

Facility Location for Congesting Commuters and Generalizing the Cost-Distance Problem

Thanasis Lianas¹, Marios Mertzaniadis², Aikaterini Nikolidaki³

¹University of West Attica

²Purdue University

³National Technical University of Athens

lianeas@corelab.ntua.gr, mmertzan@purdue.edu, aiknikol@mail.ntua.gr

Abstract

In Facility Location problems there are agents that should be connected to facilities and locations where facilities may be opened so that agents can connect to them. We depart from Uncapacitated Facility Location and by assuming that the connection costs of agents to facilities are congestion dependent, we define a novel problem, namely, Facility Location for Congesting (Selfish) Commuters. The connection costs of agents to facilities come as a result of how the agents commute to reach the facilities in an underlying network with cost functions on the edges. Inapproximability results follow from the related literature and thus approximate solutions is all we can hope for. For when the cost functions are nondecreasing we employ in a novel way an approximate version of Caratheodory's Theorem to show how approximate solutions for different versions of the problem can be derived. For when the cost functions are nonincreasing we show how this problem generalizes the Cost-Distance problem and provide an algorithm that for this more general case achieves the same approximation guarantees.

1 Introduction

Facility Location problems are among the well studied problems in Theoretical Computer Science (see, e.g., the related work section) and other fields (Reza Zanjirani Farahani 2009; Zvi Drezner 2001). In such problems, there are some facilities to be opened in available locations and agents need to connect to them. Opening facilities usually comes with a facility cost and connecting agents to their closest facility comes with connection costs. The goal is to minimize the sum of these costs. The connection costs are usually fixed and ignore congestion effects when connecting to the facilities. In this work, we assume that there are congestion effects when the agents commute to connect to the facilities.

Congestion Games are one of the most studied classes of games in Algorithmic Game Theory. In these games, agents choose subsets of resources and incur costs based on the congestion on the resources they have chosen. In Network Congestion Games, the very basic version of Congestion Games, there is an underlying network and agents choose paths in the network that connect their sources to their targets. Agents are selfish, aiming to minimize their costs without caring for any type of social welfare, which could be the

goal of a central organizer. For that reason, it makes sense to examine both outcomes where agents are satisfied by being on shortest paths, thus being at equilibrium, and outcomes where agents are assigned so that the social welfare, in any chosen sense, is minimized.

Departing from Uncapacitated Facility Location we define Facility Location for Congesting/Selfish Commuters. We assume there is an underlying network on the nodes of which facilities may be opened with a node-specific cost. Some nodes of the network are source nodes from where infinitesimally small agents, that form demands, depart to reach facilities. For any given set of facilities, there are many flows that may route the demands to the facilities, with the congestion cost varying among them. The goal is to minimize the sum of the facilities opening costs for the opened facilities plus the routing costs of the demands. When minimizing, similar to Congestion Games, we consider two cases for the flows, asking either to be simply feasible or to be equilibrium flows. In the first case, we say that we deal with Facility Location for Congesting Commuters and in the second case we say that we deal with Facility Location for Selfish Commuters.

The congestion costs are modeled using cost functions that come with the edges. Both cases of nondecreasing and nonincreasing cost functions on the edges have been considered in the Congestion Games literature. Interestingly, Facility Location for Congesting Commuters with nonincreasing cost functions generalizes the Cost-Distance problem (Meyerson, Munagala, and Plotkin 2008). In this problem, there is an underlying undirected network where demands are to be routed to some target and every edge comes with two types of costs. The first type of cost can be seen as a cost for building the edge and the second as a cost for traversing the edge. The goal is to return among all Steiner trees that connect the sources to the target the one that minimizes the building plus the routing costs.

Contribution: In this work we define Facility Location for Congesting Commuters (*FLCC* in short), a problem generalizing Facility Location Problems by assuming congestion effects that affect the agents when traveling/commuting to connect to their facilities. Moving one step further, we also define Facility Location for Selfish Commuters (*FLSC* in short) where the agents are not centrally controlled and are not assigned to paths in any convenient way. Instead, they

are selfish wanting to travel in shortest paths, thus we assume that they ought to be at equilibrium. If we see these problems through an Algorithmic Game Theory lens, it is like having Network Congestion Games where the targets for the demands have to be specified (with some cost) affecting any optimization task we wish to solve.

One can think of many variants for both *FLCC* and *FLSC* since the underlying network may be directed or undirected, may be a single-commodity, a single-source multi-commodity or a multi-commodity network and may have nonincreasing or nondecreasing cost functions, while, in addition, the opening costs for the facilities may be left arbitrary or be constrained, e.g., we may require to have the same opening cost for all facilities. We note that when the cost functions are constants the two problems coincide and reduce to Uncapacitated Facility Location for which hardness of approximation results are known (Guha and Khuller 1999). Thus, we can aim only for approximation algorithms for these problems. One can show that the techniques used to tackle Facility Location hardness results (Local search and Linear Programming) fail in this more general setting (at least if applied in a straightforward way).

Our first result concerns directed, single-source multi-commodity networks with nondecreasing latency functions and arbitrary opening costs. We employ ideas used in Congestion Games that work when the latency functions are α -Lipschitz, applying an approximate version of Caratheodory's theorem (Theorem 1) to provide an algorithm (Section 3) that performs well whenever the maximum length among the paths is $o(|V|)$ (Theorem 2). The presented algorithm approximately solves *FLSC*, but a similar and simpler approach can be used to approximately solve *FLCC*. We note that sparsification techniques similar to ours have been used in the past (Fotakis, Kaporis, and Spirakis 2012; Dimos et al. 2023) yet this is the first time that such techniques are applied for multi-target instances, where additionally the optimal placement for the targets, i.e., the facilities, has to be determined as well.

Our second result concerns undirected multi-commodity networks with nonincreasing cost functions and common opening costs for the facilities. In this case, as discussed already and as we show in Theorem 4, *FLCC* generalizes the Cost-Distance problem (Meyerson, Munagala, and Plotkin 2008) for which Chuzhoy et al. (2008) have shown an $\Omega(\log \log |S|)$ hard to approximate result, where S is the set of sources. Thus, this hardness result also holds for our more general setting. We propose an approximation algorithm (Subsection 4) for *FLCC* that is influenced by the algorithm of Meyerson, Munagala, and Plotkin (2008) but, in order to make the algorithm work for multicommodity networks and more general latency functions, it uses a novel matching mechanism (for matchings that have to be done therein). This complicates the analysis but is sufficient to provide the same approximation guarantees (Theorem 5). The proposed algorithm works for *FLCC* but can also be used for *FLSC* with the factor of approximation multiplied by $PoA(D)$, i.e., the Price of Anarchy (Koutsoupias and Papadimitriou 1999) for cost functions in class D (Correa, Schulz, and Moses 2004). Conceptually, $PoA(D)$ captures

how much worse can an equilibrium flow be when compared to the optimal flow for the cost functions used.

Detailed proofs can be found in (Lianeas, Mertzaniadis, and Nikolidaki 2025).

Related Work: Facility Location problems are among the most well-studied problems in algorithmic literature. Depending on the costs and constraints of the problem different variants of facility location emerge. Our main problem can be seen as a generalization of the Metric Uncapacitated Facility Location problem. The first constant factor approximation algorithm for this problem was provided by Shmoys, Tardos, and Aardal (1997). They provided a guarantee of 3.16 times the optimal cost. Jain and Vazirani (2001) use a Primal-Dual technique to achieve a 3-approximate algorithm. Chudak and Shmoys (2003) found a $(1 + 2/e)$ -approximate algorithm and finally Li (2013) provided a 1.488-approximation algorithm for the problem. As far as inapproximability results are concerned, the work of Guha and Khuller (1999) prove that it is NP-Hard to approximate this problem with a factor better than 1.463.

We also examine a k -median variant of our main problem. Charikar et al. (1998) were able to provide a $O(\log(n) \cdot \log \log(n))$ -approximation algorithm for the k -median facility location, which was obtained by derandomizing and refining an algorithm proposed by Bartal (1996, 1998). Charikar et al. (2002) were able to provide the first constant approximation algorithm for the k -median problem with a guarantee of $6\frac{2}{3}$. Finally, Arya et al. (2004) provided a $3 + \epsilon$ approximation algorithm. One can also derive a $(1 + \frac{2}{e} - \epsilon)$ inapproximability result by adapting the proof of hardness for the facility location problem provided by Guha and Khuller (1999) (this was observed by Jain et al. (2003)). Additionally many other variants of facility location problems have been studied such as Multi-Level Facility Location (Aardal, Chudak, and Shmoys 1999; Guha, Meyerson, and Munagala 2000), Connected Facility Location (Karger and Minkoff 2000; Gupta et al. 2001; Swamy and Kumar 2004; Gupta, Kumar, and Roughgarden 2003), and Universal Facility Location (Vygen 2007; Angel, Thang, and Renault 2014; Mahdian and Pal 2003; Pandit 2004; Bansal, Garg, and Gupta 2018).

Facility Location problems where congestion effects are present have been studied in the past although in those works the congestion does not affect the connection costs, while, additionally, such works are mainly experimental (Hajiaghayi, Mahdian, and Mirrokni 2003; Marianov, Ríos, and Icaza 2008; Dan and Marcotte 2019; Baron, Berman, and Krass 2008; Seifbarghy and Mansouri 2016; Eshaghi et al. 2019; Chakraborty and Vaze 2022; Wang et al. 2023; Jalili Marand and Hoseinpour 2024). Instead, our work is purely theoretical and to the best of our knowledge is the first to consider that the congestion affects the connection costs, and it does so in a complex way similar to that assumed in (network) Congestion Games.

Congestion Games (Rosenthal 1973) provide a natural model for non-cooperative resource allocation in large-scale communication networks and have been the subject of intensive research in Algorithmic Game Theory. Many variants have been considered in the literature but the one closely

related to ours is that of nonatomic network Congestion Games, also known as non atomic Selfish Routing. Through the use of a potential function (Rosenthal 1973) equilibrium existence is guaranteed and its computation reduces to solving a convex program, which is also the case in our problem.

The selfish behavior of the users may cause inefficiency, measured using the Price of Anarchy (Koutsoupias and Papadimitriou 1999). Related to our work, one way to reduce this inefficiency is to find and remove edges that cause the so called Braess Paradox (Braess 1968; Murchland 1970), i.e., that removing edges may improve the networks' performance. Braess Paradox seems not an artifact of optimization theory (Kelly 2008; Roughgarden 2005; Valiant and Roughgarden 2010; Chung and Young 2010) and resolving it has been proved hard to approximate even for simple instances (Roughgarden 2006). This line of work relates to our work in two ways. First, resolving the paradox lies under the umbrella of Network Design, which is also the case for Facility Location for Congesting/Selfish Commuters. Second, in this work we employ and generalize techniques that have been applied for resolving the paradox (Fotakis, Kaporis, and Spirakis 2012; Dimos et al. 2023).

Facility Location for Congesting/Selfish Commuters generalizes the Cost-Distance Problem. It was introduced by Meyerson, Munagala, and Plotkin (2000, 2008), and generalizes many important problems, e.g., single-sink buy-at-bulk with variable pipe types between different sets of nodes, facility location with buy-at-bulk type costs on edges, constructing single source multicast trees with good cost and delay properties, and multi-level facility location. In the same work Meyerson et al. proposed the first known $O(\log |S|)$ randomized approximation algorithm for the single-sink case on a given undirected graph, where S is the set of sources. This algorithm was subsequently derandomized by Chekuri, Khanna, and Naor (2001). Chuzhoy et al. (2008) presented an $\Omega(\log \log |S|)$ inapproximability result for the single-sink case, which also holds for our problems.

2 Preliminaries

For both **Facility Location for Congesting Commuters** and **Facility Location for Selfish Commuters** (*FLCC* and *FLSC* in short, respectively) an instance is defined in a similar way and the difference from one problem to the other is an extra constraint regarding the routing. An instance for these problems consists of a network $G(V, E)$ (directed or undirected) together with a set of source nodes $S \subseteq V$, a set of continuous non-negative cost functions $\{\ell_e\}_{e \in E}$ for the edges, traffic demands $\{w_i\}_{i \in S}$ for the source nodes, consisting of infinitesimal agents, and a set of costs $\{B_i\}_{i \in V}$ that capture the (possibly different) costs for opening facilities on the nodes. For convenience we set $|V| = n$ and $|E| = m$ and in the case of single source instances, i.e., instances where $S = \{s\}$, w.l.o.g. we assume that $w_s = 1$, since cost functions can be adapted appropriately (setting $\ell_e^{new}(x) = \ell_e(w_s x)$).

Cost functions. For every edge e , ℓ_e maps non-negative reals to non-negative reals, i.e. $\ell_e : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$. We care both for nondecreasing and nonincreasing cost functions.

For the first part of this work every ℓ_e is assumed to be non-decreasing. For the second part we focus on nonincreasing cost functions, additionally asking for every ℓ_e to be such that $x \cdot \ell_e(x)$ is a nondecreasing concave function. This case can be seen as agents sharing the cost of an edge they use, with the total cost of each edge increasing if more agents use the edge. In each section it will be clearly noted whether we examine the nondecreasing or nonincreasing variant.

Flows. The facilities may open on the network's nodes and act as targets to which the traffic demands travel. Once they are opened, the demands travel from their sources to the targets/facilities forming flows. Formally, let $F \subseteq V$ denote the nodes with opened facilities and $\mathcal{P}_{i,F}$ denote the network's paths that start from node $i \in S$ and end with a node in F . Also, let $\mathcal{P}_F = \cup_{i \in S} \mathcal{P}_{i,F}$. Given F , a flow $x = \{x_p\}_{p \in \mathcal{P}_F}$ is a vector assigning nonnegative reals to the paths of \mathcal{P}_F , and it is feasible if for every $i \in S : \sum_{p \in \mathcal{P}_{i,F}} x_p = w_i$. For a flow x and an edge e , we let $x_e = \sum_{p: e \in p} x_p$ denote the amount of flow that x routes through e . To denote a feasible flow for a set of facilities F we write $x(F)$ and we may write x instead of $x(F)$ whenever F is clear from the context.

Costs. Given F and a feasible flow x , we say that a path p is used if for all $e \in p : x_e > 0$. The cost of a path p under x is the sum of the costs of its edges, i.e., $\ell_p(x) = \sum_{e \in p} \ell_e(x_e)$. The total routing cost of a flow x is $RC(x) = \sum_{p \in \mathcal{P}} x_p \cdot \ell_p(x)$ or equivalently $RC(x) = \sum_{e \in E} x_e \cdot \ell_e(x_e)$.

Equilibria. We say that x is a Nash flow if for all $i \in S$, and for any used path $p \in \mathcal{P}_{i,F}$ and any path $p' \in \mathcal{P}_{i,F}$ we have $\ell_p(x) \leq \ell_{p'}(x)$, i.e., for every source the corresponding demand is routed through minimum cost paths. We say that x is an ϵ -Nash flow if for all $i \in S$, and for any used path $p \in \mathcal{P}_{i,F}$ and any path $p' \in \mathcal{P}_{i,F}$ we have $\ell_p(x) \leq \ell_{p'}(x) + \epsilon$. Using potential function arguments (the continuous version of (Rosenthal 1973)) one can show that, given F , a Nash flow always exists.

Solutions and Objective. Given an *FLCC* or an *FLSC* instance, a solution to it is a set $F \subseteq V$ determining the nodes on which a facility opens. A solution F is feasible if there exists a feasible flow x (given F). The goal for *FLCC* is to find a feasible solution that minimizes the sum of the total routing cost, under the opened facilities and searching among all feasible flows, plus the cost for the facilities that we opened,¹ i.e., among all feasible F and all feasible flows for F find the one minimizing $C(F) := \sum_{e \in E} x_e(F) \cdot \ell_e(x_e(F)) + \sum_{i \in F} B_i$. For *FLSC* the objective is the same, i.e., minimizing the above total cost, but there is an extra constraint that allows only flows $x(F)$ that are equilibria.

The following definitions serve as reference points for the problems we generalize in this work.

Cost-Distance Problem (Meyerson, Munagala, and Plotkin 2008). We are given an undirected graph $G(V, E)$ along with a set of source nodes $S \subseteq V$ which need to be connected to a single sink node $t \in V$. Each

¹We mix routing and opening costs but we can assume that, e.g., opening costs are multiplied to be measured in routing costs units.

node $s \in S$ comes with an associated demand w_s to be routed to t . Each edge $e \in E$ is equipped with a cost c_e and a length l_e . We may write such an instance as $(G(V, E), S, t, \{w_s\}_{s \in S}, \{c_e, l_e\}_{e \in E})$ in short. The goal is to find a connected subgraph $G'(V', E')$ of G which contains all the source nodes and the sink and minimizes the sum $\sum_{e \in E'} c_e + \sum_{s_i \in S} w_s \cdot l(s_i, t)$, where $l(s_i, t)$ denotes the length of the shortest s_i - t path in G' .

Facility Location and k-median. The problem we depart from is similar to the Metric Uncapacitated Facility Location Problem. We are given an undirected graph with costs $\{c_e\}_{e \in E}$ on the edges. On every vertex i there might be a demand d_i and a facility can be opened with opening cost B_i . The goal is to open some facilities where the demands can be connected to, so that the sum of the total facility opening cost and the weighted cost of connecting the demands to the facilities is minimized. For the k-median variant there are no opening costs and the goal is to open (up to) k facilities so that the weighted cost of connecting the demands to the facilities is minimized.

3 Instances with Nondecreasing Functions

In this section we consider *FLCC* and *FLSC* on directed networks with nondecreasing cost functions. One can show that the approaches that give approximation algorithms for Facility Location problems fail in our more general case (at least if applied in a straightforward way). Instead, inspired by techniques that have been used in Congestion Games, we restrict our attention on single-source instances, i.e., instances where $|S| = 1$. We additionally assume that all cost functions are α -Lipschitz.

Our proposed algorithm approximately solves *FLSC* but can be easily adapted so that it works for *FLCC*. It makes use of an approximate version of Caratheodory's theorem to provide an algorithm that solves any such instance and performs well (timewise) whenever the maximum length among the paths is $o(|V|)$, e.g., it runs in polynomial or quasipolynomial time when the length of the paths of the network is bounded by a constant or a polylog factor, respectively. Note that this restricts also the number of paths $|P|$. Since we focus on single-source instances, w.l.o.g. we assume that the total demand is $w_s = 1$.

In the following we outline our approach. The approximate version of Caratheodory's theorem that we use is the following.

Theorem 1 (Barman 2018). *Let X be a set of vectors $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ and $\epsilon > 0$. For every $\mu \in \text{conv}(X)$ and $2 \leq p \leq \infty$ there exists an $O(\frac{p \cdot \gamma^2}{\epsilon^2})$ -uniform vector $\mu' \in \text{conv}(X)$ such that $\|\mu - \mu'\|_p \leq \epsilon$, where $\gamma = \max_{x \in X} \|x\|_p$ and a vector is k -uniform when it can be expressed as an average of k vectors of X with replacements allowed, i.e., as $\sum_{j=1}^k \frac{x_{i_j}}{k}$, with $x_{i_1}, \dots, x_{i_k} \in X$.*

We will apply the theorem for points in \mathbb{R}^{n+m} using the $\|\cdot\|_2$ norm. We consider arbitrary orderings for the m edges and the n nodes of the network and for each path $p \in \mathcal{P}$ we create vector $x_p = (x_{p,1}, \dots, x_{p,m}, x_{p,m+1}, \dots, x_{p,m+n})$ where $x_{p,e} = 1$ if $e \in p$ and 0 otherwise, for $1 \leq e \leq m$,

and $x_{p,m+v} = 1$ if p ends in facility v and 0 otherwise, for $1 \leq v \leq n$. Let $X = \{x_p\}_{p \in \mathcal{P}}$ be the set of created vectors.

Using Theorem 1 one can show that there exists a flow \hat{f} that uses (at most) $k = O(\frac{\alpha^2 \cdot M^3}{\epsilon^2})$ paths of X , is an ϵ -Nash flow, and has routing cost $\leq RC(f^*) + \frac{\epsilon}{2}$, that additionally uses facilities that are opened under the optimal solution, implying that the total cost of \hat{f} is $\leq C(F^*) + \frac{\epsilon}{2}$. Our proposed algorithm exhaustively searches among all flows that may arise as k -combinations of paths in G . For each of them, it checks which facilities are opened and whether the flow is an ϵ -Nash flow. It returns the F for which the corresponding sum of routing and facilities cost is minimized, and a corresponding ϵ -Nash flow. Thus, we get the following theorem.

Theorem 2. *For any FLSC instance with α -Lipschitz cost functions and any $\epsilon > 0$ we can find in time $|\mathcal{P}|^{O(\frac{\alpha^2 \cdot M^3}{\epsilon^2})} \cdot O(\text{poly}(n))$ a feasible solution F and an ϵ -Nash flow (that routes to these facilities) with total cost at most $C(F^*) + \frac{\epsilon}{2}$, where M is an upper bound on the length of the paths*

4 Instances with Nonincreasing Functions

In this section we focus on *FLCC* instances on undirected networks with nonincreasing cost functions where additionally we require $x \cdot \ell_e(x)$ to be increasing and concave. For ease of presentation we will call such functions *good*. As we show, in the optimal solution demands do not split and thus our approach works also for unsplitable demand. Here, we assume a common opening cost $B_i = B$ for any facility i .

We start the section by revealing the structure of the optimal solution, which is needed for our algorithm's analysis but can also be used to show that this *FLCC* variant generalizes the Cost-Distance problem, in the sense that solving the Cost-Distance problem reduces to solving *FLCC* for good cost functions. Next we present an algorithm for approximately solving *FLCC* for good cost functions and we close the section with the algorithm's analysis.

The Optimal Solution is a Forest A key property on which we base our algorithm is that there exists an optimal solution where the flows form a forest on the graph, i.e., there is an optimal solution that opens k facilities and the edges used by the flows do not form cycles and thus can be seen as k trees rooted in each one of the opened facilities. We prove this in the following lemmas.

Lemma 1. *In any FLCC instance with good cost functions, there exists an optimal flow where there are no directed flow carrying cycles, i.e., cycles in which all demand flows in one direction.*

Proof. Let f^* be an optimal flow. For any cycle C that carries positive flow in the same direction, we can decrease f^* on the edges of the cycle by $\min_{e \in C} f_e^*$ getting a feasible solution with cost no more than the initial cost, since the $x \cdot \ell_e(x)$'s are nondecreasing, and thus with optimal cost. Repeatedly we can eliminate all such cycles and get an optimal acyclic flow. \square

Lemma 2. *In any FLCC instance with good cost functions, there exists an optimal flow where demands do not split their flows once they have met.*

Proof. Suppose there is a node in an optimal solution where some demand arrives and then a portion x_1 of this demand is routed through path P_1 and another portion x_2 is routed through path P_2 where $P_1 \neq P_2$. We will show that we can get another optimal flow on the same facilities where x_1 and x_2 are routed on the same path. Recall that any optimal solution minimizes the sum $\sum_{e \in E} x_e(F) \cdot \ell_e(x_e(F)) + \sum_{i \in F} B_i$ and thus $\sum_{e \in E} x_e(F) \cdot \ell_e(x_e(F))$ itself should be minimum among all flows routing to the facilities in F .

If we treat x_1 and x_2 as variables the total cost that is being paid on the edges of path P_1 that are not in P_2 and the total cost that is being paid on the edges of path P_2 that are not in P_1 are respectively: $C_1(x_1) = \sum_{e \in P_1 \setminus P_2} (w'_e + x_1) \cdot \ell_e(w'_e + x_1)$ and $C_2(x_2) = \sum_{e \in P_2 \setminus P_1} (w'_e + x_2) \cdot \ell_e(w'_e + x_2)$ where w'_e is the rest of the demand that is being routed through e in the optimal solution at hand.

Since all parts of the sum (i.e., the $(w'_e + x_1) \cdot \ell_e(w'_e + x_1)$'s) are concave functions (with respect to x_1 or x_2) then also $C_1(x_1)$ and $C_2(x_2)$ are concave functions. Since we have assumed that the flow is optimal, if we take the first derivatives of $C_1(x_1)$ and $C_2(x_2)$ it should hold that $C'_1(x_1) = C'_2(x_2)$. If this was not the case and for example it was $C'_1(x_1) > C'_2(x_2)$ then we could move a small amount of demand from P_1 to P_2 and the total cost would drop.

Using $C'_1(x_1) = C'_2(x_2)$ and that $C_1(x_1)$ and $C_2(x_2)$ are concave functions, we get that for any $d \leq x_1$

$$\begin{aligned} \frac{C_1(x_1) - C_1(x_1 - d)}{d} &\geq C'_1(x_1) = C'_2(x_2) \\ &\geq \frac{C_2(x_2 + d) - C_2(x_2)}{d} \end{aligned}$$

$$\Rightarrow C_1(x_1) + C_2(x_2) \geq C_1(x_1 - d) + C_2(x_2 + d)$$

implying, by setting $d = x_1$, that we can move all the demand x_1 from P_1 to P_2 without increasing the total cost, remaining at optimal routing cost. We can repeatedly do this for any split of demand to P_1 and P_2 ending up with a flow with optimal cost where the demands route their flows on single paths. \square

Lemma 3. *In any FLCC instance with good cost functions, there exists an optimal flow for which there are no cycles formed by the edges with positive flow.*

Proof. Let f^* be an acyclic optimal flow where the demands do not split, like the one guaranteed by Lemmas 1 and 2. Consider the flow carrying edges and, in order to reach a contradiction, assume that there is a cycle formed. Since f^* is acyclic there must be a node v where both of the edges of v send flow away from v , contradicting that in f^* the demands do not split. \square

From Lemmas 1-3 it follows that:

Theorem 3. *In any FLCC instance with good cost functions, there exists an optimal flow for which there are no cycles formed by the edges with positive flow and the demands use single paths and do not split once they have met.*

FLCC generalizes the Cost-Distance Problem:

Theorem 4. *The Cost-Distance problem polynomially reduces to solving FLCC with good cost functions.*

Proof idea. Consider an instance of the Cost-Distance problem, $I_{CD} = (G(V, E), S, t, \{w_s\}_{s \in S}, \{c_e, w_e\}_{e \in E})$. Create an FLCC instance on graph G , with set of source nodes $S \cup \{t\}$, demands as in I_{CD} and for node t , $w_t = \sum_{s \in S} w_s$, a sufficiently large B so that only one facility opens in the optimal solution and for each e , the latency function $\ell_e(x) = \frac{c(e)}{\max\{x, \min_{s \in S} w_s\}} + l(e)$. Given a solution F of the FLCC instance that (w.l.o.g., as it turns out) opens a facility on t , the solution returned for I_{cd} is the subgraph $G(V', E')$ used by the flow under F . The optimality is guaranteed by the fact that minimizing $\sum_{e \in E} x_e(F) \cdot \ell_e(x_e(F)) + \sum_{i \in F} B$ reduces to minimizing $\sum_{e \in E'} c_e + \sum_{s_i \in S} w_s \cdot l(s_i, t)$. \square

The Algorithm An observation that simplifies our approach is that we may focus on the k -median variant of FLCC. In this variant there are no opening costs for the facilities, only routing costs, and we may open at most k facilities. Solving the k -median variant for all possible k values, solves the original problem since for any k we can compute the sum of the routing cost of the optimal solution with the original problem's opening cost, which for any k will be $k \cdot B$ (recall, we have a common cost $B_i = B$) and then compare these sums for the n different values of k in order to keep the optimal one.

Below we give an algorithm that takes as input an instance of FLCC and returns k facility points for the k -median version of FLCC. The algorithm is influenced by the algorithm proposed by Meyerson, Munagala, and Plotkin (2008) for the Cost-Distance problem which returns one target point (i.e., one facility, in our setting). However, their matching mechanism and analysis are tailored made to the fact that the total cost incurred by each edge is linear. We introduce a novel matching mechanism which in turn complicates the resulting analysis. For the algorithm we will need the following distance metric: $g(u, v, w) = \sum_{e \in P_{u,v}^w} w \cdot \ell_e(w)$ where $P_{u,v}^w$ is the closest path from u to v with respect to the distance metric $\ell_e(w)$.

To give a high level view of the algorithm, the algorithm works in phases. In every phase it pairs up the demands. For every such pair it routes both demands from their sources to a chosen meeting point, and then it randomly (in a biased way) routes them back to one of the two demand sources. For the next phase, every pair of paired demands is handled as one demand of volume equal to the sum of the two demands and source the randomly chosen source in the previous phase. The algorithm will terminate when some phase ends with k demands, and it will open facilities right on the sources of these demands.

Two important ingredients of the algorithm is how the pairing is done (steps 2(a) and 2(b) below) and how the common source for the merging demands is chosen (step 2(d)ii). The reasons for doing the pairing and the routing this way will become apparent in the next section that analyzes the algorithm. For step 2(b) the requirement for having nodes

unmatched is there to avoid forcing to match nodes that are in different rooted trees of the optimal solution (in case these are of odd cardinality). Recall that the optimal solution can be seen as k trees rooted in each one of the opened facilities.

Approximation scheme for nonincreasing cost functions

1. Initialize $S_0 = S$, $w_{0,s} = w_s$ and $i = 0$.
2. While $|S_i| > k$:
 - (a) For every pair $u, v \in S_i$ find $K_i(u, v) = \min_{z \in V} \left\{ g(u, z, w_{i,u}) + g(v, z, w_{i,v}) + \frac{w_{i,u}}{w_{i,u} + w_{i,v}} \cdot g(z, u, w_{i,u} + w_{i,v}) + \frac{w_{i,v}}{w_{i,u} + w_{i,v}} \cdot g(z, v, w_{i,u} + w_{i,v}) \right\}$.
 - (b) Perform a maximum cardinality minimum cost matching on S_i with respect to the costs K_i under the constraint that, unless you get an empty matching, k nodes must be unmatched.
 - (c) Set $S_{i+1} = \{\}$.
 - (d) For each matched u, v pair:
 - i. Send both demands to the node z that minimized the expression of step 2(a).
 - ii. Choose u with probability $\frac{w_{i,u}}{w_{i,u} + w_{i,v}}$, otherwise chose v . Without loss of generality we will assume that we chose u .
 - iii. Send the combined $w_{i,u} + w_{i,v}$ demand back to u , add u to S_{i+1} and set $w_{i+1,u} = w_{i,u} + w_{i,v}$.
 - (e) Add unmatched nodes to S_{i+1}
 - (f) Set $i \leftarrow i + 1$
3. Return as facilities the k nodes that are in S_i and as flows the ones dictated by the above procedure.

The Analysis For our analysis we introduce the metric C_u^i to be the total cost payed due to the movement of demand from source u in phase i . The total cost payed in phase i is $C^i = \sum_{s \in S_i} C_s^i$. Since at every step the number of sources minus k is roughly divided by two and we end up with k (final) sources, it is not difficult to show the following lemma.

Lemma 4. *The algorithm presented terminates after $O(\log |S|)$ phases.*

If we prove that in each phase of the algorithm the expected cost payed (i.e. $\mathbf{E}[C^i]$) is at most the cost of the optimal solution, then combining this fact with Lemma 4 shows that our algorithm is an $O(\log |S|)$ -approximate algorithm for the *FLCC* with good functions. We will show that this is true for the first phase and that the expected cost of each phase is less than or equal to the cost of the first phase.

Phase 0 Phase 0 is the first phase of our algorithm where no demands have been aggregated yet. We begin with the following lemma which is a generalization of (Meyerson, Munagala, and Plotkin 2008, Lemma 4.2) and reveals some structural property of the optimal solution. Recall, again, that the optimal solution can be seen as k trees rooted in each one of the opened facilities.

Lemma 5. *Let $S' \subseteq S$ be a set of sources that are routed at a facility in node r in the optimal solution. Lets call T the tree that corresponds to the flows from nodes in S' to r . Then*

there exists a matching like the one done in steps 2(a),(b) of our algorithm where demands only use edges that they use in the optimal flow and at most one source is left unmatched.

Proof. Intuitively, we perform steps 2(a),(b) conditioned that the paths for computing the $g(\cdot, \cdot, \cdot)$'s are restricted to belong in T . For every $s \in S'$ we take its path towards r at T until it meets with the path of another source. We will call the vertex in which the two paths merge a level one meeting point. We move all sources at their corresponding level one meeting points. In each meeting point with an even number of demands we match them arbitrarily until no demand is left unmatched. In each meeting point with an odd number of demands we do the same thing however now there will be one demand left out. We continue along the path of those level one meeting points with an unmatched demand until their path merges with the path of another level one meeting point. The vertices in which those level one paths meet are called level two meeting points. We continue along the same line of thought until we reach our final meeting point r where at most one demand will be left unmatched. \square

In the algorithm in steps 2(a)-(b) we search for the best (w.r.t. cost) such matching and we choose as z (in step 2(d)i) the respective meeting point described by the proof of Lemma 5. Thus, the returned matching cannot be worse than the one dictated by Lemma 5 and it will be without loss of generality that in this phase but also in every other phase we can assume that the edges used are exactly those found by the matching (we will route the same demand on paths at most as expensive, in total, as those of the optimal solution). To bound the cost of our matching we just need to bound the cost of sending demands to those meeting points and the expected demand of sending them back again. The first half is accomplished with the following lemma.

Lemma 6. *The cost of sending all demands to their respective meeting points is less than that of the optimal solution.*

Proof. From Lemma 5 there is a subtle implication that becomes very useful right now. Each edge is traversed by demands that traverse that edge in the optimal solution. More specifically suppose that an edge $e \in E$ is traversed by a set S'' of demands. In our matching only one of this demands traverses that edge. Thus the amount of flow f'_e traversing edge e in our matching is less or equal than the amount of flow f_e that traverses e in the optimal solution. And since $x \cdot \ell_e(x)$ is nondecreasing we have that $f'_e \cdot \ell_e(f'_e) \leq f_e \cdot \ell_e(f_e)$. Summing over all edges we get that the total cost is less than the optimal cost. \square

The next Lemma holds for any phase i of the algorithm.

Lemma 7. *The expected cost of routing demands back from their meeting point to the selected source is less than or equal to the cost paid for sending them to the meeting point.*

Proof. Suppose we are at phase i and let a_1, a_2, \dots, a_m be the edges along the path of u to the meeting point and b_1, b_2, \dots, b_n the ones of v . Then the expected cost $\mathbf{E}[C_{\text{back}}]$ of sending them back to a source is the cost of sending them to u times the probability of choosing u plus the probability

of sending them to v times the cost of sending them there. In other words we have that $\mathbf{E}[C_{\text{back}}] = (w_{i,u} + w_{i,v}) \cdot \frac{w_{i,u}}{w_{i,u} + w_{i,v}} \cdot \sum_{j=1}^m \ell_{a_j}(w_{i,u} + w_{i,v}) + (w_{i,u} + w_{i,v}) \cdot \frac{w_{i,v}}{w_{i,u} + w_{i,v}} \cdot \sum_{j=1}^m \ell_{b_j}(w_{i,u} + w_{i,v}) \leq w_{i,u} \cdot \sum_{j=1}^m \ell_{a_j}(w_{i,u}) + w_{i,v} \cdot \sum_{j=1}^m \ell_{b_j}(w_{i,v})$ which is exactly the cost of sending those demands to the meeting point. The inequality holds since the demands are positive and the ℓ_e 's are nonincreasing. \square

From Lemmas 6 and 7 we can see that $C^0 \leq 2 \cdot C^*$ where C^* is the total cost of the optimal solution.

Phase i The only thing left to do is to bound the cost of sending demands to their meeting points in phase i of the algorithm. This can be done using the following lemma.

Lemma 8. *The expected cost of routing demands to their meeting points in phase i is less than the cost of the optimal solution.*

Proof. Suppose at phase i that in a node u a set S''_u of demands has been gathered with a total demand $W_u = \sum_{s \in S''_u} w_s$. The probability of u having this demand is $\frac{w_u}{W_u}$. This claim is easy to see because for u to have that demand it must have been selected in all phases with a total probability of $\frac{w_u}{w_u + w_{v_1}} \cdot \frac{w_u + w_{v_1}}{w_u + w_{v_1} + w_{v_2}} \cdot \dots \cdot \frac{\sum_{s \in S''_u} w_s - w_{u_{\text{last}}}}{\sum_{s \in S''_u} w_s}$, and in this product only the first numerator and the last denominator do not cancel out. We once again perform the matching described by Lemma 5 with respect to weights $w_{i,u}$.

We are now going to show that the expected cost of the matching described by Lemma 5 and searched for in steps 2(a)-(b) is bounded by the cost of the optimal solution. For the sake of the argument let's call $g_e(x) = x \cdot \ell_e(x)$ the cost paid for the usage of edge e (recall $g_e(x)$ is a non-decreasing concave function). Suppose that a specific edge e in the optimal solution is used by demands w_1, w_2, \dots, w_k . Thus the total cost paid for edge e in the optimal solution is $g_e\left(\sum_{j=1}^k w_j\right)$. We will show that for this arbitrary e it is $\mathbf{E}[C_{i,e}] \leq g_e\left(\sum_{j=1}^k w_j\right)$, where $C_{i,e}$ is the cost paid in the matching of phase i by e . Summing over all e will conclude the proof.

In phase i the expected cost paid by edge e because of demand w_j is the cost paid because a total demand of weight W_{w_j} passes through e times the probability that all of the demand with which w_j has been matched, actually passes through e . That probability can be expressed as the probability of W_{w_j} ending up in w_j (which we have shown earlier that is equal to $\frac{w_j}{W_{w_j}}$), times the probability that demand is not matched earlier in the Tree. We will call the later probability p_{e,w_j} . Thus the expected cost paid by edge e due to demand w_j is $\frac{w_j}{W_{w_j}} \cdot p_{e,w_j} \cdot g_e(W_{w_j})$. Also we will call $p_{e,0}$ the probability that no demand passes through e in our matching either because with probability $\prod_{j=1}^k \left(1 - \frac{w_j}{W_{w_j}}\right)$ there are no demands in the subtree underneath e or there was an even number of demands in the subtree underneath e and thus all demands have been matched lower in the Tree. So the total expected cost of edge

e is $p_{e,0} \cdot g_e(0) + \sum_{j=1}^k \frac{w_j}{W_{w_j}} \cdot p_{e,w_j} \cdot g_e(W_{w_j})$. However, one can see that $p_{e,0} + \sum_{j=1}^k \frac{w_j}{W_{w_j}} \cdot p_{e,w_j} = 1$ and $g_e(x)$ is concave, so we can use Jensen's inequality (Jensen 1906) for concave functions which leads to the following expression:

$$\begin{aligned} \mathbf{E}[C_{i,e}] &\leq p_{e,0} \cdot g_e(0) + \sum_{j=1}^k \frac{w_j}{W_{w_j}} \cdot p_{e,w_j} \cdot g_e(W_{w_j}) \\ &\leq g_e\left(p_{e,0} \cdot 0 + \sum_{j=1}^k \frac{w_j}{W_{w_j}} \cdot p_{e,w_j} \cdot W_{w_j}\right) \\ &= g_e\left(\sum_{j=1}^k w_j \cdot p_{e,w_j}\right) \leq g_e\left(\sum_{j=1}^k w_j\right) \end{aligned}$$

where the last inequality holds because $p_{e,w_j} \leq 1$ and $g_e(x)$ is nondecreasing. This argument concludes our proof. \square

Combining the aforementioned lemmas we end up with the following lemma.

Lemma 9. *The routing cost of each phase is on expectation at most the cost of the optimal solution.*

Combining this lemma with the fact that our algorithm has $O(\log |S|)$ phases we get the following theorem.

Theorem 5. *The analyzed algorithm produces an $O(\log |S|)$ -approximate solution to FLCC with good cost functions.*

Letting, $PoA(D)$ be the Price of Anarchy (Koutsoupias and Papadimitriou 1999) for cost functions in class D (Correa, Schulz, and Moses 2004), one can get the following corollary. We note that our scope is not to provide a $PoA(D)$ bound rather to provide an approximation scheme for when such a bound is known.

Corollary 1. *The approximation scheme presented outputs an $[O(\log |S|) \cdot PoA(D)]$ -approximate solution to FLSC with good cost functions.*

5 Conclusion and the Case of Discrete Agents

We introduced Facility Location problems that account for congestion dependent connection costs. Among the many possible variants, we presented algorithms for two of them. Providing results for any of these variants is an interesting direction.

Our results can be of use if one considers discrete agents. One can use the algorithm in Section 3, that cuts the flow in small discrete particles, to solve the corresponding continuous case and then apply some type of rounding to get approximation guarantees. These guarantees will be close to the ones of the continuous case when the number of agents is large enough to allow for small enough cuts in the flow. In Section 4 we showed that in the optimal solution the demands do not split. Thus, the algorithm presented here works also for the discrete agents' case. Last, derandomizing this algorithm could be an interesting extension.

Acknowledgements

This research was partially supported by the framework of H.F.R.I call “Basic research Financing (Horizontal support of all Sciences)” under the National Recovery and Resilience Plan “Greece 2.0” funded by the European Union-NextGenerationEU (H.F.R.I. Project Number:15635). Marios Mertzaniadis is supported in part by an NSF CAREER award CCF-2144208, and a research award from the Herbert Simon Family Foundation.

References

- Aardal, K.; Chudak, F. A.; and Shmoys, D. B. 1999. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Information Processing Letters*, 72(5-6): 161–167.
- Angel, E.; Thang, N.; and Regnault, D. 2014. Improved Local Search for Universal Facility Location. *Journal of Combinatorial Optimization*, 29.
- Arya, V.; Garg, N.; Khandekar, R.; Meyerson, A.; Munagala, K.; and Pandit, V. 2004. Local Search Heuristics for k-Median and Facility Location Problems. *SIAM Journal on Computing*, 33(3): 544–562.
- Bansal, M.; Garg, N.; and Gupta, N. 2018. A 5-Approximation for Universal Facility Location. *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*.
- Barman, S. 2018. Approximating Nash Equilibria and Dense Subgraphs via an Approximate Version of Carathéodory’s Theorem. *SIAM J. Comput.*, 47(3): 960–981.
- Baron, O.; Berman, O.; and Krass, D. 2008. Facility Location with Stochastic Demand and Constraints on Waiting Time. *Manuf. Serv. Oper. Manag.*, 10(3): 484–505.
- Bartal, Y. 1996. Probabilistic approximation of metric spaces and its algorithmic applications. *Proceedings of 37th Conference on Foundations of Computer Science*.
- Bartal, Y. 1998. On approximating arbitrary metrics by tree metrics. *Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC 98*.
- Braess, D. 1968. Über ein paradox aus der Verkehrsplanung. *Unternehmensforschung*, 12: 258–268.
- Chakraborty, A.; and Vaze, R. 2022. Online facility location with timed-requests and congestion.
- Charikar, M.; Chekuri, C.; Goel, A.; and Guha, S. 1998. Rounding via Trees: Deterministic Approximation Algorithms for Group Steiner Trees and k-Median. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC ’98, 114–123. New York, NY, USA: Association for Computing Machinery. ISBN 0897919629.
- Charikar, M.; Guha, S.; Éva Tardos; and Shmoys, D. B. 2002. A Constant-Factor Approximation Algorithm for the k-Median Problem. *Journal of Computer and System Sciences*, 65(1): 129–149.
- Chekuri, C.; Khanna, S.; and Naor, J. 2001. A deterministic algorithm for the cost-distance problem. *Symposium on Discrete Algorithms: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, 7(09): 232–233.
- Chudak, F. A.; and Shmoys, D. B. 2003. Improved Approximation Algorithms for the Uncapacitated Facility Location Problem. *SIAM Journal on Computing*, 33(1): 1–25.
- Chung, F.; and Young, S. J. 2010. Braess’s Paradox in Large Sparse Graphs. In *Internet and Network Economics*, 194–208. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Chuzhoy, J.; Gupta, A.; Naor, J. S.; and mitabh Sinha. 2008. On the Approximability of Some Network Design Problems. *ACM Trans. Algorithms*, 4(2).
- Correa, J. R.; Schulz, A. S.; and Moses, N. E. S. 2004. Selfish Routing in Capacitated Networks. *Math. Oper. Res.*, 29(4): 961–976.
- Dan, T.; and Marcotte, P. 2019. Competitive Facility Location with Selfish Users and Queues. *Oper. Res.*, 67(2): 479–497.
- Dimos, S.; Fotakis, D.; Lianas, T.; and Sergis, K. 2023. Escaping Braess’s paradox through approximate Caratheodory’s theorem. *Inf. Process. Lett.*, 179: 106289.
- Eshghi, A.; Jahani, H.; Aghaie, A.; and Ivanov, D. 2019. A multi-layer congested facility location problem with consideration of impatient customers in a queuing system. *IFAC-PapersOnLine*, 52: 2279–2284.
- Fotakis, D.; Kaporis, A. C.; and Spirakis, P. G. 2012. Efficient methods for selfish network design. *Theoretical Computer Science*, 448: 9–20.
- Guha, S.; and Khuller, S. 1999. Greedy Strikes Back: Improved Facility Location Algorithms. *Journal of Algorithms*, 31(1): 228–248.
- Guha, S.; Meyerson, A.; and Munagala, K. 2000. Hierarchical Placement and Network Design Problems. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS ’00, 603. USA: IEEE Computer Society. ISBN 0769508502.
- Gupta, A.; Kleinberg, J.; Kumar, A.; Rastogi, R.; and Yener, B. 2001. Provisioning a virtual private network: A network design problem for multicommodity flow. *33rd Proc. STOC*, 389–398.
- Gupta, A.; Kumar, A.; and Roughgarden, T. 2003. Simpler and Better Approximation Algorithms for Network Design. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC ’03, 365–372. New York, NY, USA: Association for Computing Machinery. ISBN 1581136749.
- Hajiaghayi, M. T.; Mahdian, M.; and Mirrokni, V. S. 2003. The facility location problem with general cost functions. *Networks*, 42(1): 42–47.
- Jain, K.; Mahdian, M.; Markakis, E.; Saberi, A.; and Vazirani, V. V. 2003. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM*, 50(6): 795–824.
- Jain, K.; and Vazirani, V. 2001. Approximation algorithms for metric facility location and k-Median problems using the primal-dual schema and Lagrangian relaxation. *Journal of The ACM - JACM*, 48.

- Jalili Marand, A.; and Hoseinpour, P. 2024. A congested facility location problem with strategic customers. *European Journal of Operational Research*, 318(2): 442–456.
- Jensen, J. L. W. V. 1906. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30: 175–193.
- Karger, D.; and Minkoff, M. 2000. Building Steiner Trees with Incomplete Global Knowledge. *Annual Symposium on Foundations of Computer Science - Proceedings*, 613–623.
- Kelly, F. 2008. *The mathematics of traffic in networks*. In *The Princeton Companion to Mathematics* (Editors: T. Gowers, J. Green and I. Leader). Princeton University Press.
- Koutsoupias, E.; and Papadimitriou, C. 1999. Worst-case equilibria. *16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, 404–413.
- Li, S. 2013. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222: 45–58. 38th International Colloquium on Automata, Languages and Programming (ICALP 2011).
- Lianas, T.; Mertzanidis, M.; and Nikolidaki, A. 2025. Facility Location for Congesting Commuters and Generalizing the Cost-Distance Problem. arXiv:2511.10228.
- Mahdian, M.; and Pal, M. 2003. *Universal Facility Location*. Springer, Berlin, Heidelberg.
- Marianov, V.; Ríos, M.; and Icaza, M. J. 2008. Facility location for market capture when users rank facilities by shorter travel and waiting times. *European Journal of Operational Research*, 191(1): 32–44.
- Meyerson, A.; Munagala, K.; and Plotkin, S. 2000. Cost-distance: two metric network design. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 624–624. IEEE Computer Society.
- Meyerson, A.; Munagala, K.; and Plotkin, S. A. 2008. Cost-Distance: Two Metric Network Design. *SIAM J. Comput.*, 38(4): 1648–1659.
- Murchland, J. D. 1970. Braess’s paradox of traffic flow. *Transportation Res.*, 4: 391–394.
- Pandit, V. 2004. *Local Search Heuristics For Facility Location Problems*. Ph.D. thesis, Department of Computer Science and Engineering Indian Institute of Technology Delhi.
- Reza Zanjirani Farahani, M. H. 2009. *Facility Location. Concepts, Models, Algorithms and Case Studies*. Springer-Verlag Berlin.
- Rosenthal, R. W. 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 7.
- Roughgarden, T. 2005. *Selfish Routing and the Price of Anarchy*. MIT press.
- Roughgarden, T. 2006. On the severity of Braess’s Paradox: Designing networks for selfish users is hard. *Journal of Computer and System Sciences*, 72(5): 922 – 953. Special Issue on FOCS 2001.
- Seifbarghy, M.; and Mansouri, A. 2016. Modelling and solving a congested facility location problem considering systems’ and customers’ objectives. *International Journal of Industrial and Systems Engineering*, 22: 281.
- Shmoys, D. B.; Tardos, E.; and Aardal, K. 1997. Approximation algorithms for facility location problems. In *Proceedings of 29th Annual ACM Symposium on Theory of Computing*, 265–274.
- Swamy, C.; and Kumar, A. 2004. Primal–Dual Algorithms for Connected Facility Location Problems. *Algorithmica*, 40: 245–269.
- Valiant, G.; and Roughgarden, T. 2010. Braess’s Paradox in large random graphs. *Random Struct. Algorithms*, 37(4): 495–515.
- Vygen, J. 2007. From stars to comets: Improved local search for universal facility location. *Oper. Res. Lett.*, 35: 427–433.
- Wang, X.; Zhao, J.; Cheng, C.; and Qi, M. 2023. A multi-objective fuzzy facility location problem with congestion and priority for drone-based emergency deliveries. *Computers & Industrial Engineering*, 179: 109167.
- Zvi Drezner, H. W. H. 2001. *Facility Location. Applications and Theory*. Springer-Verlag Berlin.