, where $\gamma = \sum_{p \in N_1(j)} x_{jp}$.*

Proof. If j is a cluster center, this result is a corollary to Lemma 2.2.4. Otherwise, client j is non-central and either no path in $N_1(j)$ is entirely opened, which occurs with probability q for some $q \in [0, 1]$, or some path in $N_1(j)$ is opened. In the latter case, the expected connection cost is at most $d^{av}(j)$ because the probability of a path from $N_1(j)$ being opened is proportional to its factional opening.

Let D be the event where no path in $N_1(j)$ is opened, and P^j be the random variable that represents the path selected by j, then:

$$\begin{aligned} E[d(j, P^j)] &\leq (1 - q) d^{av}(j) + q E[d(j, P^j) | D] \\ &\leq (1 - q) d^{av}(j) + q(2d^{\max}(j) + d^{av}(j)) \text{ (see Lemma 2.2.5)} \end{aligned}$$

Finally, since $F_k(\gamma)$ is a lower bound to $1 - q$:

$$E[d(j, P^j)] \leq F_k(\gamma) d^{av}(j) + (1 - F_k(\gamma))(2d^{\max}(j) + d^{av}(j))$$ □

Now, we can obtain the approximation factor of the algorithm in terms of $F_k(1)$:

Theorem 2.2.8. *The algorithm is a* $(3 - 2F_k(1))$ *probabilistic approximation for the k-LFLP.*

Proof. Let $P^j : \Omega \to \mathcal{P}_1$ be the random variable that represents the path client j is connected to in the random solution obtained by the algorithm. Thus, the expected cost is:

$$\begin{aligned} E[\text{cost}] &= E\left[\sum_{p\in\mathcal{P}_F} c_p\hat{x}_p + \sum_{j\in\mathcal{C}} d(j, P^j)\right] \\ &= \sum_{p\in\mathcal{P}_F} c_p E[\hat{x}_p] + \sum_{j\in\mathcal{C}} E[d(j, P^j)] \end{aligned}$$

Due to Lemma 2.2.6 and Lemma 2.2.7, we have:

$$E[\text{cost}] \leq \sum_{p\in\mathcal{P}_F} c_p x_p + \sum_{j\in\mathcal{C}} (F_k(1)d^{av}(j) + (1 - F_k(1))(2d^{\max}(j) + d^{av}(j))).$$

We can evoke Lemma 2.2.1 to conclude that $d^{\max}(j) \leq v_j$, and obtain

$$\begin{aligned} E[\text{cost}] &\leq \sum_{p\in\mathcal{P}_F} c_p x_p + \sum_{j\in\mathcal{C}} d^{av}(j) + 2(1 - F_k(1)) \sum_{j\in\mathcal{C}} v_j \\ &= \sum_{p\in\mathcal{P}_F} c_p x_p + \sum_{j\in\mathcal{C}, p\in\mathcal{P}_1} c_{jp}x_{jp} + 2(1 - F_k(1)) \sum_{j\in\mathcal{C}} v_j \\ &\leq \text{OPT}^* + (2 - 2F_k(1))\text{OPT}^* \\ &\leq \text{OPT} + (2 - 2F_k(1))\text{OPT} = (3 - 2F_k(1))\text{OPT}, \end{aligned}$$

where OPT^* is the optimal value for the relaxed LP and OPT is the optimal value for the ILP. □

To compute an approximation factor for each value of k, it is not necessary to compute $F_k(1)$ exactly, any lower bound to $F_k(1)$ can be used instead. Later we will obtain one such lower bound with Theorem 2.2.13, but first we will describe some well known results.

Lemma 2.2.9. *Given any real sequence* $\{a_i\}_{i=1}^n$ *such that* $\sum_{i=1}^n a_i \geq 0$,

$$a_1 a_2 \ldots a_n \leq \left(\frac{a_1 + a_2 + \cdots + a_n}{n}\right)^n$$

Proof. Proof by induction on the size of the sequence.

In case $n = 1$, the Lemma is trivially true.

Assume the Lemma holds for sequences of size n. Consider a real sequence $\{a_i\}_{i=1}^{n+1}$ of size $n + 1$ such that $\sum_{i=1}^{n+1} a_i \geq 0$. Let $\mu = \frac{1}{n+1}\sum_{i=1}^{n+1} a_i \geq 0$. Without loss of generality, make $a_{n+1} \geq \mu$ and $a_n \leq \mu$ (there are always two elements with those properties, or the sum of elements would have to be either strictly above or strictly below $(n + 1)\mu$, which would be a contradiction). Now consider the auxiliary sequence $\{a'_i\}_{i=1}^n$ where, for $i < n$, $a'_i = a_i$ and $a'_n = a_{n+1} + a_n - \mu$. We have that $\sum_{i=1}^n a'_i = \sum_{i=1}^{n+1} a_i - \mu = (n+1)\mu - \mu \geq 0$, thus we can apply the induction hypothesis on $\{a'_i\}$:

$$\left(\frac{a_1 + a_2 + \cdots + a_n + a_{n+1} - \mu}{n}\right)^n = \mu^n \geq a_1 a_2 \ldots a_{n-1}(a_n + a_{n+1} - \mu)$$

Multiply both sides by μ to obtain:

$$\mu^{n+1} \geq a_1 a_2 \ldots a_{n-1} a'_n \mu \tag{2.8}$$

And, finally, $(a_n + a_{n+1} - \mu)\mu - a_n a_{n+1} = (a_{n+1} - \mu)(\mu - a_n) \geq 0$, implying that $a'_n \mu \geq a_n a_{n+1}$. Substitute this inequality on (2.8) to show the result for sequences of size $n + 1$. Consequently, by induction, the lemma holds. □

Lemma 2.2.10. *Let $c, d > 0$ and $\{x_i\}_{i=1}^n$ be a real sequence such that $\sum_{i=1}^n x_i = c$ then*

$$\prod_{i=1}^{n}(1 - x_i + x_i d) \leq \left(1 - \frac{c}{n} + \frac{cd}{n}\right)^n$$

Proof. Apply Lemma 2.2.9 to a sequence $\{a_i\}_{i=1}^n$ where, for all $i \in \{1, 2, \ldots, n\}$, $a_i = 1 - x_i + dx_i$ and the result follows. □

Lemma 2.2.11. *For all $x \in \mathbb{R}$, $1 + x \leq e^x$.*

Proof. Let $f : \mathbb{R} \to \mathbb{R}$ be a function such that, for all $x \in \mathbb{R}$, $f(x) = e^x - x - 1$. We must show that $f(x) \geq 0$ for all $x \in \mathbb{R}$.

Consider the first derivative of f, $f'(x) = e^x - 1$. It is easy to verify that $x > 0$ implies $f'(x) > 0$ and $x < 0$ implies $f'(x) < 0$, since e^x is a strictly increasing function and $f'(0) = 0$.

Now, we prove the lemma by contradiction. Assume that there is $x \in \mathbb{R}$ such that $f(x) < 0$. We already know that x is different than 0 because $f(0) = 0$. There are two cases:

1. $x > 0$: in that case, by the Mean Value Theorem, there must exist c such that $0 < c < x$ and $f'(c) = \dfrac{f(x) - f(0)}{x - 0} = f(x)/x < 0$ implying that $f'(c) < 0$, but, by the previous observation, we would necessarily have $f'(c) \geq 0$ (a contradiction);

2. $x < 0$: similarly we utilize the Mean Value Theorem to show that there must exist c such that $x < c < 0$ and $f'(c) = \dfrac{f(0) - f(x)}{0 - x} = f(x)/x > 0$, implying that $f'(c) > 0$ which would be also be a contradiction.

Thus, we conclude that $f(x) \geq 0$ for all $x \in \mathbb{R}$ and the lemma holds. □

We now turn to the computation of a lower bound for $F_k(\gamma)$. Consider the case for a client j and a single tree with root $r \in \mathcal{P}_k$. Let $f_k : [0, 1] \to [0, 1]$ be a function that receives the probability of opening the root of the tree and returns the greatest lower bound to the probability of a path from root to leaf being entirely opened. For the moment, also assume that, for all $p \in \mathcal{P}_F$, x_p coincides with the utilization by j which is $y_p \triangleq \sum_{q \in \mathcal{P}_1 : q \sqsupseteq p} x_{jq}$, that is $x_p = y_p$. In this case, the probability of opening a root is equal to the fractional utilization $z \triangleq \sum_{p \in \mathcal{P}_1 : p \sqsupseteq r} x_{jp} = y_r$ of the root by j. We know that, by definition, $f_1(z) = z$ and, for $k \geq 2$:

$$f_k(z) = z \left(\inf_{n \in \mathbb{Z}^+, z_1, z_2, \ldots, z_n \geq 0 : \ \sum_i z_i = z} 1 - \prod_{i=1}^{n} \left(1 - f_{k-1}\left(\frac{z_i}{z} \right) \right) \right)$$

For all $i \in \{1, 2, \ldots, n\}$, we utilize z_i/z as the probability of opening child i because the expression in parenthesis is the probability of opening one path from root to leaf conditioned on the event that the root was opened. Given that v was opened, we scale the probability of opening every descendent u of v by a factor of $1/x_v$, for the following reason: the probability of v and u opening are respectively x_v and x_u as stated in Lemma 2.2.6 and u cannot be opened if v was not opened, thus the event of u being opened is completely contained in the event of v being opened. Notice that, as we descend the levels, this scaling of the probabilities does not produce wrong results because the scale factor also changes. For instance, consider a chain of nodes u_1, u_2 and u_3 where u_1 is the parent of u_2 and u_2 is the parent of u_3. When we "reach" u_1 the probability of opening u_2 becomes x_{u_2}/x_{u_1} and the probability of opening u_3 becomes x_{u_3}/x_{u_1}. Later, when we go down a level and "reach" u_2, the conditional probability of opening u_3 is divided by the conditional probability of opening u_2, thus becoming $(x_{u_3}/x_{u_1})/(x_{u_2}/x_{u_1}) = x_{u_3}/x_{u_2}$ as desired.

Lemma 2.2.12. *If $f_k(z) \geq z(1-c)$ for a non-negative constant c and all $z \in [0,1]$, then $f_{k+1}(z) \geq z(1 - e^{c-1})$ for all $z \in [0,1]$.*

Proof.

$$\begin{aligned} f_{k+1}(z) &= z \left(\inf_{n \in \mathbb{Z}^+, z_1, z_2, \ldots, z_n \geq 0 : \ \sum_i z_i = z} 1 - \prod_{i=1}^{n} \left(1 - f_k\left(\frac{z_i}{z} \right) \right) \right) \\ &\geq z \left(1 - \sup_{n \in \mathbb{Z}^+, z_1, z_2, \ldots, z_n \geq 0 : \ \sum_i z_i = z} \prod_{i=1}^{n} \left(1 - \frac{z_i}{z}(1-c) \right) \right) \end{aligned}$$

For a fixed value $n \in \mathbb{Z}^+$, we apply Lemma 2.2.10 and then, by applying Lemma 2.2.11, we get rid of n obtaining:

$$\begin{aligned} f_{k+1}(z) &\geq z \left(1 - \left(1 - \frac{1}{n} + \frac{c}{n} \right)^n \right) \\ &\geq z \left(1 - e^{c-1} \right) \end{aligned}$$

□

Now, we generalize f_k. Assume that $f_k(z, x)$ returns the greatest lower bound to the probability of a path from root to leaf being entirely opened in the tree, given that z is the fractional utilization by client j and x is the probability of opening the root. In the case $k = 1$, we know, by definition, that $f_1(z, x) = x$ and, for $k \geq 2$:

$$f_k(z, x) = x \left(\inf_{\substack{n \in \mathbb{Z}^+, z_1, z_2, \ldots, z_n \geq 0:\ \sum_i z_i = z \\ x \geq x_1, x_2, \ldots, x_n \geq 0:\ \forall i\ x_i \geq z_i}} 1 - \prod_{i=1}^{n} \left(1 - f_{k-1}\left(\frac{z_i}{x}, \frac{x_i}{x}\right)\right) \right)$$

In this function, the fractional utilization by client j in each node of a subtree is scaled with the fractional opening of the root because it serves as a tight lower bound to the fractional opening of each node. If this value were not scaled properly, it would be possible to reach lower values than any feasible assignment could possibly achieve.

It can be shown by induction that $f_k(z, x) \geq f_k(z)$, utilizing Inequality (2.3) to ensure that the total flow of any client that passes through a node v must not be greater than the fractional opening x_v of v (in our case, the fractional opening is even increased by scaling). Thus, we can utilize Lemma 2.2.12 to obtain a lower bound to $f_k(z, x)$ for all $z, x \in [0, 1]$.

Theorem 2.2.13. *There is a non-decreasing real sequence $\{c_k\}_{k \in \{1,2,3,\ldots\}}$, such that for all $\gamma \in \mathbb{R}^+$, $F_k(\gamma) \geq 1 - e^{(c_k - 1)\gamma} > 0$.*

Proof. We will show that Lemma 2.2.12 with the basis $f_1(z) \geq z = z(1 - 0)$ provides such sequence. This lemma generates inductively a sequence that starts with $c_1 = 0$ and, for every $k \in \{1, 2, 3, \ldots\}$, maps c_{k+1} to $e^{c_k - 1}$. Lemma 2.2.11 shows that this sequence is non-decreasing and, by induction, it can be shown that $c_k < 1$, assuring us that $1 - e^{(c_k - 1)\gamma} > 0$.

Additionally Lemma 2.2.12 guarantees that $f_k(z) \geq z(1 - c_k)$ and, since the rounding of every tree occurs independently once tokens are distributed, we can obtain the lower bound expression for $F_k(\gamma)$ in the following way:

$$\begin{aligned} F_k(\gamma) &\geq 1 - \sup_{n \in \mathbb{Z}^+, x_1, x_2, \ldots, x_n \geq 0: \sum_i x_i = \gamma} \prod_{i=0}^{n} (1 - f_k(x_i)) \\ &\geq 1 - \sup_{n \in \mathbb{Z}^+, x_1, x_2, \ldots, x_n \geq 0: \sum_i x_i = \gamma} \prod_{i=0}^{n} (1 - x_i(1 - c_k)) \\ &\geq 1 - e^{(c_k - 1)\gamma}, \end{aligned}$$

where the last step is a combination of the results in Lemma 2.2.11 and Lemma 2.2.10. ☐

Applying Theorem 2.2.13 to Theorem 2.2.8, we obtain that the algorithm is a $(3 - o(1))$-approximation. This result can be further improved by scaling opening variables by some constant $\gamma \geq 1$ and redistributing fractional connections from clients to paths in a manner analogous to the UFLP case (see [7] for an example of this technique), but we will not cover the details here. The resulting approximation factors for some fixed k are:

k	1	2	3	4	5	6	7	8	9	10
factor (without scaling)	1.74	2.06	2.25	2.38	2.46	2.53	2.58	2.62	2.66	2.68
factor (with scaling)	1.58	1.85	2.02	2.14	2.24	2.31	2.37	2.42	2.46	2.50

Chapter 3

The Dynamic Facility Location Problem

The metric Dynamic Facility Location Problem (DFLP) is a generalization of the UFLP where time is taken into account. At each time period, clients have different demands that must be supplied by facilities previously opened, but the opening cost of facilities change at each moment and connection costs might increase for facilities opened for too long due to, for example, storage costs.

Formally, the problem is composed by a set of facilities $\mathcal{F}$, a set of clients $\mathcal{C}$ and a set $\mathcal{T}$ of discrete time periods numbered from 1 to T. The cost of opening facility i at time period t is denoted by f_i^t, the amount of demand of client j at time t is d_j^t and the cost of connecting client j during time period t to facility i opened at time period s is denoted by c_{ij}^{st}, where necessarily $s \leq t$. Additionally, the metric variation has the following property related to connection costs: given $i_1, i_2 \in \mathcal{F}$, $j_1, j_2 \in \mathcal{C}$ and time periods s, s', t, t' such that $1 \leq s \leq s' \leq t, t' \leq T$, we have that

$$c_{i_1 j_1}^{st} \leq c_{i_1 j_2}^{st'} + c_{i_2 j_2}^{s't'} + c_{i_2 j_1}^{s't},$$

which we will refer as *triangle inequality* in the rest of this chapter.

A solution contains a set $S \subseteq \mathcal{F} \times \mathcal{T}$ of facility-period pairs to be opened and a map $\phi : \mathcal{C} \times \mathcal{T} \to S$ that connects each client-period pair to a facility-period pair, where, whenever $\phi(j,t) = (i,s)$, we must have that $s \leq t$. An optimal solution minimizes the cost, given by the following expression:

$$\sum_{(i,s) \in S} f_i^s + \sum_{\substack{j \in \mathcal{C}, t \in \mathcal{T}: \\ (i,s) = \phi(j,t)}} d_j^t c_{ij}^{st}.$$

When there is only one time period, the DFLP becomes the UFLP and, as a result, all inapproximability results for the UFLP are also valid for this problem (see Section 2.1).

The metric Stochastic Facility Location Problem (SFLP) is a related problem in which there is a set of facilities $\mathcal{F}$, a set of clients $\mathcal{C}$ and a set of scenarios $\mathcal{A}$. Connection costs from clients to facilities are given by a metric function $c : (\mathcal{C} \cup \mathcal{F})^2 \to \mathbb{R}^+$ and the considered time frame is subdivided into two discrete time periods or *stages*. In the first stage, any subset of facilities can be opened and each facility i costs f_i to be opened. In the second stage, exactly one of many scenarios occur according to a known probabilty distribution. Each scenario $A \in \mathcal{A}$ defines a subset of participating clients $\mathcal{C}_A \subseteq \mathcal{C}$ and, for each facility $i \in \mathcal{F}$, an opening cost f_i^A that replaces the opening cost from the first stage.

A solution to the SFLP consists of a set $S \subseteq \mathcal{F}$ of facilities to be opened in the first stage and, for each scenario $A \in \mathcal{A}$, a set of facilities $S_A \subseteq \mathcal{F}$ to be opened in case scenario A occurs and a map $\phi_A : \mathcal{C}_A \to (S_A \cup S)$ connecting every participating client to a facility available at that scenario. The cost of a solution is given by the following expression:

$$\sum_{i \in S} f_i + \sum_{A \in \mathcal{A}} p_A \left(\sum_{j \in \mathcal{C}_A} c_{\phi_A(j)j} + \sum_{i \in S_A} f_i^A \right),$$

which is the expect sum of connection and opening costs. In Section 3.1, we discuss how to reduce the SFLP to the DFLP in a way that allows algorithms for the DFLP to be used as approximation algorithms to the SFLP with the same approximation factor.

The first approximation algorithm for the SFLP is an LP-rounding algorithm by Ravi and Sinha [25] that obtains an approximation factor* of 5, combining the filtering technique by Lin and Vitter [20, 21] with a stage-constrained clustering of client-scenario pairs.

Shmoys and Swamy [26] devised a PTAS for approximating with high probability a certain class of linear relaxations for stochastic problems. In their algorithm, which is an adaptation of the ellipsoid method, they sample the scenario distribution and obtain the cost structure of the sampled scenarios by making requests to an oracle. They also described, for some problems, how to round the solution to linear relaxations obtained by their PTAS. In particular, for the FLP, they obtained a $(3.378 + \epsilon)$-approximation by using a FLP approximation algorithm that does not require knowledge of the clients' demands in each scenario. As a consequence, their SFLP algorithm does not require the exact scenario distribution or the costs in each scenario.

The first (and currently best) approximation algorithm for the DFLP is a 1.86-approximation due to Ye and Zhang [38], which we will review in Section 3.2. Their algorithm can also be used to approximate a solution for the SFLP with the same approximation factor, as we will see in Section 3.1.

3.1 Reduction from the SFLP to the DFLP

In this section, we will review a reduction presented by Ye and Zhang [38] from the SFLP to the DFLP that allows approximation algorithms for the DFLP to be used as approximation algorithms for the SFLP.

Formally, given an instance $\mathcal{I}$ to the SFLP, we will create an instance $\mathcal{I}'$ to the DFLP such that any feasible solution to $\mathcal{I}'$ can be mapped one-to-one to a solution to $\mathcal{I}$ of the

*The paper states that the algorithm is a 8-approximation, but for parameters $\alpha = 2/5$ and $\beta = 1/2$, it has an approximation factor of 5.

same cost. First, we determine that $\mathcal{I}'$ has two time periods, i.e. $T = 2$. Define its set of clients to be $\mathcal{C}' \triangleq \{(j, A) : A \in \mathcal{A}, j \in \mathcal{C}^A\}$, where $\mathcal{A}$ is the set of scenarios of $\mathcal{I}$ and $\mathcal{C}^A$ is the set of clients active during scenario A. Let all clients $(j, A) \in \mathcal{C}'$ have demand 0 during time period 1 and demand p_A during time period 2, where p_A is the probability of scenario A occurring in the second stage.

For every $A \in \mathcal{A}$, let $\mathcal{F}^A = \{(i, A) : i \in \mathcal{F}\}$ be a set of facilities related to scenario A and also let $\mathcal{F}^I = \{(i, I) : i \in \mathcal{F}\}$ be a set of facilities related to the first stage, where $\mathcal{F}$ is the set of facilities from $\mathcal{I}$. Denote the set of facilities in $\mathcal{I}'$ as $\mathcal{F}' \triangleq \bigcup_{A \in \mathcal{A} \cup \{I\}} \mathcal{F}^A$. For every facility $(i, I) \in \mathcal{F}^I$, let its opening cost in time period 1 be equal to the corresponding first-stage opening cost of i in $\mathcal{I}$ and let its opening cost in time period 2 be ∞. Analogously, for every scenario $A \in \mathcal{A}$ and, for every facility $(i, A) \in \mathcal{F}^A$, let its opening cost during time period 2 be $p_A f_i^A$, where p_A is the probability of occurrence of scenario A and f_i^A is the opening cost of facility i during scenario A, and let its opening cost during time period 1 to be ∞.

Given a client $(j, A) \in \mathcal{C}'$ and a facility $(i, B) \in \mathcal{F}'$, define the connection cost to be equal to the connection cost from i to j in $\mathcal{I}$ whenever either 1) $A = B$ and (i, B) was opened in the second time period; or 2) $B = I$ and (i, B) was opened in the first time period. Otherwise simply consider it to be ∞. We now need to ensure that the distances as defined meet the metric definition as previously stated.

Assume we have two clients (j_1, A_1) and (j_2, A_2) and two facilities (i_1, B_1) and (i_2, B_2). As previously stated, the following property holds[†]:

$$\begin{aligned} c((i_1, B_1, s_1), (j_1, A_1, t_1)) \leq\ & c((i_2, B_2, s_2), (j_1, A_1, t_1)) \\ & + c((i_2, B_2, s_2), (j_2, A_2, t_2)) \\ & + c((i_1, B_1, s_1), (j_2, A_2, t_2)), \end{aligned} \tag{3.1}$$

where s_1, s_2, t_1 and t_2 are the time periods where, respectively, facility (i_1, B_1), facility (i_2, B_2), client (j_1, A_1) and client (j_2, A_2) were opened and $1 \leq s_1 \leq s_2 \leq t_1, t_2 \leq 2$.

We will separate the proof in two cases. In the first case, we assume that both facilities were opened in the same time period. We only need to consider the case in which there is at least one pair client-facility whose distance is infinite, otherwise the triangle inequality would follow directly from the triangle inequality of $\mathcal{I}$. Without loss of generality, assume that this is the case for client (j_1, A_1) and facility (i_1, B_1). If both facilities belong to time period 1, then we must have $B_1 \neq I$ so (j_1, A_1) has infinite distance from (i_1, B_1), which would imply that client (j_2, A_2) would also have an infinite distance to (i_1, B_1), ensuring that the triangle inequality would be satisfied. In case both facilities belong to time period 2, we have that $B_1 \neq A_1$, implying that at least one of these statements must be also true: $A_1 \neq B_2$, $A_2 \neq B_1$ or $A_2 \neq B_2$. By construction, that would mean that at least one term in the right-hand side of inequality (3.1) would also be infinity, and the triangle inequality would be satisfied as desired.

In the second case, we assume facilities were opened in different time periods. Again, we only need to consider a case where at least one distance is infinite. We will also assume,

[†]For clarity, we are representing $c^{st}_{(i,B)(j,A)}$ as $c((i, B, s), (j, A, t))$.

without loss of generality, that facility (i_1, B_1) was opened in time period 1 and facility (i_2, B_2) was opened in time period two. Due to the constraints of the triangle inequality in respect to time periods, we only need to check the case where (i_1, B_1) has an infinite distance to some client, which would imply $B_1 \neq I$ by construction, ensuring that both clients have infinite distance to that facility and that the triangle inequality follows.

Thus, by case analysis, we conclude that the resulting instance $\mathcal{I}'$ is a valid DFLP instance. We claim that this reduction is a cost-preserving one-to-one mapping from feasible solutions to $\mathcal{I}'$ to feasible solutions to $\mathcal{I}$: opening a facility (i, X) in $\mathcal{I}'$ corresponds to opening a facility i either in the second stage, when scenario X occurs, or in the first stage; also, in $\mathcal{I}'$, connecting a client $(j, A) \in \mathcal{C}'$ to a facility $(i, X) \in \mathcal{F}'$ corresponds to connecting, in $\mathcal{I}$, a client j to facility i during scenario A. Therefore, an approximation for the DFLP can be used to obtain approximate solutions for the SFLP.

3.2 A 1.86-approximation for the DFLP

In this section, we review a 1.86-approximation by Ye and Zhang [38] for the DFLP. Their algorithm uses an adaptation of the classical primal-dual strategy devised by Jain and Vazirani [17] to obtain an intermediate solution that is later improved by greedy augmentation. Similarly to the algorithm presented in Chapter 2, this algorithm is also used as a basis for an algorithm that approximates a stochastic variant of the k-LFLP that will be presented in Chapter 4. In this section, the notation $[a, b]$ represents the set of integers from a to b inclusive, that is $\{a, a+1, a+2, \ldots, b-1, b\}$.

3.2.1 Algorithm description

The problem has the following ILP:

$$\text{minimize} \sum_{t=1}^{T} \sum_{i \in \mathcal{F}} \sum_{s=1}^{T} \sum_{j \in \mathcal{C}} d_j^t c_{ij}^{st} x_{ij}^{st} + \sum_{s=1}^{T} \sum_{i \in \mathcal{F}} f_i^s y_i^s \tag{3.2}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{F}} \sum_{s=1}^{t} x_{ij}^{st} = 1 \qquad \forall j \in \mathcal{C}, t \in [1, T] \tag{3.3}$$

$$x_{ij}^{st} \leq y_i^s \qquad \forall i \in \mathcal{F}, j \in \mathcal{C}, s, t \in [1, T] \tag{3.4}$$

$$x_{ij}^{st}, y_i^s \in \{0, 1\} \qquad \forall i \in \mathcal{F}, j \in \mathcal{C}, s, t \in [1, T] \tag{3.5}$$

where, for each $i \in \mathcal{F}, j \in \mathcal{C}$ and integers $s, t \in [1, T]$, x_{ij}^{st} is equal to 1 if and only if client j is supplied at time t by facility i opened at time s. For every $i \in \mathcal{F}$ and integer $s \in [1, T]$, y_i^s is 1 if and only if facility i was opened in time t. inequalities (3.3) state that, given a client $j \in \mathcal{C}$ and a time $t \in [1, T]$, j must be supplied by a facility opened before or exactly at time t. inequalities (3.4) ensure that, if a client $j \in \mathcal{C}$ is served at time $t \in [1, T]$ by facility $i \in \mathcal{F}$ at time $s \in [1, t]$, facility i must be opened at time s. The relaxed LP for this problem replaces the integrality constraints (3.5) with $x_{ij}^{st}, y_i^s \geq 0$ for all $i \in \mathcal{F}, j \in \mathcal{C}, s, t \in [1, T]$.

The dual LP is:

$$\begin{aligned}
\text{maximize} \quad & \sum_{t=1}^{T}\sum_{j\in\mathcal{C}} \alpha_j^t \\
\text{s.t.} \quad & \alpha_j^t - d_j^t c_{ij}^{st} \leq \beta_{ij}^{st} \;\; \forall i \in \mathcal{F}, j \in \mathcal{C}, s,t \in [1,T] \\
& \sum_{t=1}^{T}\sum_{j\in\mathcal{C}} \beta_{ij}^{st} \leq f_i^s \qquad \forall i \in \mathcal{F}, s \in [1,T] \\
& \alpha_j^t, \beta_{ij}^{st} \geq 0 \quad \forall i \in \mathcal{F}, j \in \mathcal{C}, s,t \in [1,T]
\end{aligned}$$

However, their primal-dual algorithm uses the following equivalent formulation to make proofs for the upcoming lemmas and theorems clearer:

$$\text{maximize} \quad \sum_{t=1}^{T}\sum_{j\in\mathcal{C}} d_j^t \alpha_j^t \tag{3.6}$$

$$\text{s.t.} \quad \alpha_j^t - c_{ij}^{st} \leq \beta_{ij}^{st} \;\; \forall i \in \mathcal{F}, j \in \mathcal{C}, s,t \in [1,T] \tag{3.7}$$

$$\sum_{t=1}^{T}\sum_{j\in\mathcal{C}} d_j^t \beta_{ij}^{st} \leq f_i^s \qquad \forall i \in \mathcal{F}, s \in [1,T] \tag{3.8}$$

$$\alpha_j^t, \beta_{ij}^{st} \geq 0 \quad \forall i \in \mathcal{F}, j \in \mathcal{C}, s,t \in [1,T] \tag{3.9}$$

which is the result of a simple scaling of variables from the former program, thus allowing us to utilize both Strong Duality (see Theorem 1.1.2) and Complementary slackness conditions (see Theorem 1.1.3).

Jain and Vazirani [17] presented an interpretation for the dual variables which can be adapted to the dual LP for the DFLP. Suppose that there was an integral solution to the original ILP that is also an optimal solution to the relaxed primal LP. In that case, we can say that, for each $(j,t) \in \mathcal{C} \times [1,T]$, α_j^t is the "budget" of the client-period pair (j,t) and, for all $(j,t) \in \mathcal{C} \times [1,T]$, β_{ij}^{st} is the amount of its budget that (j,t) uses to "pay" the opening of facility i at time period s, where i is the facility, opened at period s, that satisfies j's demand during time period t. To verify that this interpretation is compatible with the linear programs, simply apply complementary slackness: if $x_{ij}^{st} = 1$, then $\alpha_j^t = \beta_{ij}^{st} + c_{ij}^{st}$, which shows that the budget is utilized to "pay" the cost of connecting j to i through time periods from s to t and to "pay" a contribution to open i at time period s; also, if $y_i^s = 1$, $\sum_{t=1}^{T}\sum_{j\in\mathcal{C}} d_j^t \beta_{ij}^{st} = f_i^s$, which means that a facility i opened at time period s must have enough contributions from pairs $(j,t) \in \mathcal{C} \times [1,T]$ (each scaled by their respective demands) to be able to pay its opening cost.

Now, we describe the algorithm. Given a parameter $\delta > 0$, it first scales the opening costs of facilities in all time periods by δ and then proceed to three phases. In the first phase, it executes a variation of the dual-ascent strategy, obtaining a feasible dual solution. In the second, a feasible primal solution is derived from the feasible dual solution from the first phase. And, finally, in the third phase, the cost scaling is removed and the solution obtained so far is iteratively improved by greedily opening new facilities, in a process known as *greedy augmentation* that we will describe in Section 3.2.4.

3.2.2 Phase 1: obtaining a feasible dual solution

The algorithm is based on a notion of time unrelated to the time periods from the problem description. At $\theta = 0$, all dual variables are set to 0, facility-period pairs $(i, s) \in \mathcal{F} \times [1, T]$ are *closed* and client-period pairs $(j, t) \in \mathcal{C} \times [1, T]$ are *unfrozen*. Starting at $\theta = 0$, each variable α_j^t, for $j \in \mathcal{C}$ and integral $t \in [1, T]$, is increased uniformly as time passes until the client-period pair (j, t) becomes *frozen*. Whenever a pair $(j, t) \in \mathcal{C} \times [1, T]$ becomes frozen, α_j^t stops increasing and, for all $(i, s) \in \mathcal{F} \times [1, T]$, if β_{ij}^{st} was increasing due to an event (see below), it stops. To ensure feasibility of the obtained solution, there are three events that must be handled appropriately:

1. For some closed pair $(i, s) \in \mathcal{F} \times [1, T]$, the associated inequality (3.8) is met with equality, i.e. $\sum_{t=1}^{T} \sum_{j \in \mathcal{C}} d_j^t \beta_{ij}^{st} = f_i^s$. In that case, we say that (i, s) becomes *tentatively open* (it stops being closed) and, for all unfrozen pairs (j, t) such that $\beta_{ij}^{st} > 0$, (i, s) becomes the *connecting witness* of (j, t) and (j, t) becomes frozen.

2. For some pairs $(j, t) \in \mathcal{C} \times [1, T]$ and $(i, s) \in \mathcal{F} \times [1, T]$ where (i, s) is closed, α_j^t becomes equal to c_{ij}^{st}. In this point, we say that (j, t) is *tight* with (i, s) and, if (j, t) is unfrozen, β_{ij}^{st} begins to increase at the same rate as α_j^t so the associated inequality (3.7) remains satisfied.

3. For some pairs $(j, t) \in \mathcal{C} \times [1, T]$ and $(i, s) \in \mathcal{F} \times [1, T]$ where (i, s) is tentatively opened, α_j^t becomes equal to c_{ij}^{st}. In this case, we say that (j, t) becomes *tight* with (i, s), (i, s) becomes the connecting witness of (j, t) and (j, t) becomes frozen.

We simulate this process until all pairs $(j, t) \in \mathcal{C} \times [1, T]$ become frozen, when the resulting valuation of the dual variables becomes our feasible dual solution.

3.2.3 Phase 2: obtaining a feasible primal solution

First, we create a dependency graph $G(V, E)$ where the set of vertices V is composed by all the facility-period pairs $(i, s) \in \mathcal{F} \times [1, T]$. There is an edge joining two facility-period pairs (i, s) and (i', s') if and only if there is a client-period pair $(j, t) \in \mathcal{C}$ such that $\beta_{ij}^{st} > 0$ and $\beta_{i'j}^{s't} > 0$. Whenever $u, v \in V$ are connected by an edge, u and v are said to be *dependent*.

Next, we obtain a maximal independent set I on G such that, for each pair $(i, s) \in \mathcal{F} \times [1, T]$, either $(i, s) \in I$ or there is a pair $(i', s') \in I$ dependent on (i, s) such that $s' \leq s$. One way of ensuring this property is to iteratively build an independent set starting with the empty set and considering all pairs in $\mathcal{F} \times [1, T]$ in non-decreasing order of time period, adding a pair under consideration to the set only if the resulting set is still independent.

A description of the algorithm follows:

$I \leftarrow \emptyset$
$V' \leftarrow \mathcal{F} \times [1, T]$
sort V' by time period
for $v \in V'$ **do**
  if $I \cup \{v\}$ *is independent on* G **then**
    $I \leftarrow I \cup \{v\}$
  end
end

Finally, for all pairs $(i, s) \in I$, open facility at time period s. Connect every client-period pair $(j, t) \in \mathcal{C} \times [1, T]$ (j, t) to a facility-period pair $(i, s) \in I$ that minimizes c_{ij}^{st}.

3.2.4 Phase 3: greedy augmentation

Finally, there is still the possibility that opening certain facilities at certain time periods might decrease the solution's cost, because the cost of opening any of them is compensated by a reduction in connection costs as some clients could be re-routed to them. In this phase, we remove cost scaling applied at the beginning, considering the original opening costs.

Let Sol be any feasible integral solution to an instance of the DFLP. For every pair $(i, s) \in \mathcal{F} \times [1, T]$, let $\text{gain}(\text{Sol}, i, s) \triangleq C - C' - f_i^s$, where C is the total connection cost in Sol and C' is the total connection cost after Sol is modified to include facility i opened at time period s and, at each time period, clients are connected to their "closest" available facilities. Also, define $\text{ratio}(\text{Sol}, i, s) \triangleq \text{gain}(\text{Sol}, i, s)/f_i^s$.

While there are facility-period pairs that, if opened, can reduce the overall solution cost, i.e. while there is a facility-period pair (i, s) such that $\text{gain}(\text{Sol}, i, s) > 0$, the greedy augmentation procedure iteratively opens one such pair that maximizes the ratio between gain and opening cost. Or, more precisely, the procedure works as described below.

Sol $\leftarrow$ solution from Phase 2
while $\exists (i, s) \in \mathcal{F} \times [1, T] : \text{gain}(\text{Sol}, i, s) > 0$ **do**
  $(i, s) \leftarrow \arg\max_{(i,s) \in \mathcal{F} \times [1,T]} \text{ratio}(\text{Sol}, i, s)$
  Open i at time period s in Sol re-routing clients to closest available facilities.
end
return Sol

3.2.5 Analysis

We will analyze the solution obtained after Phase 2, assuming, without loss of generality, that we did not apply cost scaling.

Theorem 3.2.1. *Let* Sol *be the solution obtained by the algorithm after Phase 2 for an instance* $\mathcal{I}$*,* F_{Sol} *be the total opening cost of* Sol*,* C_{Sol} *the total connection cost of* Sol *and*

OPT *be the optimal value of the DFLP for $\mathcal{I}$. Then,*

$$3F_{\text{Sol}} + C_{\text{Sol}} \leq 3\text{OPT}$$

Proof. Let I be the maximal independent set obtained at the end of Phase 2. Define $D' \triangleq \{(j,t) : j \in \mathcal{C}, t \in [1,T], \exists (i,s) \in I, \text{ s.t. } \beta_{ij}^{st} > 0\}$. This is the set of all client-period pairs that have a non-zero contribution to facilities opened in Sol. Also, for every pair $(j,t) \in \mathcal{C} \times [1,T]$, if j is served in Sol during time period t by a facility i opened in time period s, then define $\delta_{jt} \triangleq i$ and $\gamma_{jt} \triangleq s$.

We know that, for each $(i,s) \in I$,

$$f_i^s = \sum_{t=1}^{T} \sum_{j \in \mathcal{C}} d_j^t \beta_{ij}^{st}$$

Thus, we obtain the total opening cost of Sol as a summation over D':

$$\begin{aligned} F_{\text{Sol}} = \sum_{(i,s) \in I} f_i^s &= \sum_{(i,s) \in I} \sum_{(j,t) \in \mathcal{C} \times [1,T]} d_j^t \beta_{ij}^{st} \\ &= \sum_{(i,s) \in I} \sum_{(j,t) \in D'} d_j^t \beta_{ij}^{st} \\ &= \sum_{(j,t) \in D'} \sum_{(i,s) \in I} d_j^t \beta_{ij}^{st} \\ &= \sum_{(j,t) \in D'} d_j^t \beta_{\delta_{jt} j}^{\gamma_{jt} t}, \end{aligned}$$

where the last inequality follows from I being independent.

Now, we consider the total connection cost, given by the expression

$$C_{\text{Sol}} = \sum_{(j,t) \in \mathcal{C} \times [1,T]} d_j^t c_{\delta_{jt} j}^{\gamma_{jt} t}$$

We only need to upper bound the connection costs of all pairs $(j,t) \in \mathcal{C} \times [1,T]$. Consider pairs $(j,t) \in \mathcal{C} \times [1,T]$, such that $(j,t) \in D'$. In that case, there is a pair $(i,s) \in I$ such that $\beta_{ij}^{st} > 0$. In that case, we know that $(i,s) = (\delta_{jt}, \gamma_{jt})$ and, also, $\alpha_j^t = c_{\delta_{jt} j}^{\gamma_{jt} t} + \beta_{\delta_{jt} j}^{\gamma_{jt} t}$. As a consequence, $c_{\delta_{jt} j}^{\gamma_{jt} t} = \alpha_j^t - \beta_{\delta_{jt} j}^{\gamma_{jt} t}$.

For pairs $(j,t) \in \mathcal{C} \times [1,T]$ such that $(j,t) \notin D'$, we know that there must be a facility-period pair $(i,s) \in I$ that is dependent on the connecting witness (i',s') of (j,t) such that $s \leq s'$ and can be connected to (j,t). Also, due to the definition of dependency, there must exist $(j',t') \in \mathcal{C} \times [1,T]$ such that $\beta_{i'j'}^{s't'} > 0$ and $\beta_{ij'}^{st'} > 0$. Note that (j',t') must be connected to (i,s). In this case, we can apply triangle inequality $c_{\delta_{jt} j}^{\gamma_{jt} t} \leq c_{ij}^{st} \leq c_{i'j}^{s't} + c_{i'j'}^{s't'} + c_{ij'}^{st'}$. We know that (j',t') is tight with (i,s) and (i',s') and (j,t) is tight with (i',s'), this implies that $c_{i'j}^{s't} \leq \alpha_j^t$, $c_{i'j'}^{s't'} \leq \alpha_{j'}^{t'}$ and $c_{ij'}^{st'} \leq \alpha_{j'}^{t'}$.

Let $\theta_{i'}^{s'}$ denote the moment when (i',s') became tentatively open. Since (i',s') is the connecting witness of (j,t), either (j,t) became frozen exactly when (i',s') became tentatively open or later, implying that $\theta_{i'}^{s'} \leq \alpha_j^t$. Also, if (j',t') did not became frozen by the time $\theta_{i'}^{s'}$, it would become frozen at that moment since $\beta_{i'j'}^{s't'} > 0$, making $\alpha_{j'}^{t'} \leq \theta_{i'}^{s'}$. Thus $\alpha_{j'}^{t'} \leq \alpha_j^t$ and $c_{\delta_{jt}j}^{\gamma_{jt}t} \leq \alpha_j^t + 2\alpha_{j'}^{t'} \leq 3\alpha_j^t$.

Let $D = \mathcal{C} \times [1, T]$. We can now prove the theorem:

$$
\begin{aligned}
3F_{\text{Sol}} + C_{\text{Sol}} &\leq 3 \sum_{(j,t)\in D'} d_j^t \beta_{\delta_{jt}j}^{\gamma_{jt}t} + \sum_{(j,t)\in \mathcal{C}\times[1,T]} d_j^t c_{\delta_{jt}j}^{\gamma_{jt}t} \\
&\leq 3 \sum_{(j,t)\in D'} d_j^t(\alpha_j^t - \beta_{\delta_{jt}j}^{\gamma_{jt}t} + \beta_{\delta_{jt}j}^{\gamma_{jt}t}) + \sum_{(j,t)\in D\setminus D'} 3d_j^t\alpha_j^t \\
&= 3 \sum_{(j,t)\in D} d_j^t\alpha_j^t \leq 3\text{OPT} \qquad \square
\end{aligned}
$$

Next, we show that, after greedy augmentation, the algorithm obtains a 1.86-approximation.

First, we simplify the problem, considering an instance of the UFLP that is not necessarily metric. For that instance, let $\mathcal{F}'$ be the set of facilities, $\mathcal{C}'$ be the set of the clients, $c' : \mathcal{F}' \times \mathcal{C}' \to \mathbb{R}$ be the connection cost function and $f' : \mathcal{F}' \to \mathbb{R}$ be the opening cost function. To transform the DFLP instance into an UFLP instance, simply map every facility-period pair in the DFLP instance to a facility in the UFLP instance, that is $\mathcal{F}' = \mathcal{F} \times [1, T]$, and map every client-period pair in the DFLP instance to a client in the UFLP. The connection cost function is defined in the following manner, given $(i, s) \in \mathcal{F}'$ and $(j, t) \in \mathcal{C}'$, $c'((i, s), (j, t)) \triangleq d_j^t c_{ij}^{st}$.

The classic ILP formulation for the UFLP is:

$$
\begin{aligned}
\text{minimize} \sum_{i\in\mathcal{F}'} f_i' y_i + \sum_{i\in\mathcal{F}', j\in\mathcal{C}'} c_{ij}' x_{ij} & & (3.10) \\
\sum_{i\in\mathcal{F}'} x_{ij} = 1 & \qquad \forall j \in \mathcal{C}' & (3.11) \\
x_{ij} \leq y_i & \qquad \forall i \in \mathcal{F}', j \in \mathcal{C}' & (3.12) \\
x_{ij}, y_i \in \{0, 1\} & \qquad \forall i \in \mathcal{F}', j \in \mathcal{C}' & (3.13)
\end{aligned}
$$

where, for each $i \in \mathcal{F}'$ and $j \in \mathcal{C}'$, x_{ij} is a binary variable that determines whether client j is connected to facility i (a value of 1 indicates that there is a connection) and, for each $i \in \mathcal{F}'$, y_i is 1 if and only if facility i is opened. Inequalities (3.11) indicate that all clients need to be connected to exactly one facility, and inequalities (3.12) indicate that whenever a client is connected to a facility, that facility must be opened.

Now, consider the relaxed LP of the above ILP produced by replacing inequalities (3.13) by $x_{ij}, y_i \geq 1, \forall i \in \mathcal{F}', j \in \mathcal{C}'$. Then, we can show the following lemma:

Lemma 3.2.2 (Lemma 2.6 in [10]). *Given any feasible fractional solution* Sol *for the relaxed LP for an instance $\mathcal{I}$ of the UFLP, such that $F_{\text{Sol}} \triangleq \sum_{i\in\mathcal{F}'} f_i' y_i$ is the total opening cost and $C_{\text{Sol}} \triangleq \sum_{i\in\mathcal{F}', j\in\mathcal{C}'} c_{ij}' x_{ij}$ is the total connection cost, and a feasible integral solution* Sol$'$ *to $\mathcal{I}$ with total connection cost C, then*

$$
\sum_{i\in\mathcal{F}'} y_i \text{gain}(i) \geq C - (F_{\text{Sol}} + C_{\text{Sol}})
$$

where $\text{gain}(i)$ *refers to* Sol' *(see Section 3.2.4 for a definition).*

Proof. We may assume that, for every $i \in \mathcal{F}'$ and $j \in \mathcal{C}'$, either $x_{ij} = y_i$ or $x_{ij} = 0$, because, in case that there is a pair i, j such that $y_i > x_{ij}$, it is possible replace facility i by two copies i_1 and i_2 setting their associated variables in the following manner: $y_{i_1} = x_{ij}$ and $y_{i_2} = y_i - x_{ij}$ and, for all clients j', $x_{i_1 j'} = \min(x_{ij'}, x_{ij})$ and $x_{i_2 j'} = x_{ij'} - x_{i_1 j'}$. This transformation generates a feasible fractional solution with the same fractional opening and connection costs and, thus, can be used in place of the original fractional solution for this theorem.

Let $\sigma : \mathcal{C}' \to \mathcal{F}'$ map every client to its associated facility in the feasible integral solution Sol' and define, for every $i \in \mathcal{F}'$, $D_{\text{Sol}}(i) \triangleq \{j : x_{ij} > 0\}$. Now, suppose that we reassign all clients in $D_{\text{Sol}}(i)$ to i. In that case, for each $i \in \mathcal{F}'$, we define $\text{gain}'(i) = -f_i + \sum_{j \in D_{\text{Sol}}(i)} (c'_{\sigma(j)j} - c'_{ij})$ as the net cost of opening a facility i. Note that $\text{gain}(i) \geq \text{gain}'(i)$, since greedy augmentation will essentially re-route a subset of clients to facility i in order to maximize the net cost and it is always an option to re-route exactly the clients of $D_{\text{Sol}}(i)$.

Finally, we can show the result:

$$\begin{aligned}
\sum_{i \in \mathcal{F}'} y_i \text{gain}(i) &\geq \sum_{i \in \mathcal{F}'} y_i \text{gain}'(i) \\
&= \sum_{i \in \mathcal{F}'} y_i \left(-f'_i + \sum_{j \in D_{\text{Sol}}(i)} (c'_{\sigma(j)j} - c'_{ij}) \right) \\
&= -\sum_{i \in \mathcal{F}'} y_i f'_i + \sum_{i \in \mathcal{F}'} \sum_{j \in D_{\text{Sol}}(i)} y_i (c'_{\sigma(j)j} - c'_{ij}) \\
&= -F_{\text{Sol}} + \sum_{i \in \mathcal{F}'} \sum_{j \in D_{\text{Sol}}(i)} x_{ij} (c'_{\sigma(j)j} - c'_{ij}) \\
&= -F_{\text{Sol}} - C_{\text{Sol}} + \sum_{j \in \mathcal{C}'} c'_{\sigma(j)j} \sum_{i \in \mathcal{F}'} x_{ij} \\
&= C - (F_{\text{Sol}} + C_{\text{Sol}})
\end{aligned}$$ □

The following intermediate result will help us show the next lemma:

Lemma 3.2.3. *Given two real sequences* $(a_i)_{i=1}^N$ *and* $(b_i)_{i=1}^N$ *such that, for every* $i \in \{1, 2, \ldots N\}$, $b_i > 0$,

$$\max_{i=1..N} \frac{a_i}{b_i} \geq \frac{\sum_{i=1}^N a_i}{\sum_{i=1}^N b_i}$$

Proof. Let $\ell \triangleq \arg\max_{i=1..N} a_i / b_i$. We know that, for all $i \in \{1, 2, \ldots, N\}$, $a_\ell b_i \geq a_i b_\ell$. Sum all these inequalities to obtain $a_\ell \sum_{i=1}^N b_i \geq b_\ell \sum_{i=1}^N a_i$ and finally multiply this inequality by $1/(b_\ell \sum_{i=1}^N b_i)$ to reach the desired result. □

Lemma 3.2.4 (Lemma 4.3 in [10])**.** *Let* Sol *be any feasible fractional solution for an instance* $\mathcal{I}$ *of the UFLP with total connection cost* C_{Sol} *and total fractional opening* F_{Sol}*, also let* Sol' *be any feasible integral solution to* $\mathcal{I}$ *with total opening cost* F *and total* *connection cost* C. *If we apply greedy augmentation to* Sol', *the final solution will have cost at most*

$$F + F_{\mathrm{Sol}} \max\left\{0, \ln\left(\frac{C - C_{\mathrm{Sol}}}{F_{\mathrm{Sol}}}\right)\right\} + F_{\mathrm{Sol}} + C_{\mathrm{Sol}}$$

Proof. In case $C \leq F_{\mathrm{Sol}} + C_{\mathrm{Sol}}$, the lemma holds trivially. So, we assume $C > F_{\mathrm{Sol}} + C_{\mathrm{Sol}}$.

Define Sol_k the solution at the k-th iteration of the greedy augmentation procedure, and let C_k and F_k be respectively the total connection and opening costs of Sol_k. Define $\mathrm{Sol}_0 \triangleq \mathrm{Sol}'$.

We will show that, at any iteration k, there must exist a facility $q \in \mathcal{F}'$ such that $\mathrm{ratio}(q) \geq \frac{C_i - C_{\mathrm{Sol}} - F_{\mathrm{Sol}}}{F_{\mathrm{Sol}}}$. Let $(y_i)_{i \in \mathcal{F}}$ be the fractional opening of facilities in Sol, define $\mathcal{F}^* = \{i : i \in \mathcal{F}, y_i > 0\}$ and consider sequences $(a_i)_{i \in \mathcal{F}^*} \triangleq (y_i \mathrm{gain}(i))_{i \in \mathcal{F}^*}$ and $(b_i)_{i \in \mathcal{F}^*} \triangleq (y_i f_i)_{i \in \mathcal{F}^*}$ (assume all facilities have non-zero opening cost, since facilities with zero opening costs can be opened at will), applying Lemma 3.2.3, we know that:

$$\max(\{\mathrm{ratio}(i) : i \in \mathcal{F}^*\}) = \max\left(\left\{\frac{y_i \mathrm{gain}(i)}{y_i f_i'} : i \in \mathcal{F}^*\right\}\right)$$
$$\geq \frac{\sum_{i \in \mathcal{F}^*} y_i \mathrm{gain}(i)}{\sum_{i \in \mathcal{F}^*} y_i f_i'} = \frac{\sum_{i \in \mathcal{F}^*} y_i \mathrm{gain}(i)}{F_{\mathrm{Sol}}}$$

Applying Lemma 3.2.2, we obtain that the facility in $\mathcal{F}^*$ with greatest ratio, can be chosen as q.

We can show that the increase of total opening cost from iteration k to iteration $k+1$ is at most $F_{\mathrm{Sol}} \frac{C_k - C_{k+1}}{C_k - C_{\mathrm{Sol}}}$ by using the fact that that greedy augmentation opens a facility with maximum ratio, implying that

$$\begin{aligned}
& \frac{C_k - C_{k+1} - (F_{k+1} - F_k)}{F_{k+1} - F_k} = \mathrm{ratio}(q) \geq \frac{C_k - C_{\mathrm{Sol}} - F_{\mathrm{Sol}}}{F_{\mathrm{Sol}}} \\
\Longrightarrow \quad & \frac{C_k - C_{k+1}}{F_{k+1} - F_k} \geq \frac{C_k - C_{\mathrm{Sol}}}{F_{\mathrm{Sol}}} \\
\Longleftrightarrow \quad & F_{\mathrm{Sol}} \frac{C_k - C_{k+1}}{C_k - C_{\mathrm{Sol}}} \geq F_{k+1} - F_k
\end{aligned}$$

We now argue that the there is an iteration $m \geq 1$ where $C_m \leq F_{\mathrm{Sol}} + C_{\mathrm{Sol}}$. By contradiction, assume that, in the last iteration i, $C_i > F_{\mathrm{Sol}} + C_{\mathrm{Sol}}$. We know that there is facility q with $\mathrm{ratio}(q) \geq (C_i - (F_{\mathrm{Sol}} + C_{\mathrm{Sol}}))/F_{\mathrm{Sol}}$ which, by our assumption, implies that q has $\mathrm{ratio}(q) > 0$ and thus could be selected for the next iteration of the algorithm, implying that i is not the final iteration. Thus, by contradiction, there is an iteration i where $C_i \leq F_{\mathrm{Sol}} + C_{\mathrm{Sol}}$. Let m be the first iteration with this property.

We will now show that $C_m + F_m \leq F + F_{\text{Sol}} \max\left\{0, \ln \frac{C - C_{\text{Sol}}}{F_{\text{Sol}}}\right\} F_{\text{Sol}} + C_{\text{Sol}}$:

$$
\begin{aligned}
C_m + F_m &= F + C_m + \sum_{i=1}^{m} (F_i - F_{i-1}) \\
&\leq F + C_m + F_{\text{Sol}} \sum_{i=1}^{m} \left(\frac{C_{i-1} - C_i}{C_{i-1} - C_{\text{Sol}}} \right) \\
&\leq F + C_m + F_{\text{Sol}} \sum_{i=1}^{m} \left(1 - \frac{C_i - C_{\text{Sol}}}{C_{i-1} - C_{\text{Sol}}} \right) \qquad (3.14)
\end{aligned}
$$

Note that $C_i > C_{\text{Sol}}$ for all $i < m$ because, by choice of m, $C_i > C_{\text{Sol}} + F_{\text{Sol}}$.

Now, to maximize expression (3.14), consider its partial derivative with respect to C_m, namely $1 - \frac{F_{\text{Sol}}}{C_{m-1} - C_{\text{Sol}}}$, which is strictly greater than 0 since $C_{m-1} > F_{\text{Sol}} + C_{\text{Sol}}$. This implies that (3.14) will reach its maximum value when C_m is maximum, which occurs at $C_m = F_{\text{Sol}} + C_{\text{Sol}}$. Thus, we can freely assume from now on that C_m is equal to $F_{\text{Sol}} + C_{\text{Sol}}$.

Also, we have that, for all $i \in \{1, 2, \ldots, m\}$, $1 - \frac{C_i - C_{\text{Sol}}}{C_{i-1} - C_{\text{Sol}}} \leq \ln \left(\frac{C_{i-1} - C_{\text{Sol}}}{C_i - C_{\text{Sol}}} \right)$, because, for all real $x > 0$, $1 - x \leq \ln(1/x)$ due to Lemma 2.2.11. Here, we assume that $C_m > C_{\text{Sol}}$ because $C_m = F_{\text{Sol}} + C_{\text{Sol}}$ and, if F_{Sol} were zero, we would have the trivial case where all facilities in $\mathcal{F}^*$ could be opened. Thus,

$$
\begin{aligned}
C_m + F_m &\leq F + C_m + F_{\text{Sol}} \sum_{i=1}^{m} \ln \left(\frac{C_{i-1} - C_{\text{Sol}}}{C_i - C_{\text{Sol}}} \right) \\
&\leq F + C_m + F_{\text{Sol}} \ln \left(\frac{C - C_{\text{Sol}}}{C_m - C_{\text{Sol}}} \right) \\
&= F + C_{\text{Sol}} + F_{\text{Sol}} + F_{\text{Sol}} \ln \left(\frac{C - C_{\text{Sol}}}{F_{\text{Sol}}} \right)
\end{aligned}
$$

where the last step was given by our assumption that $C_m = C_{\text{Sol}} + F_{\text{Sol}}$.

Since all iterations that follow m will only reduce the solution's cost, we have arrived at the desired conclusion. □

Theorem 3.2.5 (Adapted from Theorem 1 of [22])**.** *Let $\mathcal{I}$ be an instance of the UFLP and* Sol *be any (possibly fractional) solution to $\mathcal{I}$ where F is the total opening cost of* Sol *and C is the total connection cost of* Sol *(in particular,* Sol *could be the optimal solution for $\mathcal{I}$). If it is possible to generate a solution* Sol_1 *such that*

$$\delta F_{\text{Sol}_1} + C_{\text{Sol}_1} \leq \gamma_f \delta F + \gamma_c C,$$

for real paramaters $\delta, \gamma_f, \gamma_c > 0$, where F_{Sol_1} is the opening cost of Sol_1 *and C_{Sol_1} is the connection cost of* Sol_1*, then, if we apply greedy augmentation to* Sol_1*, resulting in solution* Sol_2*, we conclude that*

$$\text{cost}(\text{Sol}_2) \leq \max\left(\gamma_f + 1 - \frac{1}{\delta}, \gamma_f + \ln \delta, 1 + \frac{\gamma_c - 1}{\delta} \right) \text{cost}(\text{Sol}).$$

Proof. The proof is divided in two cases according to whether $C_{\text{Sol}_1} < F + C$ or $C_{\text{Sol}_1} \geq$

$F + C$.

In the first case, assume $C_{\mathrm{Sol}_1} < F + C$. Thus,

$$\begin{aligned} F_{\mathrm{Sol}_1} + C_{\mathrm{Sol}_1} &= \frac{\delta F_{\mathrm{Sol}_1} + C_{\mathrm{Sol}_1}}{\delta} + \left(1 - \frac{1}{\delta}\right) C_{\mathrm{Sol}_1} \\ &\leq \frac{\gamma_f \delta F + \gamma_c C}{\delta} + \left(1 - \frac{1}{\delta}\right)(C + F) \\ &= \left(\gamma_f + 1 - \frac{1}{\delta}\right) F + \left(1 + \frac{\gamma_c - 1}{\delta}\right) C \\ &\leq \max\left(\gamma_f + 1 - \frac{1}{\delta}, 1 + \frac{\gamma_c - 1}{\delta}\right) \mathrm{cost}(\mathrm{Sol}) \end{aligned}$$

And, since greedy augmentation does not increase cost, $\mathrm{cost}(\mathrm{Sol}_2) \leq \mathrm{cost}(\mathrm{Sol}_1)$ and the result follows.

Now, assume $C_{\mathrm{Sol}_1} \geq F + C$. This assumption implies in $\ln\left(\frac{C_{\mathrm{Sol}_1} - C}{F}\right) \geq 0$, thus, by Lemma 3.2.4, we know that

$$\mathrm{cost}(\mathrm{Sol}_2) \leq F_{\mathrm{Sol}_1} + F \ln\left(\frac{C_{\mathrm{Sol}_1} - C}{F}\right) + F + C$$

By our initial assumption, we have that $C_{\mathrm{Sol}_1} \leq \gamma_f \delta F + \gamma_c C - \delta F_{\mathrm{Sol}_1}$, and, as a consequence:

$$\mathrm{cost}(\mathrm{Sol}_2) \leq F_{\mathrm{Sol}_1} + F \ln\left(\frac{\gamma_f \delta F + (\gamma_c - 1)C - \delta F_{\mathrm{Sol}_1}}{F}\right) + F + C$$

We will maximize the right-hand expression, choosing a appropriate value for F_{Sol_1}. Notice that $0 \leq F_{\mathrm{Sol}_1} \leq (\gamma_f - 1)F + \frac{\gamma_c - 1}{\delta} C$ and the partial derivate of the expression with respect to F_{Sol_1} is non-negative in this interval. Thus, the expression is maximized at $F_{\mathrm{Sol}_1} = (\gamma_f - 1)F + \frac{\gamma_c - 1}{\delta} C$ and, as a consequence,

$$\begin{aligned} \mathrm{cost}(\mathrm{Sol}_2) &\leq (\gamma_f + \ln \delta)F + \left(1 + \frac{\gamma_c - 1}{\delta}\right) C \\ &\leq \max\left(\gamma_f + \ln \delta, 1 + \frac{\gamma_c - 1}{\delta}\right) \mathrm{cost}(\mathrm{Sol}) \end{aligned}$$

And we have obtained the desired result also in this case. □

And, finally, we can prove, combining Theorem 3.2.5 and Theorem 3.2.1, that, at the end of Phase 3, the obtained solution is a 1.86-approximation.

Theorem 3.2.6. *The algorithm described in this section is a* 1.86*-approximation.*

Proof. The solution Sol_1 after Phase 2, if we apply cost scaling, has the following property due to Theorem 3.2.1:

$$3\delta F_{\mathrm{Sol}_1} + C_{\mathrm{Sol}_1} \leq 3(\delta F_{\mathrm{Sol}} + C_{\mathrm{Sol}})$$

where F_x and C_x are, respectively, the total opening and connection cost of solution $x \in \{\mathrm{Sol}_1, \mathrm{Sol}\}$ and Sol is an optimal solution to the instance without scaling.

This property ensures, via Theorem 3.2.5, that the algorithm has an approximation factor of

$$\max\left(2 - \frac{1}{3\delta}, 1 + \ln(3\delta), 1 + \frac{2}{3\delta}\right)$$

Setting $\delta = 0.782$, shows that the algorithm is a 1.86-approximation. $\square$

Chapter 4

Approximations for the k-Level Stochastic Facility Location Problem

In this chapter, we will consider the k-Level Stochastic Facility Location Problem (k-LSFLP), which is a combination of both the k-LFLP and the Stochastic FLP (refer to Chapters 2 and 3 for more details on these problems). Let us first define the problem precisely.

An instance of the k-LSFLP is comprised by a set of clients $\mathcal{C}$, a set of facilities $\mathcal{F}$, a set of scenarios $\mathcal{A}$, a metric c over $\mathcal{C} \cup \mathcal{F}$ and facility opening costs.

As in the SFLP, decisions are separated in two time periods called *stages*. In the first stage, the set of participating clients is not known, but some facilities can be opened in advance. In the second stage, exactly one scenario randomly occurs, revealing the set of participating clients and possibly changing the opening cost for some facilities. The cost of opening facility i in the first stage is denoted by f_i^{I}. For each scenario $A \in \mathcal{A}$, the set of participating clients is $\mathcal{C}^A$, the opening cost of facility $i \in \mathcal{F}$ during that scenario is f_i^A and the probability of A ocurring is given by p_A.

As in the k-LFLP, each facility belongs to one of k levels and we denote the set of all facilities at level ℓ by $\mathcal{F}_\ell$. However, we can only connect clients by chains of facilities in the second stage, when the set of participating clients is known. Thus, for each scenario A, every client $j \in \mathcal{C}^A$ must be connected by a chain of facilities $\phi^A(j) = (i_1, i_2, \ldots, i_k) \in \mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \times \mathcal{F}_k$ resulting in a connection cost of $d(j, \phi^A(j)) = c_{ji_1} + \sum_{p=2}^{k} c_{i_{p-1}i_p}$.

Therefore, a feasible solution to this problem is comprised by: a subset of facilities F^{I} to be opened in the first stage; for each $A \in \mathcal{A}$, a subset of facilities F^A to be opened in the second stage if scenario A is realized; and, for each $A \in \mathcal{A}$ and $j \in \mathcal{C}^A$, a sequence $\phi^A(j) = (i_1, i_2, \ldots, i_k)$ such that $i_\ell \in F^{\mathrm{I}} \cup F^A$ is an open facility of level ℓ. An optimal solution minimizes the expected cost, which is given by the following expression

$$\textstyle\sum_{i \in F^I} f_i^I + \sum_{A \in \mathcal{A}} p_A \left(\sum_{i \in F^A} f_i^A + \sum_{j \in \mathcal{C}^A} d(j, \phi^A(j)) \right).$$

We may refer to the solution's expected cost from here onwards simply as the solution's cost to avoid confusion when dealing with randomized algorithms for this problem.

We also consider a variant of the k-LSFLP introduced by Wang et al. [31], where a client may be connected only to facilities opened in the same stage. That is, for every $A \in \mathcal{A}$, and $j \in C^A$, if $\phi^A(j) = (i_1, i_2 \ldots, i_k)$, then either $\{i_1, i_2, \ldots, i_k\} \subseteq F^{\mathrm{I}}$, or $\{i_1, i_2, \ldots, i_k\} \subseteq F^A$. A facility can be opened in both stages if it helps minimize the total cost, in which case, both its first and second stage opening costs must be paid. In this thesis, we will refer to this variant as the k-Level Stochastic Stage-Constrained Facility Location Problem (k-LSSCFLP).

Note that this variant is only applicable to situations where it is possible to distinguish the stages in which facilities were opened, since constraining the stage of facilities to which a client can connect may impose an unbounded increase in cost that would be otherwise unjustified. For instance, if we have $k = 2$, $\mathcal{A} = \{A\}$, $\mathcal{F}_1 = \{i_1\}$ and $\mathcal{F}_2 = \{i_2\}$ such that $f^I_{i_1} = f^A_{i_2} = 1$ and $f^A_{i_1} = f^I_{i_2} = M$, where M is a large positive number; it is better to open facility i_1 during the first stage and facility i_2 during the second stage for a total opening cost of 2, while the stage constrained version would result in a cost of at least $1 + M \gg 2$.

For the k-LSSCFLP, Wang et al. [33] presented a combinatorial 7-approximation algorithm, and Wang et al. [32] presented a 4-approximation based on linear programming. Recently, Wu et al. [36] presented a primal dual $(3 + \epsilon)$-approximation for the metric 2-LSSCFLP. In Section 4.1, we will show an algorithm reminiscent of [32] that adapts Byrka and Rybicki [9] k-LFLP approximation to obtain better approximation factors for all $k \geq 2$.

For the k-LSFLP, to the best of our knowledge, there were no previous known results at the time of writing. We will show in Section 4.2 how to obtain a $(2k - 1 + o(1))$-approximation that has an approximation ratio of 3.495, 5.157 and 7.034 for respectively $k = 2$, $k = 3$ and $k = 4$.

4.1 k-LSSCFLP

In this section, we demonstrate that Byrka and Rybicki's $(3 - o(1))$-approximation for the k-Level Facility Location Problem (k-LFLP) [9] (see Section 2.2 for a detailed review) can be adapted to provide a $(4 - o(1))$-approximation for the k-LSSCFLP. Their algorithm is based on the rounding of an LP-formulation based on trees of facilities. Once the LP relaxation is solved, the fractional openings of tree nodes and a clustering procedure are used to recursively round each tree. In the stochastic case, where facilities might be opened during the first stage or in one of the scenarios during the second stage, we create, for every tree in the original formulation, one copy of that tree corresponding to the first stage and $|\mathcal{A}|$ copies corresponding to each one of the scenarios. After obtaining a fractional solution to the LP relaxation, we scale fractional openings and connections, restricting the rounding procedure for every client-scenario pair to one of the stages according to which stage receives more connections for that client-scenario pair. The following table shows the approximation factor obtained by this approach some values of k:

k	1	2	3	4	5	6	7
factor	2.27	2.56	2.78	2.95	3.07	3.17	3.25

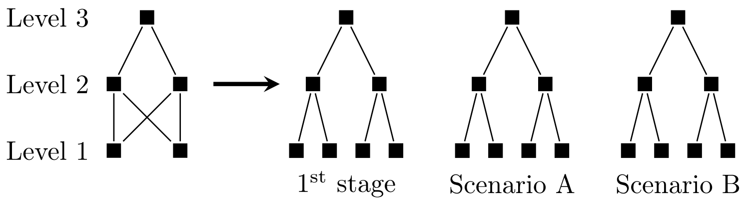

Figure 4.1: Original graph and obtained forest.

4.1.1 LP Formulation

Our LP formulation is similar to the one presented by Byrka and Rybicki, with the addition of structure related to the stochastic nature of this problem.

Next, we obtain a forest $\mathcal{R}$ by making copies of trees in $\mathcal{R}'$: for each tree of $\mathcal{R}'$, we make one copy corresponding to the first stage, and one copy for each scenario of the second stage. Formally, we define $\mathcal{P}_\ell = \mathcal{P}'_\ell \times (\{\mathrm{I}\} \cup \mathcal{A})$, for $1 \leq \ell \leq k$, $\mathcal{P}_F := \cup_{\ell=1}^{k} \mathcal{P}_\ell$, and $\mathcal{R}$ is the corresponding induced forest on $\mathcal{P}_F$. An example of the obtained forest is given in Figure 4.1.

Let $\mathcal{C}'$ be the set of all pairs (j, A), henceforth denoted by j^A, such that $A \in \mathcal{A}$ and $j \in \mathcal{C}^A$. In the following, we will interpret each j^A as a client that must be served in scenario A. Let $\mathcal{P}_C^I$ be the set of tuples of the form* $(j^A, i_1, i_2, \ldots, i_k)_I \in \mathcal{C}' \times \mathcal{P}_1$. Also, for a fixed $A \in \mathcal{A}$, let $\mathcal{P}_C^A$ be the set of tuples of the form $(j^A, i_1, i_2, \ldots, i_k)_A \in \mathcal{C}' \times \mathcal{P}_1$. Each tuple in $\mathcal{P}_C^X$, $X \in \{\mathrm{I}, A\}$, starting with a client j^A represents a path of facilities that j^A may use to satisfy its demand, that is, a path whose facilities should all be opened in the first stage or all be opened in scenario A. The set of all such paths is $\mathcal{P}_C = \bigcup_{X \in \{\mathrm{I}\} \cup \mathcal{A}} \mathcal{P}_C^X$, and the set of all considered paths is $\mathcal{P} = \mathcal{P}_C \cup \mathcal{P}_F$.

We define costs for each element of $\mathcal{P}$ in the following way. Recall that p_A is the probability of scenario A occurring, and c is the metric function over $\mathcal{C} \cup \mathcal{F}$. For each path $p = (j^A, p')_X \in \mathcal{P}_C$, where $X \in \{I, A\}$, define $c_p = p_A \cdot d(j, p')$, where $d(j, p') = c_{ji_1} + \sum_{\ell=2}^{k} c_{i_{\ell-1} i_\ell}$. This corresponds to the increase in the solution's cost caused by serving client j in scenario A by the path $(i_1, i_2, \ldots, i_k)$ where facilities were opened either in the first stage, when $X = \mathrm{I}$, or in scenario X, when $X \in \mathcal{A}$. For $p = (i_\ell, i_{\ell+1}, \ldots, i_k)_X \in \mathcal{P}_F$, define c_p as the amount increased in the solution's cost if facility i_ℓ were opened in scenario X, if $X \in \mathcal{A}$, or the cost of opening it in the first stage, if $X = \mathrm{I}$. Hence, c_p is equal to

*We made a small abuse of notation to improve readability: $(\ldots, i_{k-1}, i_k)_X \triangleq (\ldots, i_{k-1}, i_k, X)$.

$p_X f^X_{i_\ell}$, if $X \in \mathcal{A}$, or $c_p = f^I_{i_\ell}$, if $X =$ I. The resulting ILP for the k-LSSCFLP is given by:

$$\begin{aligned}
\min \quad & \sum_{p \in \mathcal{P}} c_p x_p && && (4.1)\\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}_C : j^A \in p} x_p \geq 1 && \forall\, j^A \in \mathcal{C}' && (4.2)\\
& \sum_{p \in \mathcal{P}_C : j^A \in p, p \sqsupset q} x_p \leq x_q && \forall\, j^A \in \mathcal{C}',\ \forall q \in \mathcal{P} \setminus \mathcal{P}_C && (4.3)\\
& x_p \leq x_{\pi(p)} && \forall\, p \in \mathcal{P} \setminus (\mathcal{P}_C \cup \mathcal{P}_k) && (4.4)\\
& x_p \in \{0,1\} && \forall\, p \in \mathcal{P} && (4.5)
\end{aligned}$$

where $p \sqsupset q$ denotes that tuple q is suffix of tuple p, and $\pi(p)$ is the parent path of path p in forest $\mathcal{R}$. Note that this means that, if $p \sqsupset q$, both p and q belong in the same scenario/stage, also note that all client-scenario pairs have paths that belong in the first stage, the ones in $\mathcal{P}^I_C$. For each $p = (j^A, i_1, \dots, i_k)_X \in \mathcal{P}_C$, variable x_p indicates whether client j in scenario A is served by the path $(i_1, i_2, \dots, i_k)$ of facilities opened either in the first stage, when $X =$ I, or in the scenario X, if $X \in \mathcal{A}$. Variables x_p for all $p \in \mathcal{P} \setminus \mathcal{P}_C$ indicate whether the tree node p is opened. Inequalities (4.2) state that every client j in a scenario A is served by a path. Inequalities (4.3) state that if a client is served by a path, the tree nodes in that path are opened. Inequalities (4.4) state that, whenever a tree node is open, its parent node is also open. In the relaxed version, integrality constraints (4.5) are replaced by inequalities $x_p \geq 0$ for all $p \in \mathcal{P}$.

The dual program of the relaxed LP is:

$$\begin{aligned}
& \max \quad \sum_{j^A \in \mathcal{C}'} v^A_j \\
\text{s.t.} \quad & v^A_j - \sum_{q \in \mathcal{P} \setminus \mathcal{P}_C : q \sqsupset p} w^A_{j,q} \leq c_p && \forall\, p = (j^A, \dots) \in \mathcal{P}_C \qquad (4.6)\\
& -y_p + \sum_{j^A \in \mathcal{C}'} w^A_{j,p} \leq c_p && \forall\, p \in \mathcal{P}_1\\
& -y_p + \sum_{q \in \mathcal{P}_{\ell-1} : p \sqsupset q} y_q + \sum_{j^A \in \mathcal{C}'} w^A_{j,p} \leq c_p && \forall\, \ell \in \{2, \dots, k-1\},\ p \in \mathcal{P}_\ell\\
& \sum_{q \in \mathcal{P}_{k-1} : p \sqsupset q} y_q + \sum_{j^A \in \mathcal{C}'} w^A_{j,p} \leq c_p && \forall\, p \in \mathcal{P}_k\\
& v^A_j, w^A_{j,p}, y_p \geq 0 && \forall\, p, j^A
\end{aligned}$$

4.1.2 Clustering and rounding

Based on an optimal solution of the relaxed LP, Byrka and Rybicki's algorithm [9] obtains a clustering of the clients such that, for each cluster, one client is considered the center. By using a careful randomized opening procedure, the expected facility opening cost becomes equal to the opening cost of the LP. Moreover, the procedure guarantees that there is a path in $\mathcal{P}_C$ of opened facilities that starts at the cluster center and has a "small" expected length. Non-center clients can utilize this path as "backup" if there is no preferable path of opened facilities nearby. However, note that there is at least a constant probability of preferable paths being available.

In the stochastic case, this strategy cannot be used directly, since some paths associated with a cluster center may not belong to the scenario of a non-center client in the same cluster. Therefore, before the clustering procedure, we decide at which stage each client will be served and we restrict clusters to only contain clients that will be served in the same stage. In our algorithm, a client $j^A \in \mathcal{C}'$ is clustered in the first stage whenever $\sum_{p \in \mathcal{P}_C^{\mathrm{I}} : j^A \in p} x_p \geq 1/2$, where $\mathbf{x}$ is an optimum solution to the relaxed LP. Otherwise, it is clustered in the second stage.

Define $\mathcal{P}^{Cl}(j^A) = \{p \in \mathcal{P}_C^X : j^A \in p, x_p > 0\}$ where $X = \mathrm{I}$, if j^A is clustered in the first-stage, and $X = A$ otherwise. Notice that $\mathcal{P}^{Cl}(j^A)$ contains all paths that serve j^A in the *stage* it was clustered. We will obtain a modified feasible solution $\bar{\mathbf{x}}$ from $\mathbf{x}$, where each client is served only in one stage. Solution $\bar{\mathbf{x}}$ is defined as follows: for $p \in \mathcal{P} \setminus \mathcal{P}_C$, $\bar{x}_p = 2x_p$; for every pair j^A and path $p \in \mathcal{P}^{Cl}(j^A)$, $\bar{x}_p = x_p / \sum_{q \in \mathcal{P}^{Cl}(j^A)} x_q$; and, for every remaining path p, $\bar{x}_p = 0$.

All pairs j^A are inspected in non-decreasing order of $d_{j,A}^{\max} + \overline{C}_{j,A}$, where $d_{j,A}^{\max}$ is the maximum length of a path in $\mathcal{P}^{Cl}(j^A)$ and $\overline{C}_{j,A}$ is the average path length in $\mathcal{P}^{Cl}(j^A)$ weighted by the corresponding fractional opening in $\bar{\mathbf{x}}$. Whenever an unclustered pair j^A is inspected, a new cluster is created, composed of j^A, which is also the cluster center, and all unclustered pairs $j'^{A'}$ such that there are paths $p \in \mathcal{P}^{Cl}(j^A)$ and $p' \in \mathcal{P}^{Cl}(j'^{A'})$ that belong to the same stage and scenario (if applicable) and reach the same k-th level facility, that is, there exist $i_k \in \mathcal{F}_k$, and $X \in \{\mathrm{I}\} \cup \mathcal{A}$ such that $p = (j^A, \ldots, i_k)_X$, and $p' = (j'^{A'}, \ldots, i_k)_X$.

In the following, we will use the randomized rounding procedure by [9], denoted by ROUND($\cdot$) as a black-box, that receives a tree from $\mathcal{R}$ rooted at some vertex $i_k \in \mathcal{F}_k$, a fractional solution $\mathbf{x}$ to the LP associated with $\mathcal{R}$ and a bit that informs whether a *token* was placed in i_k, and returns a set of facilities from the tree to open. Refer to Section 2.2.3 for more details and results. In this procedure, whenever a token is passed, we are guaranteed that there is at least one path starting at a first level whose facilities are all opened.

For each cluster center j^A, we randomly select one path $p \in \mathcal{P}^{Cl}(j^A)$ with probability $\bar{x}_p$ and place a token in the root node at the end of this path. Finally, we execute the procedure ROUND($\cdot$) on the root of each tree passing the fractional openings of $\bar{\mathbf{x}}$. This guarantees that, for each cluster center j^A, at least one path of $\mathcal{P}^{Cl}(j^A)$ is opened.

4.1.3 Algorithm and analysis

In summary, the algorithm executes the following steps:

1. Solve the relaxed LP and obtain a solution $\mathbf{x}$;

2. For every pair $j^A \in \mathcal{C}'$, if $\sum_{p \in \mathcal{P}_C^{\mathrm{I}} : j^A \in p} x_p \geq 1/2$, classify it as first-stage served, otherwise it is second-stage served;

3. Obtain the feasible solution $\bar{\mathbf{x}}$ and execute the clustering procedure, distributing tokens accordingly, as defined in Section 4.1.2;

4. Execute procedure ROUND($\cdot$) according to $\bar{\mathbf{x}}$ and the distribution of tokens;

5. For every scenario A, connect client $j \in \mathcal{C}^A$ in that scenario to the shortest path $i_1 i_2 \ldots i_k$ such that every facility i_ℓ in the path belongs to the ℓ-th level and all facilities were opened during the same stage and are available in scenario A.

The above algorithm solves a linear program of size $O(mn^k|\mathcal{A}|)$ where n is the maximum number of facilities per level and m is the number of clients and, if k is made constant, the size is polynomial.

For every scenario $A \in \mathcal{A}$ and client $j \in \mathcal{C}^A$, let $P_{j,A}$ be the random variable that represents the path assigned to j during scenario A and $C_{j,A} \triangleq d(j, P_{j,A})$ be the length of the path $jP_{j,A}$. The next lemma, which is very similar to Lemma 2.2.7, bounds the expected value of $C_{j,A}$:

Lemma 4.1.1. *Let $j^A \in \mathcal{C}'$, then*

$$E[C_{j,A}] \leq 2F_k(2)C^*_{j,A} + 2(1 - F_k(2))(d^{\max}_{j,A} + C^*_{j,A}),$$

where $C^*_{j,A} = \sum_{p \in \mathcal{P}_1} x_{(j^A,p)} d(j,p)$.

Proof. By applying Lemma 2.2.7, we obtain that:

$$E[C_{j,A}] \leq F_k(2)\overline{C}_{j,A} + (1 - F_k(2))(2d^{\max}_{j,A} + \overline{C}_{j,A}),$$

where $\overline{C}_{j,A} = \sum_{p \in \mathcal{P}_1^X} \bar{x}_{(j^A,p)} d(j,p)$ with $X = A$ if j^A is served in the second stage and $X = I$ if j^A is served in the first stage.

To obtain the desired result, we only need to show that $\overline{C}_{j,A} \leq 2C^*_{j,A}$:

$$\begin{aligned}
\overline{C}_{j,A} &= \sum_{p \in \mathcal{P}_1^X} \bar{x}_{(j^A,p)} d(j,p) \\
&= \sum_{p \in \mathcal{P}_1^X} x_{(j^A,p)} d(j,p) \Bigg/ \sum_{p \in \mathcal{P}_1^X} x_{(j^A,p)} \\
&\leq \sum_{p \in \mathcal{P}_1} x_{(j^A,p)} d(j,p) \Bigg/ \sum_{p \in \mathcal{P}_1^X} x_{(j^A,p)} \\
&\leq 2 \sum_{p \in \mathcal{P}_1} x_{(j^A,p)} d(j,p) \\
&= 2C^*_{j,A}
\end{aligned}$$

□

Theorem 4.1.2. *The expected solution cost generated by the rounding algorithm is at most* $(4 - 2F_k(2))\mathrm{OPT}$, *where* OPT *is the optimal cost.*

Proof. The expect cost of opening facilities through the rounding algorithm equals to the opening cost in the feasible LP solution as demonstrated in Lemma 2.2.6. Since in the solution $\bar{\mathbf{x}}$ we have doubled the opening of $\mathbf{x}$, the expected opening cost is $2F^*$, where F^* is the total opening cost associated with $\mathbf{x}$. Now we can use Lemma 4.1.1 to bound the expected value of the solution's cost:

$$\begin{aligned}
E[\text{Cost}] &\le 2F^* + \sum_{j\in\mathcal{C},A\in\mathcal{A}} p_A E[C_{j,A}] \\
&\le 2F^* + \sum_{j\in\mathcal{C},A\in\mathcal{A}} p_A(2F_k(2)C^*_{j,A} + 2(1-F_k(2))(d^{\max}_{j,A} + C^*_{j,A}))\,.
\end{aligned}$$

By Inequality (4.6) of the dual LP program, and using complementary slackness we obtain $d^{\max}_{j,A} \le v^A_j/p_A$. Let C^* be the total connection cost associated with $\mathbf{x}$. We get:

$$\begin{aligned}
E[\text{Cost}] &\le 2F^* + \sum_{j\in\mathcal{C},A\in\mathcal{A}} p_A(2F_k(2)C^*_{j,A} + 2(1-F_k(2))(d^{\max}_{j,A} + C^*_{j,A})) \\
&\le 2F^* + 2C^* + 2(1-F_k(2)) \sum_{j\in\mathcal{C},A\in\mathcal{A}} v^A_j \\
&\le 2F^* + 2C^* + 2(1-F_k(2))(F^* + C^*) \\
&\le (4 - 2F_k(2))\text{OPT}.
\end{aligned}$$

□

To bound the function $F_k(\gamma)$, we can simply use Theorem 2.2.13.

4.2 k-LSFLP

In this section, we describe a 3.495-approximation algorithm for 2-LSFLP that can be extended to a $(2k-1+o(1))$-approximation for the k-LSFLP. Our algorithm splits one problem instance into two 1-LSFLP instances that can be solved individually by a known algorithm. The main idea is that, after solving each level separately, one may assign each pair client-scenario to one opened facility of each level, resulting in worse performance for instances whose optimal solutions have cost dominated by the connection cost. To mitigate this issue, before solving each one-level instance, we scale all opening costs by a given parameter δ, solve both instances and, then, apply the greedy augmentation technique [14] to each solution.

4.2.1 Algorithm

The algorithm proceeds as follows. Given a 2-LSFLP instance $\mathcal{I}$, it creates two 1-LSFLP instances:

1. $\mathcal{I}_1$ containing all scenarios, clients and first-level facilities; connection costs are twice the corresponding values in $\mathcal{I}$, and opening costs are equal to the opening costs in $\mathcal{I}$;

2. $\mathcal{I}_2$ containing all scenarios, clients and second-level facilities; connection costs and opening costs are equal to the corresponding values in $\mathcal{I}$.

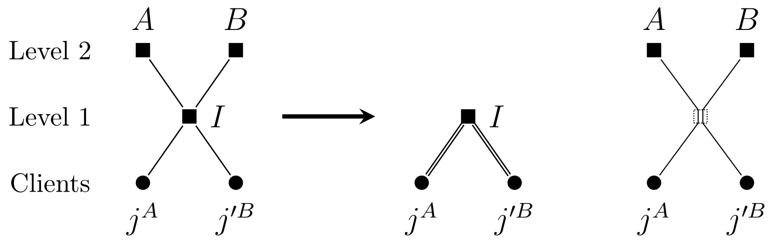

Figure 4.2: Optimal solution for original problem, and solutions $\mathrm{Sol}_1^\dagger$ and $\mathrm{Sol}_2^\dagger$ for instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. Connections costs in $\mathrm{Sol}_1^\dagger$ are doubled.

Then, it solves both 1-LSFLP instances by applying Ye and Zhang's primal-dual algorithm [37] (see Chapter 3) with opening costs scaled by a parameter $\delta > 0$, and applies greedy augmentation to each solution individually. Finally, it combines both solutions by opening the same facilities in the same stage/scenario and, for each scenario A and client $j \in \mathcal{C}^A$, whenever A is realized, j is connected to the shortest path of available facilities.

Now we detail the greedy augmentation in the context of solutions to a 1-LSFLP instance. For each $(u, X) \in \mathcal{F} \times (\{\mathrm{I}\} \cup \mathcal{A})$, "opening (u, X)" is defined as opening facility u either when scenario X is realized (if $X \in \mathcal{A}$), or in the first stage (if $X = \mathrm{I}$). Given a solution for 1-LSFLP, for every pair $(u, X) \in \mathcal{F} \times (\{\mathrm{I}\} \cup \mathcal{A})$, define $\mathrm{gain}(u, X) = C - C' - \bar{f}_u^X$, where C is the expected connection cost of current solution, C' is the expected connection cost after (u, X) is opened and, for each scenario, clients are connected to the shortest available path of facilities, and $\bar{f}_u^X$ is the expected cost of opening (u, X), that is, $\bar{f}_u^X = p_X f_u^X$ if $X \in \mathcal{A}$, and $\bar{f}_u^X = f_u^\mathrm{I}$ if $X = \mathrm{I}$.

For a solution to an instance of the FLP, the greedy augmentation procedure consists in iteratively selecting a pair $(u, X) \in \mathcal{F} \times (\{\mathrm{I}\} \cup \mathcal{A})$ that maximizes $\mathrm{gain}(u, X)/\bar{f}_u^X$ and opening it, until there is no pair (u, X) with $\mathrm{gain}(u, X) > 0$.

4.2.2 Analysis

For the analysis, we first obtain feasible solutions to both $\mathcal{I}_1$ and $\mathcal{I}_2$ from an optimal solution to $\mathcal{I}$. Given an optimal solution of cost $F^* + C^*$ for instance $\mathcal{I}$, where F^* is the total opening cost and C^* is the total connection cost, we create two feasible solution $\mathrm{Sol}_1^\dagger$ and $\mathrm{Sol}_2^\dagger$ for instances $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. For each $\ell \in \{1, 2\}$, we proceed as follows (see Figure 4.2): solution $\mathrm{Sol}_\ell^\dagger$ opens all ℓ-th level facilities in the same stage/scenario as in the optimal solution for the original problem; and, for every scenario $A \in \mathcal{A}$ and client $j \in \mathcal{C}^A$ connected to a path (j, i_1, i_2) in the original solution, connect client j in scenario A to facility i_ℓ (notice that i_ℓ may be opened either in the first stage, or in scenario A).

Let the cost of $\mathrm{Sol}_\ell^\dagger$ (for $\ell \in \{1, 2\}$) be equal to $F_\ell^* + C_\ell^*$ where F_ℓ^* is the total opening cost and C_ℓ^* the total connection cost. It is easy to show that $F_1^* + F_2^* = F^*$ and $C_1^* + C_2^* \leq 3C^*$ (see Figure 4.2).

By reasoning as in the proof of Theorem 3.2.5, it is possible to show the following lemma:

Lemma 4.2.1. *For $\ell \in \{1, 2\}$, let Sol_ℓ be a feasible solution to instance $\mathcal{I}_\ell$ with opening costs scaled by $\delta > 0$. Let the total cost of Sol_ℓ be $\delta F_{\mathrm{Sol}_\ell} + C_{\mathrm{Sol}_\ell}$ where $\delta F_{\mathrm{Sol}_\ell}$ is the total* *(scaled) opening cost and C_{Sol_ℓ} is the total connection cost. If $3\delta F_{\mathrm{Sol}_\ell} + C_{\mathrm{Sol}_\ell} \le 3(\delta F^*_\ell + C^*_\ell)$ then, after applying greedy augmentation to Sol_ℓ, the resulting solution has total cost upper bounded by*

$$\max\left(1 + \ln 3\delta, 2 - \frac{1}{3\delta}\right) F^*_\ell + \left(1 + \frac{2}{3\delta}\right) C^*_\ell .$$

We can now obtain the approximation factor.

Theorem 4.2.2. *The algorithm described in Section 4.2.1 is a 3.495-approximation for the 2-LSFLP.*

Proof. For $\ell \in \{1, 2\}$, let Sol_ℓ be the solution obtained by the Ye and Zhang's primal-dual procedure [37] to the instance $\mathcal{I}_\ell$ with opening costs scaled by a parameter $\delta > 0$. Let the cost of Sol_ℓ be $\delta F_{\mathrm{Sol}_\ell} + C_{\mathrm{Sol}_\ell}$. The primal-dual guarantees that: $3\delta F_{\mathrm{Sol}_\ell} + C_{\mathrm{Sol}_\ell} \le 3\,\mathrm{OPT}_\ell$, where OPT_ℓ is the cost of an optimal solution to $\mathcal{I}_\ell$ with scaled opening costs. Since solution $\mathrm{Sol}^\dagger_\ell$ is feasible, we conclude that $3\delta F_{\mathrm{Sol}_\ell} + C_{\mathrm{Sol}_\ell} \le 3(\delta F^*_\ell + C^*_\ell)$ and we can apply Lemma 4.2.1:

$$\mathrm{cost}(\mathrm{Sol}'_\ell) \le \max\left(1 + \ln 3\delta, 2 - \frac{1}{3\delta}\right) F^*_\ell + \left(1 + \frac{2}{3\delta}\right) C^*_\ell,$$

where $\mathrm{cost}(\mathrm{Sol}'_\ell)$ is the solution cost after applying greedy augmentation to Sol_ℓ.

Notice that the sum $\mathrm{cost}(\mathrm{Sol}'_1)$ and $\mathrm{cost}(\mathrm{Sol}'_2)$ is an upper bound to the cost of the final solution, since we open the same facilities in the same stages/scenarios, and we can connect, without increasing connection costs, each client $j \in \mathcal{C}^A$ in a scenario $A \in \mathcal{A}$, to a path (j, i_1, i_2), where i_ℓ ($\ell \in \{1, 2\}$) is the facility of Sol'_ℓ to which j was connected in scenario A.

Thus, the cost of the final solution can be upper bounded by

$$\begin{aligned} &\max\left(1 + \ln 3\delta, 2 - \frac{1}{3\delta}\right)(F^*_1 + F^*_2) + \left(1 + \frac{2}{3\delta}\right)(C^*_1 + C^*_2) \\ \le &\max\left(1 + \ln 3\delta, 2 - \frac{1}{3\delta}\right) F^* + 3\left(1 + \frac{2}{3\delta}\right) C^* . \end{aligned}$$

By setting $3\delta = 12.1215$, we obtain the desired result. □

The same idea can be generalized for k levels, resulting in a $(2k - 1 + o(1))$-approximation. Given an instance $\mathcal{I}$, first, create k instances, each associated with facilities of one of the k levels, and double the connection costs of instances for the first $k-1$ levels. To obtain an approximate solution, scale the opening costs by an appropriate parameter $\delta > 0$, solve each instance with Ye and Zhang's primal-dual procedure, applying greedy augmentation, and, finally, open the facilities opened in the solutions of each one-level instance (at exactly the same stage/scenario), connecting clients by shortest paths.

Theorem 4.2.3. *The algorithm described in the paragraph above is a $(2k - 1 + o(1))$-approximation.*

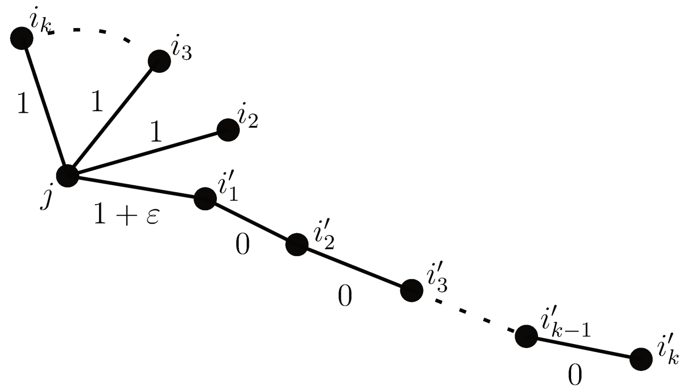

Figure 4.3: Assume that $\mathcal{C} = \{j\}$, $\mathcal{F}_1 = \{i_1\}$ and, for $1 < \ell \leq k$, $\mathcal{F}_\ell = \{i_\ell, i'_\ell\}$; define the distance function be the shortest path in the graph above. The shortest chain from client j would be $ji_1i'_2 \dots i'_k$ resulting in total cost of $1+\epsilon$, however, if we create one auxiliary instance per level, the solution to the ℓ-th instance would connect j to facility i_ℓ resulting, in the final solution, in the chain $ji_1i_2 \dots i_k$ of cost $2k - 1 + 2\epsilon$.

Proof. Analogously to the case where $k = 2$, fix an optimal solution Sol^* to $\mathcal{I}$ of cost $F^* + C^*$ and, for each $\ell \in \{1, 2, \dots, k\}$, let $\mathrm{Sol}^\dagger_\ell$ be a feasible solution to instance $\mathcal{I}_\ell$ of cost $\delta F^*_\ell + C^*_\ell$ created by opening the same ℓ-th level facilities opened in Sol^*.

Similarly to Theorem 4.2.2, define Sol_ℓ to be the solution obtained by the primal-dual SFLP algorithm for instance $\mathcal{I}_\ell$ with opening costs scaled by $\delta > 0$, $\ell \in \{1, 2, \dots, k\}$. We have that $3\delta F_{\mathrm{Sol}_\ell} + C_{\mathrm{Sol}_\ell} \leq 3\mathrm{OPT}_\ell \leq 3(\delta F^*_\ell + C^*_\ell)$.

Again, Lemma 4.2.1 guarantees that:

$$\mathrm{cost}(\mathrm{Sol}'_\ell) \leq \max\left(1 + \ln 3\delta, 2 - \frac{1}{3\delta}\right) F^*_\ell + \left(1 + \frac{2}{3\delta}\right) C^*_\ell,$$

where Sol'_ℓ is the solution to $\mathcal{I}_\ell$ after applying greedy augmentation to Sol_ℓ.

We know that $F^* = \sum_{\ell=1}^k F^*_\ell$. By triangle inequality, it must be the case that, for $\ell < k$, $C^*_\ell \leq 2C^*$ and $C^*_k \leq C^*$, thus $\sum_{\ell=1}^k C^*_\ell \leq (2k-1)C^*$.

Since, by triangle inequality, $\sum_{\ell=1}^k \mathrm{cost}(\mathrm{Sol}'_\ell)$ is an upper bound to the solution Sol obtained by the algorithm, we derive that:

$$\mathrm{cost}(\mathrm{Sol}) \leq \max\left(1 + \ln 3\delta, 2 - \frac{1}{3\delta}\right) F^* + (2k-1)\left(1 + \frac{2}{3\delta}\right) C^*.$$

Finally, by setting $\delta = e^{2k-2}/3$, we reach the desired result. $\square$

Note that the factor associated with C^* in the proof of Theorem 4.2.3 becomes $(2k-1)\left(1 + \frac{2}{3\delta}\right)$, implying that the approximation ratio of the algorithm must be at least $2k-1$. This is inherent to the strategy of creating one auxiliary instance per level and solving each separately, because, without knowing the optimal solution to $\mathcal{I}$, algorithms based in this strategy can be "tricked" into taking local decisions that are globally suboptimal, as illustrated in Figure 4.3.

Chapter 5

Concluding remarks

In this thesis, we reviewed approximation algorithms for some variations of the Facility Location Problem and presented new α-approximations for variations of the k-Level Stochastic FLP adapting the ones we had seen.

We started in Chapter 2 with a review of the $(3-o(1))$-approximation for the k-LFLP by Byrka and Rybicki [9]. Their algorithm solved a slightly modified LP for the k-LFLP that organized facilities in a forest of k-level trees, then proceeded to use an adaptation of the results by Garg, Konjevod and Ravi [13] for the Group Steiner Tree Problem to round the LP.

In Chapter 3, we reviewed the 1.86-approximation by Ye and Zhang [38] for the DFLP. Their algorithm adapted the 3-approximation by Jain and Vazirani [17] for the UFLP (see Section 1.2) by refining the way a maximal independent set of facilities is selected and, then, applying the greedy augmentation procedure that improves the result in the worst case [22, 10].

Finally, in Chapter 4, we demonstrated that, for the k-LSSCFLP, Byrka and Rybicki algorithm can be adapted to obtain $(4-o(1))$-approximation that has approximation factors 2.56, 2.78 and 2.95 for, respectively, $k = 2, 3$ and 4, improving the best known factor of 4 by Wang et al. [32] for the k-LSSCFLP and $3+\epsilon$ by Wu et al. [36] for the 2-LSSCFLP. Also, we applied Ye and Zhang's algorithm to obtain a 3.495-approximation for the 2-LSFLP that can be adapted to a $(2k-1+o(1))$-approximation for the k-LSFLP, which, to the best of our knowledge, is the first approximation algorithm for this problem.

In future research, new approximation techniques for stochastic problems might be attempted for the k-LSSCFLP to reduce its current gap in relation to the k-LFLP. Also, Byrka, Li and Rybicki' algorithm for the prize-collecting k-LFLP [8] might have an adaptation to the k-LSSCFLP that might obtain a better approximation factor. For the k-LSFLP, there might be untried techniques that reduce the approximation factor below the $2k-1$ threshold or that factor might be shown to be optimal.

Bibliography

[1] 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(5-6):161–167, 1999.

[2] A. A. Ageev, Y. Ye, and J. Zhang. Improved Combinatorial Approximation Algorithms for the k-Level Facility Location Problem. In *Proc. 30th International Colloquium on Automata, Languages and Programming, ICALP.*, pages 145–156, 2003.

[3] A.A. Ageev. Improved approximation algorithms for Multilevel Facility Location problems. *Operations Research Letters*, 30(5):327 – 332, 2002.

[4] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[5] A. Bumb and W. Kern. A simple dual ascent algorithm for the Multilevel Facility Location Problem. In Michel Goemans, Klaus Jansen, José D.P. Rolim, and Luca Trevisan, editors, *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, volume 2129 of *Lecture Notes in Computer Science*, pages 55–63. Springer, 2001.

[6] 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.

[7] J. Byrka, M. R. Ghodsi, and A. Srinivasan. LP-rounding algorithms for Facility-Location Problems. *Computing Research Repository*, abs/1007.3611, 2010.

[8] J. Byrka, S. Li, and B. Rybicki. Improved approximation algorithm for k-Level UFL with penalties, a simplistic view on randomizing the scaling parameter. In *International Workshop on Approximation and Online Algorithms*, pages 85–96. Springer, 2013.

[9] J. Byrka and B. Rybicki. Improved LP-rounding approximation algorithm for k-Level Uncapacitated Facility Location. In *International Colloquium on Automata, Languages, and Programming*, pages 157–169. Springer, 2012.

[10] M. Charikar and S. Guha. Improved combinatorial algorithms for the Facility Location and k-Median problems. In *Proc. 40th Annual Symposium on Foundations of Computer Science*, pages 378–388. IEEE, 1999.

[11] V. Chvátal. *Linear Programming*. Freeman, 1983.

[12] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. The Uncapacitated Facility Location Problem. Technical report, DTIC Document, 1983.

[13] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the Group Steiner Tree Problem. In Howard J. K., editor, *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 253–259. ACM/SIAM, 1998.

[14] S. Guha and S. Khuller. Greedy strikes back: improved facility location algorithms. *Journal of Algorithms*, 31(1):228–248, 1999.

[15] S. Guha, A. Meyerson, and K. Munagala. Hierarchical placement and network design problems. In *Proc. 41st Annual Symposium on Foundations of Computer Science*, pages 603–612, 2000.

[16] INFORMS. Section on Location Analysis. https://www.informs.org/Community/SOLA, 2015.

[17] K. Jain and V. Vazirani. Approximation algorithms for metric Facility Location and k-Median problems using the Primal-Dual schema and Lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296, 2001.

[18] L. G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

[19] R. Krishnaswamy and M. Sviridenko. Inapproximability of the Multi-level Uncapacitated Facility Location Problem. In *Proc. 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 718–734, 2012.

[20] J. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44(5):245–249, 1992.

[21] J. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraint violation (extended abstract). In S. Rao Kosaraju, M. Fellows, A. Wigderson, and J. A. Ellis, editors, *Proc. 24th Annual ACM Symposium on Theory of Computing*, pages 771–782. ACM, 1992.

[22] M. Mahdian, Y. Ye, and J. Zhang. Improved Approximation Algorithms for Metric Facility Location Problems. In *Proc. 5th International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 229–242, 2002.

[23] A. Meyerson, K. Munagala, and S. Plotkin. Cost-distance: two metric network design. In *Proc. 41st Annual Symposium on Foundations of Computer Science*, pages 624–630, 2000.

[24] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[25] R. Ravi and Amitabh Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In D. Bienstock and G. Nemhauser, editors, *Integer Programming and Combinatorial Optimization*, volume 3064 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 2004.

[26] D. B. Shmoys and C. Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *J. ACM*, 53(6):978–1012, 2006.

[27] D. B. Shmoys, E. Tardos, and K. Aardal. Approximation Algorithms for Facility Location Problems (Extended Abstract). In *Proc. 29^{th} Annual ACM Symposium on the Theory of Computing, May 4-6*, pages 265–274, 1997.

[28] R. J. Vanderbei. *Linear Programming: Foundations and Extensions.* Springer, 4^{th} edition, 2016.

[29] V. Vazirani. *Approximation algorithms.* Springer, 2003.

[30] J. Vygen. *Approximation algorithms facility location problems.* Forschungsinstitut für Diskrete Mathematik, Rheinische Friedrich-Wilhelms-Universität, 2005.

[31] Z. Wang, D. Du, A. F. Gabor, and D. Xu. An approximation algorithm for the k-level stochastic facility location problem. *Operations Research Letters*, 38(5):386–389, 2010.

[32] Z. Wang, D. Du, A. F. Gabor, and D. Xu. Erratum to: "An approximation algorithm for the k-level stochastic facility location problem" [Operations Research Letters 38(2010) 386-389]. *Operations Research Letters*, 39(2):160–161, 2011.

[33] Z. Wang, D. Du, and D. Xu. A Primal-Dual Approximation Algorithm for the k-Level Stochastic Facility Location Problem. In B. Chen, editor, *Algorithmic Aspects in Information and Management*, volume 6124 of *LNCS*, pages 253–260. Springer, 2010.

[34] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms.* Cambridge University Press, 2011.

[35] L. A. Wolsey and G. L. Nemhauser. *Integer and combinatorial optimization.* Wiley, 1990.

[36] C. Wu, D. Du, and D. Xu. Primal-dual approximation algorithm for the two-level facility location problem via a dual quasi-greedy approach. *Theoretical Computer Science*, 562:213 – 226, 2015.

[37] Y. Ye and J. Zhang. An Approximation Algorithm for the Dynamic Facility Location Problem. In M. Cheng, Y. Li, and D.-Z. Du, editors, *Combinatorial Optimization in Communication Networks*, volume 18 of *Combinatorial Optimization*, pages 623–637. Springer US, 2006.

[38] Y. Ye and J. Zhang. An approximation algorithm for the dynamic facility location problem. In *Combinatorial Optimization in Communication Networks*, pages 623–637. Springer, 2006.

[39] J. Zhang. Approximating the two-level facility location problem via a quasi-greedy approach. *Mathematical Programming*, 108(1):159–176, 2006.