AUT Journal of Mathematics and Computing

AUT J. Math. Comput., 3(2) (2022) 193-206
DOI: 10.22060/AJMC.2022.21392.1086

Review Article

Algorithmic approaches for network design with facility location: A survey

Mohsen Rezapour*[a]

[a]*Department of Computer Science and Statistics, Faculty of Mathematics, K. N. Toosi University of Technology, Tehran, Iran*

ABSTRACT: We consider a family of problems that combine network design and facility location. Such problems arise in many practical applications in different fields such as telecommunications, transportation networks, logistic, and energy supply networks. In facility location problems, we want to decide which facilities to open and how to assign clients to the open facilities so as to minimize the sum of the facility opening costs and client connection costs. These problems typically do not involve decisions concerning the routing of the clients' demands to the open facilities; once we decided on the set of open facilities, each client is served by the closest open facility. In network design problems, on the other hand, we generally want to design and dimension a minimum-cost routing network providing sufficient capacities to route all clients' demands to their destinations. These problems involve deciding on the routing of each client's demand. But, in contrast to facility location problems, demands' destinations are predetermined. In many modern day applications, however, all these decisions are interdependent and affect each other. Hence, they should be taken simultaneously. The aim of this work is to survey models, algorithmic approaches and methodologies concerning such combined network design facility location problems.

Review History:

Received:11 May 2022
Accepted:29 July 2022
Available Online:01 September 2022

Keywords:

Network design
Facility location
Approximation algorithm
Linear and integer programming

AMS Subject Classification (2010):

90-02; 90B10; 68M10

(Dedicated to Professor S. Mehdi Tashakkori Hashemi)

1. Introduction

A wide range of combinatorial optimization problems occur in the field of designing telecommunication networks. A typical telecommunication network, in its simplest form, consists of a backbone network with (almost) unlimited capacity on the links and several local access networks. In such a network, the traffic originating from the clients is sent through access networks to gateways or core nodes, which provide routing functionalities and access to the backbone network. The backbone then provides the connectivity among the core nodes, which is necessary to route the traffic further towards its destination. Designing such a network involves *locating* the core nodes, *connecting* them with each other, and *designing* a network by installing cables of different costs and capacities to route the traffic from the clients to the selected core nodes. As all these decisions are interdependent and affect each other, one has to integrate *facility location* decisions and *network design* decisions in order to cost-efficiently design such telecommunication networks. This has motivated several combined network design facility location problems. The aim of this paper is to survey models and algorithmic approaches concerning such combined network design facility location problems.

We start this introductory section with reviewing some of the most common approaches for coping with NP-hard combinatorial optimization problems. We then discuss well-studied variants and extensions of the facility location

**Corresponding author.*
E-mail addresses: mrezapour@kntu.ac.ir

and network design problems and review results related to these two very well-known problems.
We will survey models and algorithmic approaches concerning several interesting combined network design facility location problems in the upcoming sections. We shall discuss several related open problems as well.

1.1. Solution Approaches

There are many important optimization problems in practice that are difficult to solve optimally. In fact, many of those problems are known as **NP-hard** problems. Notice that no polynomial-time algorithm exists that solves any NP-hard problem optimally, assuming $\mathbf{P} \neq \mathbf{NP}$; we refer the reader to [35] for a thorough introduction to complexity theory. The most common approaches for coping with such problems are *approximation algorithms*, *heuristics*, and *Integer Programming*. We note that all the problems we consider in this survey are **NP-hard** and so the main focus of this work is on approximation techniques.

1.1.1. Approximation Algorithms

For some of the NP-hard problems, one may devise polynomial-time algorithms to solve them efficiently at the cost of providing solutions that are guaranteed to be only slightly sub-optimal. This leads to the notion of approximation algorithms.

We call algorithm A as an α-approximation algorithm for a minimization problem if A runs in polynomial time and returns a solution of cost no more than α times of the optimum. The value $\alpha > 1$ is known as the *approximation ratio* of the algorithm. There are several powerful techniques (e.g., *greedy procedure*, *primal-dual*, *dual-fitting*, *LP-rounding*, *sampling*) that can be used in the design of approximation algorithms; see [90, 94] for an introduction to these approximation techniques. The class **APX** is the set of NP problems that allow *constant-factor approximation algorithms* (or, more precisely, those that allow approximation algorithms with an approximation ratio bounded by a constant).

Of course not all the NP-hard problems allow constant-factor approximation algorithms. In fact, there are problems (e.g. *traveling salesman*, *set cover*) which are so hard that even finding constant-factor approximation for them can be shown to be NP-hard. This is where heuristics, for example, can come into play to solve such complex problems.

1.1.2. Heuristics

Algorithms that run in polynomial time and provide a solution which is good enough for instances at hand are called heuristics. We should remark that heuristics in contrast to approximation algorithms do not come always with a guarantee on the quality of their solution. In fact, heuristic algorithms often work well on most of the instances, but perhaps not on all of them. Heuristics can be classified into those which gradually build a feasible solution by a sequence of decisions, called *constructive algorithms* (e.g., greedy algorithm), and those which take a solution as input and produce a new improved solution by performing a sequence of operations, called *improvement algorithms* (e.g., local search algorithm). We refer the reader to [52] for an introduction to heuristic techniques.

One may still solve the NP-hard problems exactly, but of course not in a polynomial time. This leads us to the field of Integer Programming and many techniques there.

1.1.3. Integer Programming

Integer Programming (IP) is the natural way of modeling many real-world problems, including numerous NP-hard problems. Most of the techniques used to solve IPs are based on solving LP relaxations. In fact, this is because solving linear programs is much easier than solving integer programs. More precisely, solving integer programs is NP-hard (one can model some NP-hard problems as integer programs), whereas linear programs can be solved in polynomial time; see [61]. Integer linear programs are typically solved by using *Branch-and-Bound*, a widely known exact solution technique which creates a tree of nodes, called the *Branch-and-Bound tree*. The original problem is at the root node and subproblems are created by fixing variables. In fact, *Branch-and-Bound* handles integrality by branching this tree. The *cutting plane* method is another exact technique one can use to solve an IP. It works by iteratively solving the LP relaxation of the given IP which is gradually refined by adding more linear constraints called *cuts*. We refer the reader to [95] for an introduction to the subject. The *Branch-and-Bound* technique when used together with cutting plane methods is called *Branch-and-Cut*.

These approaches may be useful for problems of moderate size. However, there exist successful techniques (e.g. *column generation* [29, 70] and *Benders decomposition* [12]) that may be used to attack even very large scale problems by exploiting some specific structures of the problems. The Branch-and-Bound technique, when used together with column generation and cut separation is called *Branch-Cut-and-Price* [95, 70].

1.2. Facility Location and Variants

One of the most well-studied problems in the operations research and computer science literature is the *facility location* problem. In this problem, in its simplest form, we are given a set of clients and facilities, an opening cost associated with each facility, and a nonnegative distance between any pair of elements. The task is to open a subset of the facilities and assign each client to an open facility, such that the sum of opening costs and the distance between each client and the facility it is assigned to is minimized. This class of problems has a wide range of applications such as deciding placement of factories, warehouses, libraries, fire stations, hospitals, and base station for mobile phone service; see [91].

1.2.1. Uncapacitated Facility Location

The most widely studied model in discrete facility location is the *metric Uncapacitated Facility Location* (**UFL**) problem. In this problem, given are a finite set of locations V, potential facilities $F \subseteq V$ with opening costs $\mu_i \in \mathbb{Z}_{\geq 0}$, $i \in F$, clients $D \subseteq V$, and metric cost $l_{ij} \in \mathbb{Z}_{\geq 0}$, for assigning client $j \in D$ to facility $i \in F$. A solution to the problem consists of a set of open facilities $F^* \subseteq F$ and an assignment $\sigma^*(j): D \to F^*$. The aim is to find a solution that minimizes the total cost: $\min \sum_{i \in F^*} \mu_i + \sum_{j \in D} l_{\sigma^*(j)j}$.

The UFL problem is widely studied in the computer science literature. A greedy algorithm (similar to one for the set cover problem [28]) with $O(\log(n))$-approximation guarantee for the UFL problem was given in [51], where n is the number of clients. The first constant factor approximation algorithm for UFL was given in [83], and was based on LP rounding and a filtering technique due to [67]. Since then this factor has been gradually reduced to 1.488 [65] by a long series of papers (we point the reader to a survey by Vygen [91] for details). A number of elegant and powerful techniques have been used in the design of these approximation algorithms, e.g. *LP-rounding* [84, 25], *greedy procedure* [20, 41], *primal-dual* [57], and *dual-fitting* [55]. There are also results that combine the above techniques. For example, the authors of [56] presented a greedy algorithm that uses the LP-relaxation implicitly to obtain a lower bound for a primal-dual analysis; authors of [71] use Jain's algorithm [56] and the greedy procedure to get an approximation factor of 1.52; Byrka et al. [18] combine an LP-rounding based algorithm and Jain's algorithm [56] to obtain a 1.5-approximation algorithm; and finally Li [65] who combines the algorithm presented in [18] and Jain's algorithm [56] to achieve an approximation guarantee of 1.488.

On the hardness side, the authors of [41] showed (by a reduction from the set cover problem) that it is hard to approximate UFL within a factor of 1.463, assuming $\mathbf{NP} \not\subseteq \mathbf{DTIME}(n^{\log\log n})$. Later, this was generalized by the authors of [56] who show that the existence of a (λ_f, λ_c)-approximation algorithm with $\lambda_c < 1 + 2e^{-\lambda_f}$ implies $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\log\log n})$. An algorithm is a (λ_f, λ_c)-approximation algorithm if the solution the algorithm delivers has total cost at most $\lambda_f \cdot F^* + \lambda_c \cdot C^*$, where F^* and C^* are the facility and the assignment cost of an optimal solution, respectively.

1.2.2. Capacitated Facility Location

The *Capacitated Facility Location* problem (**CFL**) is one of the very well-studied variants of UFL. As in UFL, we are given a set of locations V, potential facilities $F \subseteq V$ with opening costs $\mu_i \in \mathbb{Z}_{\geq 0}$, $i \in F$, clients $D \subseteq V$, and metric cost $l_{ij} \in \mathbb{Z}_{\geq 0}$, for assigning client $j \in D$ to facility $i \in F$ and the problem asks us to open a subset of facilities and assign every client to an open facility. However, in addition to this, in CFL each facility has a capacity (upper bound) $U_i \in \mathbb{Z}_{>0}$, which limits the number of clients it can serve. The cost of a feasible solution is given as the sum of the facility opening costs and the assignment distances, as in UFL.

There are several approximation algorithms for CFL based on local search techniques. For the case of uniform capacities, Korupolu et al. [63] gave the first constant factor approximation algorithm, with ratio 8. This was later improved to 5.83 [26] and 3 [2]. The first constant factor approximation for the case of non-uniform capacities was proposed by Pal et al. [75] who gave an 9-approximation, which was eventually improved to 5 [8]. An LP-based approach to CFL was employed by Shmoys et al. [83] who gave the first bicriteria approximation for uniform capacities; this was later extended to non-uniform capacities in [1]. Levi et al. [64] obtained an LP-based true 5-approximation algorithm that only works when facilities opening costs are uniform. For a long time it was an open question to prove a constant factor approximation for CFL based on LP-rounding. This was recently solved by An et al. [4] who gave an LP-based 288-approximation algorithm for CFL which works for the general case.

1.2.3. Lower Bounded Facility Location

The *Lower Bounded Facility Location* problem (**LBFL**) is another interesting variant of UFL. As in UFL, we are given a set of locations V, potential facilities $F \subseteq V$ with opening costs $\mu_i \in \mathbb{Z}_{\geq 0}$, $i \in F$, clients $D \subseteq V$, and metric cost $l_{ij} \in \mathbb{Z}_{\geq 0}$, for assigning client $j \in D$ to facility $i \in F$ and the problem asks us to open a subset of facilities and assign every client to an open facility. However, in addition to this, in LBFL each facility has a lower bound

$L_i \in \mathbb{Z}_{\geq 0}$ on the number of clients it must serve if it is opened. The cost of a feasible solution is given as the sum of the facility opening costs and the assignment distances, as in UFL.

This problem was introduced independently by Guha et al. [42] and Karger et al. [60] who gave a bicriteria approximation. The first true approximation algorithm for LBFL was given by Svitkina [85] with ratio 448. The factor was then improved to 82.6 [3] by applying a modified variant of the algorithm of [85]. We note that the approaches of both papers work only if all lower bounds are uniform. A true approximation for LBFL when the lower bounds are non-uniform was given by Li [66] who gave a 4000-approximation algorithm for the problem with general facility lower bounds.

1.2.4. Lower and Upper Bounded Facility Location

The *Lower and Upper Bounded Facility Location* problem (**LUFL**) is a natural generalization of CFL and LBFL. We are given a complete graph $G = (V, E)$, with metric edge lengths $c_e \in \mathbb{Z}_{\geq 0}, e \in E$ containing a set of potential facilities $F \subseteq V$ and a set of demand points (clients) $D \subseteq V$. Each facility $i \in F$ has an opening cost $\mu_i \in \mathbb{Z}_{\geq 0}$ and a capacity (upper bound) $U_i \in \mathbb{Z}_{>0}$, which limits the amount of demand it can serve. Moreover, each facility i has a lower bound $L_i \in \mathbb{Z}_{\geq 0}$ on the amount of demand it must serve if it is opened.

A feasible solution to LUFL consists of a set of facilities $I \subseteq F$ to open, and a *valid* assignment $\sigma : D \to I$ of clients to the open facilities: an assignment is valid if it satisfies the lower and upper bounds

$$L_i \leq |\sigma^{-1}(i)| \leq U_i \qquad \forall i \in I.$$

The goal is to minimize the total cost, i.e., $\sum_{i\in I} \mu_i + \sum_{j\in D} c_{\sigma(j)j}$.

The first approximation algorithm for LUFL was given by the authors of [33] who gave a constant-factor bicriteria approximation algorithm for LUFL with uniform upper bounds and non-uniform lower bounds. Their algorithm violates both the upper and the lower bound by a constant factor. Gupta et al. [47] recently gave a constant factor approximation algorithm for LUFL that only violates the upper bound)without violating the lower bounds). However, their algorithm works only for the case when both the lower and the upper bounds are uniform.

1.3. Network Design and Variants

Network Design is one of the central topics in both computer science and operations research literature. The network design problems, in their simplest forms, only deal with building minimum-cost networks which satisfy a certain connectivity requirement between a set of terminals. This class of network design problems, known as *Connectivity*, has a large number of practical applications; e.g., in the design process of communication networks.

Another important class of network design problems arise, for example, in telecommunication networks where one has to design a network by installing cables of different costs and capacities to route traffic of a set of demand sources to a (multiple) sink(s); high-capacity cables are more expensive than low-capacity cables, while there are often economies of scale. We refer to this class of the problems as *Buy-at-Bulk Network Design* problems.

1.3.1. Connectivity

The most general version of the connectivity problems is called the *survivable network design* problem (**SND**). In this problem given are an undirected graph $G = (V, E)$, edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E$, a set of demand pairs $D \subseteq V \times V$, and an integer connectivity requirement $r_{uv} > 0$ for each pair of $(u, v) \in D$. A solution to the problem consists of an edge set $E^* \subseteq E$ containing r_{uv} edge-disjoint (u, v)-paths for each $(u, v) \in D$. The aim is to find a solution that minimizes the total cost: $\min \sum_{e\in E^*} c_e$. Note that if the paths are required to be vertex-disjoint, the problem is referred to as Vertex-Disjoint SND (**VD-SND**); and when all demand pairs D have a common vertex, say r, the problem is referred to as *rooted* SND (**rSND**).

These problems are well studied in the literature. The first non-trivial approximation algorithm for SND was given in [92, 93] where they gave a $2K$-approximation, where $K = max_{(u,v)\in D}\ r_{u,v}$. The factor was later improved to $2H_K$ [37], where $H_K = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{K}$. Finally a 2-approximation for this problem was obtained by Jain [54] who introduced the influential iterated rounding technique to design his algorithm. Then, this result has been generalized to the case of VD-SND when $r_{uv} \in \{0, 1, 2\}$ [32]. There are also several exact approaches proposed for this problem; see [82].

The *Steiner tree* problem (**ST**) is one of the most fundamental connectivity problems: Given are undirected graph $G = (V, E)$, edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E$, and a set of terminals $T \subseteq V$. A solution consists of a tree $S^* \subseteq E$ spanning terminals T. The aim is to find a solution that minimizes the total cost: $\min \sum_{e\in S^*} c_e$. Notice that ST can be viewed as the special case of the SND problem when $r_{uv} = 1$ iff $u, v \in T$.

The Steiner tree problem is **NP**-hard, even when edge costs are either 1 or 2; see [13]. The minimum cost terminal spanning tree on the fully connected graph of the metric closure containing only the terminals as vertices and the edges between them is well-known to be a 2-approximation for the Steiner tree problem [87]. More specifically, it is a $(2 - \frac{2}{|R|})$-approximation. This factor was later improved to $(1 + \frac{\ln 3}{2}) \approx 1.55$ in [80], and then to $(\ln 4 + \epsilon) \approx 1.39$ by Byrka et al. [19] who use the iterative rounding technique to obtain the currently best approximation ratio for the ST problem.

On the negative side, the authors of [24] show that there is no $(\frac{96}{95} - \epsilon)$-approximation algorithm for the Steiner tree problem, unless $\mathbf{P} = \mathbf{NP}$. Note that the same inapproximability result extends to the SND problem, too.

In some real world networks (e.g., telecommunications), to guarantee a desired level of quality of service, one has to pose a limit on the number of edges (hops) of the (routing) paths. This leads to an interesting variant of the SND problem, called the *Survivable Hop-Constrained Network Design* problem (**SHND**). In this problem given are an undirected graph $G = (V, E)$, edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E$, a set a of demand pairs $D \subseteq V \times V$, an integer connectivity requirement $r_{uv} > 0$ for each pair of $(u, v) \in D$, and an integer hop limit $H > 0$. A solution to the problem consists of an edge set $E^* \subseteq E$ containing r_{uv} edge-disjoint (u, v)-paths with at most H edges for each $(u, v) \in D$. The aim is to find a solution that minimizes the total cost: $\min \sum_{e \in E^*} c_e$

We call the rooted case when all demand pairs D have a common vertex as *rooted Survivable Hop-Constrained Network Design* (**rSHND**). Given a set of terminals $T \subseteq V$; we call a special case of the SHND problem where $r_{uv} = 1$ iff $u, v \in T$ as *Hop-Constrained Steiner Tree* (**HST**).

The HST problem is not in **APX**, even if the edge weights satisfy the triangle inequality [73]. Recall that **APX** is the set of NP optimization problems that allow constant-factor approximation algorithms. Note that this inapproximability result can be extended to SHND and rSHND, too.

The first IP formulation for SHND has been presented by the authors of [53] who only consider the case with $H \leq 4$ and $r_{uv} = 2$ for all (u, v) in D. Later, a more general (but rooted) version of this problem, with uniform connectivity requirement $K > 1$ and $H > 1$, has been considered in [17] where they present a branch-and-cut algorithm to solve the problem. The formulation given in [17] then has been strengthened by Mahjoub et al. [72] who presented an extended formulation for the rSHND by introducing additional variables which indicate the distance of each demand node to the root.

1.3.2. Buy-at-Bulk Network Design

The most general form of the buy-at-bulk problem is called *Non-uniform Buy-at-Bulk Network Design* problem (**Non-uniform BBND**), and is defined as follows. Given are an undirected graph $G = (V, E)$, edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E$, a set a of source-sink pairs $D \subseteq V \times V$ with demands $d_{(u,v)} \in \mathbb{Z}_{>0}$, $(u, v) \in D$, and a sub-additive monotone function $f_e : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ which gives the cost (per unit length) of transporting demand along edge e. A solution to the problem consists of en edges set $E^* \subseteq E$ such that, all pairs (u, v) are connected in $G[E^*]$. The aim is to find a solution that minimizes the total cost: $\min \sum_{e \in E^*} f_e(\hat{x}_e) \cdot c_e$, where $\hat{x}_e$ denotes the total units of demand routed along edge e. We refer to the problem as *single-sink* case when all source-sink pairs share the same sink terminal. When the sink terminals can be any vertices in the graph, we refer to the problem as *multi-sink* case. We call the case when $f_e = f$ for all $e \in E$ as the *uniform* case.

Buy-at-bulk network design problems have been considered in both operations research and computer science literature. The first non-trivial approximation algorithm to the buy-at-bulk network design problem was given in [21] where the authors obtained an approximation ratio of $e^{O(\sqrt{\log |D| \log \log |D|})} \cdot \log \bar{d}$, where $\bar{d} = \sum_{(u,v) \in D} d_{(u,v)}$. They also obtained a $O(\log^2 |D|)$-approximation for the single-sink case. Their algorithm is a simple randomized greedy algorithm based on shortest-path approach. The first poly-logarithmic approximation for non-uniform BBND was given in [22]. The authors obtained an approximation ration of $O(\min\{\log^3 |D| \cdot \log \bar{d}, \log^5 |D| \log \log |D|\})$, which was then improved to $O(\log^3 |D|)$ for the case when demand values can be polynomially bounded with respect to $|D|$ [62].

For the uniform case, a randomized $O(\log^2 n)$-approximation was obtained in [7], where n is the number of vertices in the graph. Their algorithm is based on the tree-embeddings [10]. Thus the approximation ratio naturally can be improved to $O(\log n \log \log n)$ and then to $O(\log n)$ using the improved results on approximation of metrics by trees; see [11, 31].

Regarding the single-sink case, the first result is an $O(\log |D|)$ randomized approximation algorithm due to [74]. They referred to the problem as *Cost-Distance*. The algorithm of [74] was then derandomized by the authors of [23] who use an LP rounding approach, establishing an integrality gap of $O(\log |D|)$.

On the hardness side, Andrew [5] obtained a hardness result of $\Omega(\log^{\frac{1}{2}} n)$ and $\Omega(\log^{\frac{1}{4}} n)$ for the non-uniform and uniform cases, respectively. Moreover, a hardness result of $\Omega(\log \log n)$ is obtained in [27] for the single-sink case.

The uniform single-sink case of the BBND problem under a *cable capacity* cost model is called *Single-Sink Buy-at-Bulk Network Design* (**SSBB**). In this problem given are an undirected graph $G = (V, E)$, edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E$, a set a of demands $D \subseteq V$ with demands $d_j \in \mathbb{Z}_{>0}$, $j \in D$, a sink vertex $t \in V$, a set of cable types K with capacity $u_k \in \mathbb{Z}_{>0}$, $k \in K$ & setup cost (per unit length) such that $\sigma_k \in \mathbb{Z}_{\geq 0}$, $k \in K$ $\sigma_1 < ... < \sigma_K$ & $\frac{\sigma_1}{u_1} > ... > \frac{\sigma_K}{u_K}$. A solution to the problem consists of an edges set $E^* \subseteq E$ with an cable installation $\alpha : E^* \times K \to \mathbb{Z}_{\geq 0}$ such that, all demands in D can be sent to t via the resulting capacitated network. The aim is to find a solution that minimizes the total cost: $\min \sum_{e \in E^*} \sum_{l \in K} \sigma_k c_e \alpha_{e,l}$. We refer to the problem as *splittable* when the demand of each client is allowed to be routed along several paths. When the entire demand of each client must be routed along a single path, we refer to the problem as *unsplittable*.

Several approximation algorithms for this problem have been proposed in the computer science literature. For the unsplittable case, an $O(K)$ approximation, using LP rounding techniques, has been developed in [36]. The first constant factor approximation for this problem is due to [43, 44]. The authors of [88] showed that an LP formulation of this problem has a constant integrality gap and provided a 216 approximation algorithm. Using sampling techniques, this factor was reduced to 145.6 [58], and later to 40.82 [39].

For the splittable case, Gupta et al. [46] presented a simple 76.8-approximation algorithm using random-sampling techniques. Unlike the algorithms mentioned above, their algorithm does not guarantee that the solution is a tree. Modifying Gupta's algorithm, the approximation for the splittable case was later reduced to 65.49 [58], and then to 24.92 [39].

On the negative side, a 1.278-inapproximability bound for SSBB may be obtained trivially from inapproximability of the Single-Sink Rent-or-Buy problem [40].

The SSBB problem is also well studied in the operation research literature. However, in the operation research literature, it is mostly known as *single-source network loading problem* (e.g. [69]), or (in the case of telecommunication network planning) as *Local Access Network Design Problem (LAN)* (e.g. [81]). The LAN problem with only two cable types under the assumption that the solution must be a tree (with unsplittable flow) was considered in [77]. The authors provide a multicommodity flow based formulation for the problem and solve it by applying Benders' decomposition. The LAN problem with multiple cable types was then considered in [81]. They apply flow-based MIP formulations and work with relaxations obtained by approximating the capacity step cost function by its lower convex envelope to provide a special branch-and-bound algorithm for LAN design. Their technique was later reformulated as a stylized branch-and-bound algorithm [76]. Finally, a stronger multicommodity flow formulation for the problem was considered by the authors of [69] who applied a branch-and-cut algorithm based on Benders decomposition for solving the problem.

2. Facility Location with Connectivity

In this section we survey problems that integrate connectivity into the classical facility location problem. We also discuss some interesting open problems related to these problems.

2.1. Facility Location with 'Simple' Connectivity

An interesting variant of UFL occurs in communication networks (in particular in distribution networks in telecommunications) where facilities want to communicate with each other, and hence a connectivity among facilities (via high bandwidth links) is required. In a distributed network, for example, the facilities represent servers which need to be able to communicate with each other in order to ensure consistency of data. This leades to a variant of UFL that is called *Connected Facility Location* (**ConFL**). In this problem given are an undirected graph $G = (V, E)$, metric edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E$, potential facilities $F \subseteq V$ with opening costs $\mu_i \in \mathbb{Z}_{\geq 0}$, $i \in F$, clients $D \subseteq V$ with demands $d_j \in \mathbb{Z}_{>0}$, $j \in D$, core cable type with infinite capacity and setup cost (per unit length) $M > 1$. A solution to this problem consists of a subset of open facilities $F^* \subseteq F$, an Steiner tree $T^* \subseteq E$ of core cables spanning F^*, and an assignment $\sigma^*(j) : D \to F^*$. The aim is to find a solution that minimizes the total cost: $\min \sum_{i \in F^*} \mu_i + \sum_{e \in T^*} M c_e + \sum_{j \in D} d_j \cdot l(j, \sigma^*(j))$; where $l(u, v)$ is the shortest path distance between vertices u and v in G.

Several approximation algorithms for this problem have been proposed in the computer science literature. A 10.66-approximation for this problem, based on LP rounding, has been proposed in [45]. This later was improved in [86] where the authors obtained an approximation ratio of 8.55, using a primal-dual algorithm. Then, using LP rounding techniques, the approximation factor was improved to 8.29 in the general case and to 7 in case all opening costs are equal in [48]. A 6.55-approximation primal-dual algorithm for the ConFL problem was proposed in [59]. Finally, using *sampling techniques*, the guarantee was reduced to 4 in [30], and to 3.19 in [40].

On the hardness side, the results by [41] for the facility location problem can be adapted to prove that ConFL is hard to approximate within 1.463 (unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\log \log n})$), as observed in [40].

The ConFL problem is also widely studied in the operation research community. The first heuristic algorithm for the ConFL problem was given in [68]. A greedy randomized adaptive search with a multi-start iterative construction was also proposed in [89]. Later, the ConFL problem was formulated as a directed Steiner tree problem with a unit degree constraint in [9] where the authors proposed a dual-based heuristic for the problem. It is worth noting that their dual-based algorithm is able to provide both upper and lower bounds (by returning a primal feasible solution together with a dual feasible solution) for a given instance. This can be used to assess the quality of the solutions. We refer the reader to the paper by Gollowitzer et al. [38] for an overview of formulations and exact approaches for ConFL.

A special case of the ConFL problem where all opening costs are 0 and facilities may be opened anywhere ($F = V$) is called the *Single-Sink Rent-or-Buy* problem (**SSRoB**). The SSRoB problem is well-studied in the literature; see [60, 45, 86, 46, 30]. The approximation algorithms proposed for the ConFL problem obviously work for SSRoB too, however some of them may come with improvements in their approximation guarantees. For example, the algorithms in [45], [86], and [30] have improved approximation ratios of 9.002, 4.55, and 2.92, respectively, for this special case. On the hardness side, Grandoni et al. [40] obtained a 1.278-inapproximability bound for SSRoB.

Facilities in real-world applications are usually capacitated. There is still no true constant factor approximation algorithm for the capacitated version of the ConFL problem in the literature, so the obvious open problem is to get a constant factor approximation algorithm for this very practically important variant of the ConFL problem.

2.2. Facility Location with 'Complex' Connectivity

A typical metropolitan telecommunication network consists of several local access networks, that are connected by a (regional) core network to a central hub node, that provides connectivity to the national or international backbone. The traffic originating at the clients is sent through the access networks to the (regional) core nodes. From there, it traverses the core network(s) to reach the national core or the access network of its destination. Routing functionalities are typically only available at the regional or central core nodes. Hence, the core networks usually play a vital role for the service availability and the service quality in such networks. To guarantee the service availability, it is common to increase the number of reserved edge-disjoint (hop-limited) routing paths between each pair of core nodes. Also, routing paths with hop-length constraints can guarantee a required level of quality of service; as long routing paths may lead to unacceptable delays in the network. This leads to a complex version of the connected facility location problem, called the *Survivable Hop Constrained Connected Facility Location* problem (**SHConFL**) [15, 79], which can be used to model telecommunication networks that require both survivability and hop-length constraints; e.g., [49, 50].

In SHConFL, given are an undirected graph $G = (D\dot{\cup}S, E)$ containing clients D and core nodes S, core edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E_S$, where $E_S := \{uv \in E : u, v \in S\}$, potential facilities $F \subseteq S$ with opening costs $\mu_i \in \mathbb{Z}_{\geq 0}$, $i \in F$, root $r \in S \setminus F$, assignment costs $a_{ij} \in \mathbb{Z}_{\geq 0}$ for assigning client j to facility i, a hop limit $H \geq 1$, and a connectivity requirement $\lambda \geq 1$. The task is to open a subset of facilities $I^* \subseteq F$, assign the clients to the open facilities $\sigma^*(j) : D \to I \cup \{r\}$, and select an edge set $E^* \subseteq E_S$ containing λ edge-disjoint H-bounded paths between r and each facility $i \in I^*$ interconnect the open facilities in such a way, that the resulting network (core network) contains at least λ edge-disjoint paths, each containing at most H edges, between the root and each open facility. The objective is to minimize the total cost for opening facilities, assigning clients to open facilities, and installing core connections: $\sum_{i\in I^*} \mu_i + \sum_{j\in D} a_{\sigma^*(j)j} + \sum_{e\in E^*} c_e$.

SHConFL was introduced in the work by Bley et al. [15]. It is shown that there is no approximation algorithm for SHConFL, even with the edge weights satisfying the triangle inequality, which guarantees a worst case approximation ratio better than $\Theta(log(|V|))$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\log\log n})$; see [79].

Bley et al. [15] undertook the first computational study for the SHConFL problem. They proposed two strong extended formulations for the problem and devised a practically efficient branch-and-cut algorithm based on Benders decomposition for finding the solution. They also provided a theoretical comparison between the models they proposed and suggested some heuristic ideas to speed up the algorithm; see [15, 79] for details.

To the best of our knowledge there is still no approximation algorithms for SHConFL. Therefor, devising a non-trivial approximation algorithm for SHConFL remains an interesting open task.

3. Facility Location with Buy-at-Bulk Network Design

In this section we survey problems that integrate buy-at-bulk network design into the classical facility location problem. We also discuss some interesting open problems related to these problems.

3.1. Multifacility Buy-at-Bulk Network Design

In the facility location problem, the aim is to decide which facilities to open and how to assign clients to these open facilities so that the sum of the facility opening costs and client connection costs is minimized. In the (single-sink) buy-at-bulk network design problem on the other hand, the aim is to design a minimum cost routing network providing sufficient capacities to route all clients' demands to their sink. In many modern day applications, particularly in the planning of telecommunication networks, however, all these decisions are interdependent and affect each other and hence they should be taken simultaneously. In the planning of telecommunication networks, for example, this corresponds to locating routing and switching devices (facilities) and dimensioning access cables that are used to route the traffic from clients to facilities. Such a combined network design facility location problem can be formulalted as the *Multifacility Buy-at-Bulk Network Design* problem (**MFBB**) [6, 34, 79].

In MFBB, we are given a complete graph $G=(V,E)$ with nonnegative edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E$ satisfying triangle inequality; a set $F \subseteq V$ of facilities with opening costs $\mu_i \in \mathbb{Z}_{\geq 0}$, $i \in F$; and a set of clients $D \subseteq V$ with demands $d_j \in \mathbb{Z}_{>0}$, $j \in D$. We are also given K *types* of *access cables* that may be used to connect clients to open facilities. A cable of type i has capacity $u_i \in \mathbb{Z}_{>0}$ and cost (per unit length) $\sigma_i \in \mathbb{Z}_{\geq 0}$. We assume that access cable types obey economies of scale. That is, $\sigma_1 < \sigma_2 < \cdots < \sigma_K$ and $\frac{\sigma_1}{u_1} > \frac{\sigma_2}{u_2} > \cdots > \frac{\sigma_K}{u_K}$. A feasible solution consists of (1) a subset $F_0 \subseteq F$ of facilities to open; (2) a forest (access network) $A^* \subseteq E$ such that, for each client $j \in D$, A^* contains exactly one path P_j from j to some open facility $i_j \in F^*$. Furthermore, on each edge of this forest we have to specify a list of possibly multiple copies and types of access cables to install, in such a way that the entire demand of each client can be routed along a single path to an open facility: an access cable installation $x : A^* \times K \to \mathbb{Z}_{\geq 0}$ of sufficient capacity, i.e., $\sum_{j:\ e \in P_j} d_j \leq \sum_k u_k x_{e,k}$. The objective of MFBB is to minimize the total cost of opening facilities and access networks: $\min \sum_{i \in F^*} \mu_i + \sum_{e \in A^*} \sum_{k \in K} \sigma_k c_e x_{e,k}$.

Such a problem also has applications in transportation logistics [78], where one has to locate manufacturing facilities and select trucks of different capacities shipping goods to the clients so that the entire demand of each client is shipped by the same truck (unsplittable).

The MFBB problem has been considered for the first time in [74]. They show that the problem can be seen as a special case of the *Cost-Distance* problem, and thereby provide the first $\mathrm{O}(\log(|D|))$ approximating algorithm for MFBB. Ravi et al. [78] later developed an $O(K)$ approximation for this problem and called it *Integrated logistics*.

Arulselvan et al. [6] undertook the first computational study for the BBFL problem. They provided the integer programming formulations both compact and exponential-sized for the problem. In particular, they modeled the problem as a path-based formulation and developed a branch-cut-and-price algorithm for finding the solution. They also studied several classes of valid inequalities and presented different types of primal heuristics; see [6, 79] for details.

There is still no $O(1)$ approximation for the MFBB problem in the literature, so the obvious open problem is to get a constant factor approximation algorithm for MFBB.

Facilities in real-world applications are usually capacitated and this is not considered in the model introduced by Arulselvan et al. [6]. It would be practically interesting to extend their models and algorithmic approaches to the variant with capacitated facilities.

3.2. Deep-Discount Facility Location

Problems similar to MFBB also arise in the planning of water and energy supply networks or transportation networks. In some of those applications, however, the consideration of different connection types on the edges of the access network is not motivated by the different capacities but by the different per unit shipping cost of alternative technologies or operational modes, while the maximum capacity is seemingly unlimited. In transportation logistics, for example, the per unit shipping cost on a connection typically is strongly dependent on the chosen transportation mode, while the maximum capacity is (seemingly) unlimited. This naturally leads to another interesting combined facility location network design problem where each cable type, instead of having a fixed cost and a fixed capacity, has unlimited capacity but a traffic-dependent variable cost in addition to its fixed cost. This version of the problem is called the *Deep-Discount Facility Location* problem (**DDFL**) [79]. More precisely, in the DDFL version, an access cable of type k has a fixed *setup cost* (per unit length) $\sigma_k \in \mathbb{Z}_{\geq 0}$ and a flow dependent *incremental cost* (per unit length and per flow unit) of $r_k \in \mathbb{Z}_{\geq 0}$.

Friggstad et al. [34] proved an upper bound of $O(K)$ on the integrality gap of a natural flow-based linear programming formulation of the DDFL problem. They also observed that one can transform between DDFL and MFBB with factor 2 loss, implying that an ρ-approximation to DDFL gives a 2ρ-approximation to MBBB and vice versa; see [79] for details.

Proving an upper bound of $O(1)$ on the integrality gap of DDFL remains an interesting open problem. A potentially easier problem is to get an α-approximation for DDFL with running time $n^{f(k)}$ for some function f where α is a constant that does not depend on k.

4. Facility Location with Buy-at-Bulk Connectivity

In this section we survey problems that integrate both connectivity and buy-at-bulk network design into the classical facility location problem. We also discuss some interesting open problems related to these problems.

4.1. Buy-at-Bulk Connected Facility Location

In the planning of point-to-point optical access networks an operator must decide on which nodes (locations) the routing and switching devices (these are called central offices) should be installed; how to route the traffic originating from clients to central offices via tree-like access networks of cables; and how to inter-connect central offices via a high bandwidth core network. A combination of different cable types may be installed on the edges of access trees to support the traffic flow. This allows for multiple fibers emanating from different clients to share a single, larger cable and the same trunk on their common path towards their common central office. The central offices are connected amongst each other or to some higher network level via a core network which is required to route the traffic further towards its destination. Designing such a network involves selecting the facilities, connecting them via high bandwidth links, and dimensioning the access links that are used to route the traffic from the clients to facilities. This can be modeled as a Buy-at-Bulk Connected Facility Location problem (**BBConFL**) [34, 14, 79].

In BBConFL, we are given a complete graph $G = (V, E)$ with nonnegative edge lengths $c_e \in \mathbb{Z}_{\geq 0}$, $e \in E$ satisfying triangle inequality; a set $F \subseteq V$ of facilities with opening costs $\mu_i \in \mathbb{Z}_{\geq 0}$, $i \in F$; and a set of clients $D \subseteq V$ with demands $d_j \in \mathbb{Z}_{>0}$, $j \in D$. We are also given K *types* of *access cables* that may be used to connect clients to open facilities. A cable of type i has capacity $u_i \in \mathbb{Z}_{>0}$ and cost (per unit length) $\sigma_i \in \mathbb{Z}_{\geq 0}$. Furthermore, we are given an extra type of cable, called *core cable*, having a cost (per unit length) of $M > \sigma_K$ and infinite capacity, which may be used to connect the open facilities with each other. We assume that access cable types obey economies of scale. That is, $\sigma_1 < \sigma_2 < \cdots < \sigma_K$ and $\frac{\sigma_1}{u_1} > \frac{\sigma_2}{u_2} > \cdots > \frac{\sigma_K}{u_K}$. A feasible solution for BBConFL consists of (1) A subset $F_0 \subseteq F$ of facilities to open; (2) a Steiner tree of G (core network) connecting all open facilities via core cables; and (3) a forest (access network) connecting all clients to the open facilities. Furthermore, on each edge of this forest we have to specify a list of possibly multiple copies and types of access cables to install, in such a way that the entire demand of each client can be routed along a single path to an open facility. Note that we allow the demand crossing a single edge to use different access cables, but the collection of edges trasversed must be a path in G. The objective of BBConFL is to minimize the total cost of opening facilities, and constructing core and access networks; where the cost for using edge e in the core network is Mc_e, and the cost for installing a single copy of access cable of type i on an edge e is $\sigma_i c_e$.

It is worth noting that we are allowed to install core cables on edges incident to closed facilities, to clients, or even to nodes in $V \setminus (F \cup D)$. Nevertheless, the demand from a client to its facility *is not allowed to* use core cables. The rationality for this constraint is that in real-life situations core and access networks are run independently. The only way to access from the access network to the core network is via an open facility.

The BBConFL was introduced by in the work by Bley et al. [16]. They developed the first constant factor approximation algorithms for the BBConFL problem based on the random sampling techniques, achieving a 192-approximation for BBConFL. Later, Friggstad et al. [34] devised the first LP-based approximation algorithm for the problem and proved an integrality gap bound of $O(1)$.

Similar to that for MFBB there are various interesting variants of BBConFL that differ with respect to the structure of the access or core network, called the *deep-discount edge costs* problem (**DDConFL**). In this problem, instead of capacitated access cables, we are given K *discount cable types*, where cable type i has a fixed cost (setup cost) of σ_i, a flow dependent incremental cost of δ_i, and unbounded capacity. We assume that $\delta_1 > \delta_2 > \cdots > \delta_k$ (i.e discount cables obey economies of scale). The cost for installing one copy of discount type i on edge e and transporting R flow units on e is $(\sigma_i + R\delta_i)c_e$.

Bley et al. [16] observed that BBConFL and DDConFL are very closely related and that a ρ-approximation algorithm for one problem carries over to a 2ρ-approximation algorithm for the other. They obtained a polynomial time 384-approximation algorithm for the DDConFL problem which was later improved to 234-approximation in [34].

The factors proven in [16, 34] are too large to be of practical interest. Improving the current ratios to values relevant for applications, either via new algorithmic concepts or via better analytical tools, remains an interesting open problem.

It is also an interesting open question whether their algorithmic approach can be extended to obtain approximation algorithms also for the case with capacitated facilities.

References

[1] Z. Abrams, A. Meyerson, K. Munagala, and S. Plotkin, *The integrality gap of capacitated facility location*, Technical Report CMU-CS-02-199, Carnegie Mellon University, (2002).

[2] A. Aggarwal, A. Louis, M. Bansal, N. Garg, N. Gupta, S. Gupta, and S. Jain, *A 3-approximation algorithm for the facility location problem with uniform capacities*, Mathematical Programming, 141 (2013).

[3] S. Ahmadian and C. Swamy, *Improved approximation guarantees for lower-bounded facility location*, Approximation and Online Algorithms, (2012), pp. 257–271.

[4] H.-C. An, M. Singh, and O. Svensson, *LP-based algorithms for capacitated facility location*, In proceedings of FOCS 2014, (2014), pp. 256–265.

[5] M. Andrews, *Hardness of buy-at-bulk network design*, Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on, (2004), pp. 115–124.

[6] A. Arulselvan, M. Rezapour, and W. A. Welz, *Exact approaches for designing multifacility buy-at-bulk networks*, INFORMS Journal on Computing, 29 (2017), pp. 597–611.

[7] B. Awerbuch and Y. Azar, *Buy-at-bulk network design*, Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on, (1997), pp. 542–547.

[8] M. Bansal, N. Garg, and N. Gupta, *A 5-approximation for capacitated facility location*, In proceedings of ESA 2012, (2012), pp. 133–144.

[9] M. G. Bardossy and S. Raghavan, *Dual-based local search for the connected facility location and related problems*, INFORMS Journal on Computing, 22 (2010), pp. 584–602.

[10] Y. Bartal, *Probabilistic approximation of metric spaces and its algorithmic applications*, Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on, (1996), pp. 184–193.

[11] ——, *On approximating arbitrary metrices by tree metrics*, Proceedings of the thirtieth annual ACM symposium on Theory of computing, (1998), pp. 161–168.

[12] J. F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, Numerische mathematik, 4 (1962), pp. 238–252.

[13] M. Bern and P. Plassmann, *The steiner problem with edge lengths 1 and 2*, Information Processing Letters, 32 (1989), pp. 171–176.

[14] A. Bley, S. M. Hashemi, and M. Rezapour, *Approximation algorithms for a combined facility location buy-at-bulk network design problem*, Theory and Applications of Models of Computation (TAMC), Lecture Notes in Computer Science Vol. 7876, (2013), pp. 72–83.

[15] ——, *Ip modeling of the survivable hop constrained connected facility location problem*, Electronic Notes in Discrete Mathematics, 41 (2013), pp. 463–470.

[16] A. Bley and M. Rezapour, *Combinatorial approximation algorithms for buy-at-bulk connected facility location problems*, Discrete Applied Mathematics, 213 (2016), pp. 34–46.

[17] Q. Botton, B. Fortz, L. Gouveia, and M. Poss, *Benders decomposition for the hop-constrained survivable network design problem*, INFORMS journal on computing, 25 (2013), pp. 13–26.

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

[19] J. Byrka, F. Grandoni, T. Rothvoss, and L. Sanità, *An improved lp-based approximation for steiner tree*, Proceedings of the 42nd ACM symposium on Theory of computing, (2010), pp. 583–592.

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

[21] M. Charikar and A. Karagiozova, *On non-uniform multicommodity buy-at-bulk network design*, Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, (2005), pp. 176–182.

[22] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour, *Approximation algorithms for nonuniform buy-at-bulk network design*, SIAM Journal on Computing, 39 (2010), pp. 1772–1798.

[23] C. Chekuri, S. Khanna, and J. S. Naor, *A deterministic algorithm for the cost-distance problem*, Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms, (2001), pp. 232–233.

[24] M. Chlebík and J. Chlebíková, *The steiner tree problem on graphs: Inapproximability results*, Theoretical Computer Science, 406 (2008), pp. 207–214.

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

[26] F. A. Chudak and D. P. Williamson, *Improved approximation algorithms for capacitated facility location problems*, Mathematical programming, 102 (2005).

[27] J. Chuzhoy, A. Gupta, J. S. Naor, and A. Sinha, *On the approximability of some network design problems*, ACM Transactions on Algorithms (TALG), 4 (2008), p. 23.

[28] V. Chvatal, *A greedy heuristic for the set-covering problem*, Mathematics of operations research, 4 (1979), pp. 233–235.

[29] G. B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, Operations research, 8 (1960), pp. 101–111.

[30] F. Eisenbrand, F. Grandoni, T. Rothvoss, and G. Schäfer, *Connected facility location via random facility sampling and core detouring*, Journal of Computer and System Sciences, 76 (2010), pp. 709–726.

[31] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, (2003), pp. 448–455.

[32] L. Fleischer, K. Jain, and D. P. Williamson, *Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems*, Journal of Computer and System Sciences, 72 (2006), pp. 838–867.

[33] Z. Friggstad, M. Rezapour, and M. R. Salavatipour, *Approximating connected facility location with lower and upper bounds via lp rounding*, 15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016), (2016).

[34] Z. Friggstad, M. Rezapour, M. R. Salavatipour, and J. A. Soto, *Lp-based approximation algorithms for facility location in buy-at-bulk network design*, Algorithmica, 81 (2019), pp. 1075–1095.

[35] M. Garey and D. Johnson, *Computers and intractability: a guide to the theory of np-completeness*, WH Freeman & Co., San Francisco, (1979).

[36] N. Garg, R. Khandekar, G. Konjevod, R. Ravi, F. S. Salman, and A. Sinha, *On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design problem*, Integer Programming and Combinatorial Optimization, (2001), pp. 170–184.

[37] M. X. Goemans, A. V. Goldberg, S. A. Plotkin, D. B. Shmoys, E. Tardos, and D. P. Williamson, *Improved approximation algorithms for network design problems.*, SODA, 94 (1994), pp. 223–232.

[38] S. Gollowitzer and I. Ljubić, *Mip models for connected facility location: A theoretical and computational study*, Computers & Operations Research, 38 (2011), pp. 435–449.

[39] F. Grandoni and T. Rothvoss, *Network design via core detouring for problems without a core*, Automata, Languages and Programming, (2010), pp. 490–502.

[40] ——, *Approximation algorithms for single and multi-commodity connected facility location*, Integer Programming and Combinatoral Optimization, (2011), pp. 248–260.

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

[42] S. Guha, A. Meyerson, and K. Munagala, *Hierarchical placement and network design problems*, Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on, (2000), pp. 603–612.

[43] ——, *A constant factor approximation for the single sink edge installation problems*, Proceedings of the thirty-third annual ACM symposium on Theory of computing, (2001), pp. 383–388.

[44] ——, *A constant factor approximation for the single sink edge installation problem*, SIAM Journal on Computing, 38 (2009), pp. 2426–2442.

[45] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, *Provisioning a virtual private network: a network design problem for multicommodity flow*, Proceedings of the thirty-third annual ACM symposium on Theory of computing, (2001), pp. 389–398.

[46] A. Gupta, A. Kumar, and T. Roughgarden, *Simpler and better approximation algorithms for network design*, Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, (2003), pp. 365–372.

[47] N. Gupta, S. Grover, and R. Dabas, *Respecting lower bounds in uniform lower and upper bounded facility location problem*, International Computing and Combinatorics Conference, (2021), pp. 463–475.

[48] M. K. Hasan, H. Jung, and K.-Y. Chwa, *Approximation algorithms for connected facility location problems*, Journal of Combinatorial Optimization, 16 (2008), pp. 155–172.

[49] S. M. Hashemi, A. Moradi, and M. Rezapour, *An aco algorithm to design umts access network using divided and conquer technique*, Engineering Applications of Artificial Intelligence, 21 (2008), pp. 931–940.

[50] S. M. Hashemi, M. Rezapour, and A. Moradi, *An effective hybrid pso-based algorithm for planning umts terrestrial access networks*, Engineering Optimization, 42 (2010), pp. 241–251.

[51] D. S. Hochbaum, *Heuristics for the fixed cost median problem*, Mathematical programming, 22 (1982), pp. 148–162.

[52] J. Hromkovic, *Algorithmics for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics*, Springer-Verlag, (2010).

[53] D. Huygens, M. Labbé, A. R. Mahjoub, and P. Pesneau, *The two-edge connected hop-constrained network design problem: Valid inequalities and branch-and-cut*, Networks, 49 (2007), pp. 116–133.

[54] K. Jain, *A factor 2 approximation algorithm for the generalized steiner network problem*, Combinatorica, 21 (2001), pp. 39–60.

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

[56] K. Jain, M. Mahdian, and A. Saberi, *A new greedy approach for facility location problems*, Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, (2002), pp. 731–740.

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

[58] R. Jothi and B. Raghavachari, *Improved approximation algorithms for the single-sink buy-at-bulk network design problems*, Algorithm Theory-SWAT 2004, (2004), pp. 336–348.

[59] H. Jung, M. K. Hasan, and K.-Y. Chwa, *A 6.55 factor primal-dual approximation algorithm for the connected facility location problem*, Journal of combinatorial optimization, 18 (2009), pp. 258–271.

[60] D. Karget and M. Minkoff, *Building steiner trees with incomplete global knowledge*, Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on, (2000), pp. 613–623.

[61] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Proceedings of the sixteenth annual ACM symposium on Theory of computing, (1984), pp. 302–311.

[62] G. Kortsarz and Z. Nutov, *Approximating some network design problems with node costs*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, (2009), pp. 231–243.

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

[64] R. Levi, D. B. Shmoys, and C. Swamy, *LP-based approximation algorithms for capacitated facility location*, Mathematical programming, 131 (2012).

[65] S. Li, *A 1.488 approximation algorithm for the uncapacitated facility location problem*, Information and Computation, 222 (2013), pp. 45–58.

[66] ——, *On facility location with general lower bounds*, Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, (2019), pp. 2279–2290.

[67] J.-H. Lin and J. S. Vitter, *Approximation algorithms for geometric median problems*, Information Processing Letters, 44 (1992), pp. 245–249.

[68] I. Ljubić, *A hybrid vns for connected facility location*, Hybrid Metaheuristics, (2007), pp. 157–169.

[69] I. Ljubić, P. Putz, and J.-J. Salazar-González, *Exact approaches to the single-source network loading problem*, Networks, 59 (2012), pp. 89–106.

[70] M. E. Lübbecke and J. Desrosiers, *Selected topics in column generation*, Operations Research, 53 (2005), pp. 1007–1023.

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

[72] A. R. Mahjoub, L. Simonetti, and E. Uchoa, *Hop-level flow formulation for the survivable network design with hop constraints problem*, Networks, 61 (2013), pp. 171–179.

[73] P. Manyem, *Constrained spanning, steiner trees and the triangle inequality*, Optimization, (2009), pp. 355–367.

[74] A. Meyerson, K. Munagala, and S. Plotkin, *Cost-distance: Two metric network design*, Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on, (2000), pp. 624–630.

[75] M. Pal, E. Tardos, and T. Wexler, *Facility location with nonuniform hard capacities*, In proceedings of FOCS 2001, (2001), pp. 329–338.

[76] S. Raghavan and D. Stanojević, *A note on search by objective relaxation*, Telecommunications planning: innovations in pricing, network design and management, (2006), pp. 181–201.

[77] C. Randazzo, H. P. L. Luna, P. Mahey, et al., *Benders decomposition for local access network design with two technologies.*, Discrete Mathematics & Theoretical Computer Science, 4 (2001), pp. 235–246.

[78] R. Ravi and A. Sinha, *Approximation algorithms for problems combining facility location and network design*, Operations Research, 54 (2006), pp. 73–81.

[79] M. Rezapour, *Network design with facility location: Approximation and exact techniques*, Ph.D. Thesis, Technische Universität Berlin, (2015).

[80] G. Robins and A. Zelikovsky, *Improved steiner tree approximation in graphs*, Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms, (2000), pp. 770–779.

[81] F. S. Salman, R. Ravi, and J. N. Hooker, *Solving the capacitated local access network design problem*, INFORMS Journal on Computing, 20 (2008), pp. 243–254.

[82] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*, Springer, 24 (2003).

[83] D. B. Shmoys, É. Tardos, and K. Aardal, *Approximation algorithms for facility location problems*, Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, (1997), pp. 265–274.

[84] M. Sviridenko, *An improved approximation algorithm for the metric uncapacitated facility location problem*, Integer programming and combinatorial optimization, (2002), pp. 240–257.

[85] Z. Svitkina, *Lower-bounded facility location*, ACM Transactions on Algorithms (TALG), 6 (2010), p. 69.

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

[87] H. Takahashi and A. Matsuyama, *An approximate solution for the steiner problem in graphs*, Math. Japonica, 24 (1980), pp. 573–577.

[88] K. Talwar, *The single-sink buy-at-bulk lp has constant integrality gap*, Integer Programming and Combinatorial Optimization, (2002), pp. 475–486.

[89] A. Tomazic and I. Ljubić, *A grasp algorithm for the connected facility location problem*, Applications and the Internet, 2008. SAINT 2008. International Symposium on, (2008), pp. 257–260.

[90] V. V. Vazirani, *Approximation algorithms*, springer, (2001).

[91] J. Vygen, *Approximation algorithms facility location problems*, Report No. 05950-OR, Research Institute for Discrete Mathematics, University of Bonn, (2005).

[92] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani, *A primal-dual approximation algorithm for generalized steiner network problems*, Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, (1993), pp. 708–717.

[93] ——, *A primal-dual approximation algorithm for generalized steiner network problems*, Combinatorica, 15 (1995), pp. 435–454.

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

[95] L. A. Wolsey, *Integer programming*, Wiley New York, 42 (1998).

Please cite this article using:

Mohsen Rezapour, Algorithmic approaches for network design with facility location: A survey, AUT J. Math. Comput., 3(2) (2022) 193-206
DOI: 10.22060/AJMC.2022.21392.1086