Discrete Mathematics and Mathematical Programming

MASTER'S THESIS

Perturbation resilience for the facility location problem

M.B. Tijink

Supervisor:
Dr. B. Manthey (University of Twente, DMMP)

Graduation committee:
Prof.dr. M.J. Uetz (University of Twente, DMMP)
Dr. B. Manthey (University of Twente, DMMP)
Dr. J.C.W. van Ommeren (University of Twente, SOR)

Enschede,
August 25, 2017

UNIVERSITY OF TWENTE.

Abstract

Several NP-hard problems, e.g. k-means clustering, are solved very quickly and near optimality for practical instances, even though there are known worst-cases from a theoretical perspective. These worst-cases do not seem to appear in practice. There even is a saying; “Clustering is either easy or pointless”, indicating that the difficulty of these problems is only a theoretical concern. Several approaches to formalize this saying, by assuming that instances have a natural stability property in practice, recently proved to be successful in closing the gap between theory and practice.

In this thesis we look at the facility location problem, which is NP-hard. We add the γ-perturbation resilience assumption, which requires that the instance must allow small perturbations of all costs in the instance without changing its optimal solution. Many instances in practice likely already satisfy this assumption, so we focus on the theoretical impact of this assumption. We found several consequences this assumption; local search algorithms will always result in the optimal solution, for $\gamma \geq 3$, and may not give the optimal solution for all $\gamma < 3$. We also show that several greedy algorithms do not work either to solve the facility location problem with the γ-perturbation resilience assumption, even for high γ. A relation between this assumption and approximation algorithms seems obvious, but we prove that this relation is false. Finally, we show that, for small γ, the existence of an efficient algorithm to solve the facility location problem with the γ-perturbation resilience assumption implies that RP = NP.

Contents

1 Introduction

In this thesis we look at the facility location problem. This is a hard problem, as further explained in subsection 1.1, but the theoretical results for this problem show it to be more difficult than is observed in practice [17]. To solve this contrast between the theoretical and practical difficulty, we introduce the γ-perturbation resilience assumption for the facility location problem (see subsection 1.2): γ-perturbation resilient instances of the facility location problem do not have different optimal solutions when perturbed with small amounts. This assumption is somewhat natural: we think that many instances already have this property in practice, yet it makes the problem easier from the theoretical perspective. We will look into the consequences of making this assumption. In the following two subsections we will go into more detail about the facility location problem and the difference between NP-hardness in theory and practice, and we will detail the contributions made in this thesis in the final subsection.

1.1 The facility location problem

In the facility location problem we have a finite set of customers and a finite set of locations where we can open a facility. The goal of the facility location problem is to serve all customers while minimizing the total cost. This requires that each customer is connected to exactly one facility, which must be open. Thus, the key choice is which facilities to open and the secondary choice is which customer connects to which open facility. For every customer-location pair we have an associated cost, which we must pay if we we open a facility on that location and let the facility serve that customer. We also have associated costs for every location, which we must pay if we open a facility at that location. Thus, there is a tradeoff between opening many facilities such that the customer-location costs stay low and opening few facilities such that the facility costs are low. A common application of the facility location problem is determining where to build e.g. warehouses or hospitals. For more details on the facility location problem, see Krarup and Pruzan [20] or Melo et al. [25].

A lot of variants of this simple problem exist. For example, the uncapacitated facility location problem, as described above, allows an arbitrary number of customers to connect to an open facility. Thus, facilities must be very flexible, but this is not realistic for all application. The capacitated facility location problem limits the number of customers to connect to an open facility. This way you could, for example, limit a warehouse capacity to five customers. If the capacity is reached, another facility needs to be opened to serve the remaining customers, either at the same location or at another location. The metric facility location problem requires the customer-location costs to be metric, making them behave like distances. This variant is suitable when the distance or time used in a road network is important. The non-metric facility problem does not restrict the customer-location costs, making it more suited for optimizing other types of resources. Some facility location problem variants have per-customers demands and per-unit customer-location prices, instead of just a single customer-location cost. This may not seem useful on its own, but is useful when combined with e.g. the metric facility location problem, as the demand and unit price formulation of the facility location problem allows more instances when these prices must be metric. And, of course, these variants can usually be combined for even more variants of the facility location problem [25, 32].

The facility location problem can also be seen as a variant of the k-means problem, but with variable amount of cluster centers to open and costs depending on which cluster center you open [11, 12, 24].

Even the simplest of the facility location problem variants – the uncapacitated metric facility location problem, which we use in this thesis – is NP-hard, which implies there is no proof yet that deterministic polynomial time algorithms exist for solving all possible instances [20, 25].

The uncapacitated non-metric facility location problem is even NP-hard to approximate within a factor $O(\log|D|)$, where $|D|$ is the number of customers [28, 32], so even guaranteeing a good solution is a hard problem.

The uncapacitated metric facility location problem, which we use in this thesis, is NP-hard to approximate within a factor 1.463 [16]. On the other hand, there are approximation algorithms that guarantee a solution within a factor 1.5 of the optimal solution [10].

1.2 NP-hardness in practice

Problems in computer science, e.g. the shortest path problem or the traveling salesman problem, can be classified depending on how fast they can be solved, for the fastest algorithm which solves them. So even if a problem `A` has a very slow algorithm, there could also be faster algorithms, so `A` does not need to be classified corresponding with the slow algorithm. The most well known complexity classes are P, meaning Polynomial, and NP, meaning Nondeterministic Polynomial. If a problem `A` is in P, there is an algorithm that solves `A` in polynomial time, for any possible instance of `A`. Problems in NP can be solved in non-deterministic polynomial time or, equivalently, solutions to problems in NP can be verified in polynomial time given a proof of the solution [15, 27]. We often say a problem `A` in P is easy or that it has an efficient algorithm. On the other hand, we call call a problem `A` hard or say that it has only inefficient known algorithms if `A` is in NP. Other complexity classes also exist, for example we use complexity class RP, meaning Randomized Polynomial, in this thesis. Algorithms for problems in RP can use random numbers, so their output is not necessarily deterministic. An algorithm for some problem `A` in RP must always answer no if the given instance of `A` has no as answer, and the algorithm must answer yes with probability at least $\frac{1}{2}$ if the answer is yes. [27].

Proving that a problem is at least as hard as other problems is usually done with reductions. A reduction transforms instances of some problem `B` to an instance of some problem `A` such that the answer to the instance of problem `B` is identical to the answer of transformed instance. This way, if you have algorithm which can solve `A`, you can solve `B` by first running the transformation and then solving the resulting instance of problem `A`. Thus, a reduction proves that problem `B` is at least as hard as problem `A` [15, 27].

Cook showed that every problem in NP can be reduced to the boolean satisfiability problem [13], making it the first so-called NP-complete problem. If you prove that an NP-complete problem is in P, you have proved that P = NP. Likewise if you prove that an NP-complete problem is not in P, you have proved that P $\neq$ NP. Thus, NP-complete problems can be seen as a characterization of NP. Using a reduction from an NP-complete problem `B` to some other problem `A` in NP proves that `A` also is NP-complete. Examples of NP-complete problems are the SAT, clique, traveling salesman and knapsack problems [19]. The facility location problem is also NP-complete [20].

Despite all the research being done on those problems, to date no algorithms have been found to solve these problems in deterministic polynomial time; the question "P = NP or P $\neq$ NP?" is still open. For optimization problems, where the goal is to find the optimal solution, this is even worse: some problems of approximation, where the goal is to find a solution with value within a certain factor of the optimal solution, are still NP-hard, e.g. for the facility location problem and traveling salesman problem [14, 16].

In practice these theoretical problems have turned out to be less important: heuristics and local search algorithms often yield good results, even though they have worst cases on which they do not give good results or take exponential time [30]. This even gave rise to the saying "Clustering is either easy or pointless" [8]. Further examples are 2-OPT for the traveling salesman problem [18] and the simplex algorithm [30]. It seems as if those difficult, worst-case instances do not appear in practice, either because these instances are very rare or because practical instances have some inherent properties making them easier. The first of those reasons gave rise to smoothed analysis, in which small random perturbations are applied to instances. The idea in smoothed analysis is that the difficult instances are so specific that nearly all similar instances are not very difficult. This has been successfully applied to e.g. the simplex algorithm [30] and the 2-OPT heuristic for the traveling salesman problem [22].

Another approach to better understand and analyze NP-hard problems is to constrain which instances are allowed. Examples of this approach are looking only at planar graphs instead of all graphs, or looking at the traveling salesman problem with metric distances. This yields useful results, but is generally not applicable in practice since real-world instances do not have those properties. A more recent approach is to choose a assumption on the allowed instances which is true, close to true or often true in practice [9]. An example where this approach was used is for MAX-CUT, where instances which are stable under perturbation can be solved efficiently [8]. Graph coloring is efficiently solvable if you assume that adding a couple of edges at arbitrary places in the graph does not increase its chromatic number [21]. The traveling salesman problem becomes easier if its optimal solution is some factor better than non-optimal solutions [26]. The most research with these kinds of assumptions has been done on the (k-means) clustering problem, with several approaches [1, 4, 5, 6, 7, 29].

In the k-means clustering problem you have a set of points which you have to divide into k clusters. Equivalently, you can choose k cluster centers, as these directly correspond to clusters, and vice versa. The goal is to find clusters that minimize the sum of distances from point to its assigned cluster center. Awasthi et al. looked at γ-perturbation stability for clustering: if you assume that the k-means instance has the same optimal solution after arbitrary perturbations of at most a factor γ, then finding this solution is relatively easy [4].

As the facility location problem can be seen as a variant of the k-means clustering problem, we will use a γ-perturbation resilience assumption similar to Awasthi et al. in this thesis. It seems likely that many real-world instances already satisfy this γ-perturbation resilience, for small γ. In this thesis we will see how the γ-perturbation resilience assumption makes the facility location problem easier.

1.3 Contributions made in this thesis

In this thesis we first look at the facility location problem in general, before adding the γ-perturbation resilience assumption and seeing what basic consequences this has (Section 2).

Then we look at several reasons why the facility location problem becomes somewhat easier with γ-perturbation resilience. The first such reason is that local search algorithms will always find the optimal solution for γ-perturbation resilient instances with $\gamma \geq 3$. On the flip side, there exist $(3-\varepsilon)$-perturbation resilient instances that have local minima not equal to the optimal solution (Section 3). Several greedy algorithms have similar problems for small γ, as we will show: these greedy algorithms make mistakes while determining the solution, even for γ-perturbation resilient instances with relatively high γ (Section 4). This turns out to happen even for one of the currently best performing approximation algorithms, the Jain-Mahdian-Saberi algorithm [17]. This is a surprise, since a connection between γ-perturbation resilience and approximation algorithms seems obvious. We prove that this connection is unfounded with a counterexample (Section 5).

We also look at how hard the facility location problem is with γ-perturbation resilience, and found that if the ability to solve the perturbation resilient facility location problem for small γ exists, it implies that RP = NP (Section 6).

2 The facility location problem

The facility location problem has several variants. The variant we use in this thesis is usually called the uncapacitated metric facility location problem:

Definition. *The* uncapacitated metric facility location problem *(referred to as the* facility location problem *or the* FLP *in this thesis) is the following optimization problem.*

***Instance**: an instance (F, D, f, c) of the FLP consists of a finite set of locations F, a finite set of customers D, facility costs $f_i \geq 0$ for all $i \in F$ and service costs $c_{ij} \geq 0$ for all $i \in F, j \in D$. The service costs are metric, i.e. $c_{ij} \leq c_{i'j} + c_{i'j'} + c_{ij'}$ for all $i, i' \in F, j, j' \in D$.*

***Solutions**: a solution (X, σ) for an instance (F, D, f, c) of the FLP consists of a nonempty set of open facilities $X \subseteq F$ and a customer assignment $\sigma : D \to X$ to open facilities.*

***Objective**: The cost of a solution is $c(X, \sigma) = \sum_{i \in X} f_i + \sum_{j \in D} c_{\sigma(j)j}$. The objective is to minimize $c(X, \sigma)$.*

This variant is called uncapacitated since all facilities can handle an arbitrary number of customers and it is called metric because the service costs satisfy an extension of the triangle inequality.

The optimal solution to an FLP instance is usually denoted as (X^*, σ^*). The solution costs of the FLP, $c(X, \sigma)$, can be split into two parts: the facility costs $c_F(X) = \sum_{i \in X} f_i$ and service costs $c_S(X, \sigma) = \sum_{j \in D} c_{\sigma(j)j}$. Additionally, given an instance of the FLP and a set of open facilities X, it is easy to compute an optimal corresponding customer assignment σ: $\sigma(j) = \arg\min_{i \in X} c_{ij}$, breaking ties arbitrarily. Thus, the customer assignment is often dropped in the cost notations, which implies that an optimal assignment is used.

Besides the metric FLP, as defined above, we sometimes use the non-metric version of the FLP in this thesis:

Definition. *The* uncapacitated facility location problem *(referred to as the* non-metric FLP *in this thesis) is identical to the FLP except for one detail: an instance (F, D, f, c) of the non-metric FLP has service costs which need not be metric, i.e. $c_{ij} \not\leq c_{i'j} + c_{i'j'} + c_{ij'}$ is allowed.*

Thus, an instance of the FLP is also an instance of the non-metric FLP.

2.1 Results on the facility location problem

There is a decision variant of the FLP, where for a given instance (F, D, f, c) and bound L the question is whether the optimal solution X^* has cost $c(X^*) \leq L$. This decision variant of the FLP is NP-complete, as we will show in the following theorem and proof (for more information, see Krarup et al. [20]).

Theorem 1. *The decision variant of the FLP is NP-complete.*

Proof. First, we show that the decision variant of the FLP is in NP. For an arbitrary given instance (F, D, f, c) and bound L, choose an arbitrary nonempty set $X \subseteq F$ nondeterministically. Now check if the cost $c(X) \leq L$. If so, output "yes", otherwise, output "no".

If the instance has $c(X^*) \leq L$, this will output "yes", since $X^* \subseteq F$ and X^* is nonempty, so it will be one of the nondeterministic choices. If the instance has $c(X^*) > L$, then because $c(X) \geq c(X^*) > L$ no nondeterministic choice X can yield "yes", so the answer will be "no". Thus, the decision variant of the FLP is in NP.

Now, to show that the decision variant of the FLP is NP-hard, we reduce the set-covering problem to the FLP. Because the decision variant of the set-covering problem is NP-complete [19], this will prove that the decision variant of the FLP is NP-hard.

The instances (U, S) of the set-covering problem consist of the universe U and a collection of subsets $S = \bigcup S_i$, where each S_i is a subset of U. The goal of the set-covering problem is to find the minimum cardinality set $C \subseteq S$ that covers U, i.e. $\{x \mid x \in S_i, S_i \in C \text{ for some } i\} = U$.

Given an instance (U, S) and bound k, construct the following FLP instance and bound:

$$
\begin{aligned}
F &= S, \\
D &= U, \\
f_i &= 1, \\
c_{ij} &= \begin{cases} k & \text{if } j \in i, \text{ and} \\ 3k & \text{otherwise,} \end{cases} \\
L &= k + k|D|.
\end{aligned}
$$

Clearly, if (U, S) is a yes-instance, so is (F, D, f, c), by opening facilities $X = C$, yielding $c(X) = |C| + k|D| \leq k + k|D| = L$. If (F, D, f, c) is a yes-instance, some X has $c(X) \leq L = k + k|D|$, so $c_{\sigma(j)j} = k$ for every $j \in D$. Thus, $C = X$ is a set cover with $|C| = c(X) - k|D| \leq k$ and (U, S) is a yes-instance. As a result, this is a valid reduction, proving that the decision variant of the FLP is NP-hard. □

As a consequence of this theorem, the (search variant of the) FLP is NP-hard.

NP-hard optimization problems often still have approximation algorithms which run in deterministic polynomial time, which give a solution with cost at most a factor $\alpha > 1$ times the optimal cost. Depending on the problem, there may be a lower bound for α, below which the problem is still NP-hard. The following three theorems show the current best approximation hardness results for the FLP.

Theorem 2 (Guha and Kuller 1999 [16]). *Approximating the FLP within a factor $\alpha = 1.463$ is NP-hard.*

Theorem 3 (Byrka and Aardal 2010 [10]). *The FLP can be approximated within a factor $\alpha = 1.5$ by a deterministic polynomial time algorithm.*

Theorem 4 (Li 2013 [23]). *The FLP can be approximated with a expected factor $\alpha = 1.488$ by a randomized polynomial time algorithm, i.e. for any possible instance, the algorithm computes solutions X such that $\mathbb{E}[c(X)] \leq 1.488c(X^*)$.*

These results are not particularly promising if you want to find a good solution to your FLP instance, since solutions 50% more expensive than the optimal solution does not seem very attractive. However, in practice several of the best approximation algorithms do a lot better. For example, the Jain-Mahdian-Saberi algorithm, with an approximation ratio $\alpha = 1.61$, achieves a ratio of 1.03 on average on real-world instances [17]. This difference between the observed performance and guaranteed performance leads us to making an assumption about real-world instances, as we show in the next subsection.

2.2 Perturbation resilience

Seeing how the FLP is an NP-hard problem [20], it is only natural to look for conditions on the instances which make the problem easier. Based on ideas by Bilu et al. [8] and Balcan et al. [5], we want to make an assumption which many real-world instances already satisfy. The perturbation resilience assumption for clustering problems by Awasthi et al. [4], adapted to the FLP, looks like a good choice. First, we need to define what it means to perturb an FLP instance.

Definition. *An instance (F, D, f', c') of the non-metric FLP is a* γ-perturbation *of instance (F, D, f, c) of the FLP, with $\gamma \geq 1$, iff $f_i \leq f'_i \leq \gamma f_i$ for all $i \in F$ and $c_{ij} \leq c'_{ij} \leq \gamma c_{ij}$ for all $i \in F, j \in D$. If it is clear from the context which F and D are used, a γ-perturbed instance can also be denoted as (f', c').*

A γ-perturbed instance is any instance that is a γ-perturbation of some FLP instance.

Note that a γ-perturbed instance might or might not be metric. This definition only allows greater costs than the original instance, which might seem weird for a perturbation, but this is simply a matter of scaling. An FLP instance remains equivalent under scaling all costs with the same value, so an equivalent definition of γ-perturbation also allowing lower costs would, for instance, be: $f_i \leq \sqrt{\gamma} f'_i \leq \gamma f_i$ and similar for the service costs.

Using this definition, we can define γ-perturbation resilience for the FLP.

Definition. *An instance (F, D, f, c) of the FLP is* γ-perturbation resilient *with $\gamma \geq 1$ iff all γ-perturbations (f', c') of (F, D, f, c) have the same unique optimal solution (X^*, σ^*).*

For small γ (i.e. $\gamma < 1.05$ or similar), it seems likely that many real-world instances satisfy this assumption, since stability of solutions makes a lot of sense: if an instance is very sensitive to perturbations, then it means that the exact solution probably does not matter much. For small γ, it also implies that the instance is resilient in the face of small changes to the costs. In practice, this often happens too: you are not going to change the location of your warehouse if it is 3% farther away than previously estimated. Nevertheless, we will consider all values γ in this thesis. In the conclusion (section 7) we will come back to the results and see if they are applicable in practice.

If costs of γ-perturbed instances are compared, the notation of the perturbed costs follow from the names given to the perturbed instances. So, for example, if (f', c') is a γ-perturbed instance, then $c'(X)$ denotes the cost of solution X in the perturbed instance.

Any γ-perturbation resilient instance of the FLP is a valid γ-perturbation of itself, so γ-perturbation resilience implies that the original instance has the same optimal solution as any of its perturbations. Note that, since γ-perturbation resilience requires an unique solution, there are instances of the FLP which are not γ-perturbation resilient for any γ, namely exactly those instances with multiple optimal solutions.

The definition of γ-perturbation resilience, although intuitive, is somewhat hard to work with since it requires checking the optimal solution for all γ-perturbations of an FLP instance. The following theorem introduces an equivalent but easier to check definition.

Theorem 5. *Consider, for some instance (F, D, f, c) of the FLP with optimal solution X^*, customer assignment σ^* and nonzero optimal cost $c(X^*)$, the following γ-perturbation:*

$$f'_i = \begin{cases} \gamma f_i & \text{if } i \in X^*, \text{ and} \\ f_i & \text{otherwise,} \end{cases}$$

$$c'_{ij} = \begin{cases} \gamma c_{ij} & \text{if } \sigma^*(j) = i, \text{ and} \\ c_{ij} & \text{otherwise.} \end{cases}$$

This γ-perturbed instance (F, D, f', c') has the same optimal solution X^* and customer assignment σ^* iff the instance (F, D, f, c) is γ-perturbation resilient. If so, the solution (X^*, σ^*) is unique for both (F, D, f, c) and (F, D, f', c').

Proof. If the instance is γ-perturbation resilient, then the given γ-perturbation has the same unique optimal solution, by definition of γ-perturbation resilience.

Thus, assume that the solution (X^*, σ^*) is also the solution to the given (F, D, f', c') instance. Consider all possible γ-perturbations $(\tilde{f}, \tilde{c})$ of the (F, D, f, c) instance and construct the following γ-perturbation:

$$\tilde{f}'_i = \begin{cases} \tilde{f}_i & \text{if } i \in X^*, \text{ and} \\ f_i & \text{otherwise,} \end{cases}$$
$$\tilde{c}'_{ij} = \begin{cases} \tilde{c}_{ij} & \text{if } \sigma^*(j) = i, \text{ and} \\ c_{ij} & \text{otherwise.} \end{cases}$$

By construction, it holds that $\tilde{c}(X^*, \sigma^*) = \tilde{c}'(X^*, \sigma^*)$ and $\tilde{c}'(X, \sigma) \leq \tilde{c}(X, \sigma)$ for all nonempty open facility sets $X \subseteq F$ and customer assignments $\sigma : D \to X$. Now relate $c'(X^*, \sigma^*)$ to $\tilde{c}'(X^*, \sigma^*)$ and $c'(X, \sigma)$ to $\tilde{c}'(X, \sigma)$:

$$c'(X^*, \sigma^*) = \tilde{c}'(X^*, \sigma^*) + \sum_{i \in X^*} \gamma f_i - \tilde{f}_i + \sum_{j \in D} \gamma c_{\sigma^*(j)j} - \tilde{c}_{\sigma^*(j)j}, \tag{1}$$
$$c'(X, \sigma) = \tilde{c}'(X, \sigma) + \sum_{i \in X \cap X^*} \gamma f_i - \tilde{f}_i + \sum_{\substack{j \in D \\ \sigma(j) = \sigma^*(j)}} \gamma c_{\sigma^*(j)j} - \tilde{c}_{\sigma^*(j)j}. \tag{2}$$

By assumption $c'(X^*, \sigma^*) < c'(X, \sigma)$. So $\tilde{c}'(X^*, \sigma^*) < \tilde{c}'(X, \sigma)$, because the sum terms in equation (1) are at most the sum terms in equation (2). As noted above, by construction it follows that $\tilde{c}(X^*) < \tilde{c}(X)$.

This holds for all γ-perturbations $(\tilde{f}, \tilde{c})$ of (F, D, f, c) and all solutions (X, σ). Because $c(X^*) \neq 0$ the solution is unique, proving that the instance (F, D, f, c) is γ-perturbation resilient. □

We now have two equivalent definitions for γ-perturbation resilience, so it is useful to know for which range of γ γ-perturbation resilient instances exist. FLP instances need to be metric, so this is not a trivial fact. The following two theorems establish that γ-perturbation resilient instances exist for all $\gamma \geq 1$.

Theorem 6. *If an FLP instance (F, D, f, c) is γ-perturbation resilient, it also is γ'-perturbation resilient for all $\gamma' \in \mathbb{R}$ which satisfy $1 \leq \gamma' \leq \gamma$.*

Proof. Let (F, D, f, c) be any possible γ-perturbation resilient FLP instance, and $\gamma' \in \mathbb{R}$ be an arbitrary number which satisfies $1 \leq \gamma' \leq \gamma$. Now consider all γ' perturbations (f', c') of (F, D, f, c). Since $\gamma' \leq \gamma$, (f', c') is a valid γ-perturbation of (F, D, f, c). By definition of γ-perturbation resilience, this implies that the same solution is the unique optimal solution to both (F, D, f, c) and (f', c').

This satisfies the requirement for γ'-perturbation resilience, proving the theorem. □

Theorem 7. *For all $x \in \mathbb{R}$, there exist γ-perturbation resilient instances of the FLP with $\gamma > x$.*

Proof. Assume to the contrary that there is some x such that all instances of the FLP are γ-perturbation resilient with $\gamma < x$. Assume that $x > 1$, which can be done without loss of generality. Now consider the following instance (see Figure 1):

$$
\begin{aligned}
F &= \{1_f, 2_f\}, \\
D &= \{1_d, 2_d\}, \\
f_i &= 0, \\
c_{ij} &= \begin{cases} 1 & \text{if } i = k_f, j = k_d \text{ for some } k, \text{ and} \\ 2x & \text{otherwise.} \end{cases}
\end{aligned}
$$

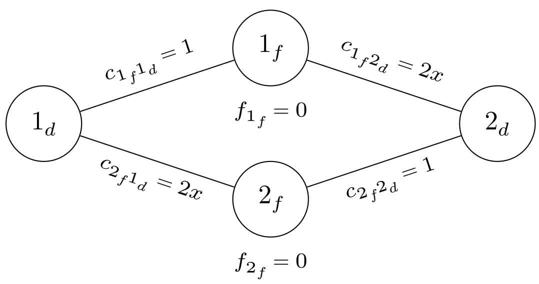

Figure 1: Example of Theorem 7

This instance has three possible solutions, with the following costs:

$$
\begin{aligned}
c(\{1_f\}) &= 2x + 1, \\
c(\{2_f\}) &= 2x + 1, \\
c(\{1_f, 2_f\}) &= 2.
\end{aligned}
$$

Since all γ-perturbations increase the cost of a solution X at most by a factor γ, a trivial lower bound for the perturbation resilience γ' of this instance can be calculated as follows: $\gamma' = \frac{2x+1}{2} = x + \frac{1}{2}$. This is a contradiction to the assumption that all instances of the FLP are γ-perturbation resilient with $\gamma < x$, since $\gamma' > x$. □

For most NP-hard problems, it is trivial to find the optimal solution to an instance once you have an algorithm that solves the decision variant of the problem (i.e. "is the cost of the optimal solution below a given bound?"). The following theorem looks at the decision variant of the FLP for γ-perturbation resilient instances.

Theorem 8. *Assume a deterministic algorithm* `A` *exists that decides $g(|F|, |D|)$-perturbation resilient FLP instances (F, D, f, c) in polynomial time, for some function g. Then a deterministic algorithm exists that finds the optimal solution X^* for γ'-perturbation resilience FLP instances (F, D, f, c) with $\gamma' > g(|F|, |D|)$ in polynomial time.*

This theorem implies that, like for many other NP-hard problems, discovering the value of a γ-perturbation resilient instance is not more difficult than finding the corresponding solution. The only exception, where this may or may not be true, is the border case of exactly $g(|F|, |D|)$-perturbation resilient instances. Although a theorem like this for general FLP can easily be proven using self-reducibility, this is not true here. Self-reducibility can destroy the γ-perturbation resilience of an instance, and as result the assumed existing algorithm cannot be applied to the resulting instances.

Proof. Assume algorithm `A` exists for some function g. Now, take an arbitrary γ'-perturbation resilient instance (F, D, f, c) with $\gamma' > g(|F|, |D|)$. First, calculate the exact optimal cost $c(X^*)$ using algorithm `A`, which can be done in polynomial time using a binary search with calls to `A`.

Because $\gamma' > g(|F|, |D|)$, it is possible to $\frac{\gamma'}{g(|F|,|D|)}$-perturb the instance (F, D, f, c) to some instance (F, D, f', c'). Because this is a valid γ'-perturbation, both instances yield the same optimal solution X^* and customer assignment σ^*. Note that all $g(|F|, |D|)$-perturbations of instance (F, D, f', c') are valid γ'-perturbations of (F, D, f, c). Thus, algorithm `A` can be used on all such instances (F, D, f', c'), since they are $g(|F|, |D|)$-perturbation resilient, and use the same optimal solution X^* to decide if it is a yes- or no-instance.

So, consider the following $\frac{\gamma'}{g(|F|,|D|)}$-perturbation, for some fixed facility $i' \in F$, which will be chosen later:

$$f'_i = \begin{cases} \frac{\gamma'}{g(|F|,|D|)} f_i & \text{if } i = i', \text{ and} \\ f_i & \text{otherwise.} \end{cases}$$

This perturbed instance (F, D, f', c) has cost $c'(X^*) = c(X^*)$ iff $i' \notin X^*$. So by running algorithm `A` on instance (F, D, f', c) with threshold $c(X^*)$, a part of the optimal solution X^* is discovered, if $f_{i'} = 0$. If $f_{i'} = 0$, we know that including i' in X^* never results in a higher cost. Doing this for all facilities $i' \in F$ yields the optimal solution X^*.

All parts of this algorithm run in polynomial time and use algorithm `A` a polynomial number of times, which completes the proof. □

3 Local search

Local search techniques are a common way to solve problems. In a local search algorithm, we always start from some initial solution, like a trivial solution (e.g. $X = F$ for the FLP) or a solution computed by a heuristic. From there, local search tries to improve its current solution. It does that by computing the neighbourhood of the current solution, a set of solutions which are similar to the current solution. What this neighbourhood is differs per problem and can be chosen by the creator of the local search algorithm. The local search algorithm now computes the cost of each solution in the neighbourhood. If at least one is better than the current solution, it updates its current solution to one of the better solutions. If none are, the local search algorithm terminates. All solutions which are at least as good as all their neighbouring solutions are called local minima.

Local search techniques are commonly used in practice, since they are easy to implement: they only require you to define a neighbourhood, which is quite simple. Additionally, real-world problems often have extra constraints on the solution, which do not correspond to the theoretical problems. An example for the FLP would be that there are a few facilities that, when open, must have exactly three assigned customers. These extra constraints are not difficult to add in a local search algorithm.

In this section we look at local minima and local search algorithms in general, and see how γ-perturbation resilient instances impact them.

3.1 Existence of local minima

In this subsection we will show that 3-perturbation resilient instances do not have local minima except for the global minimum. We also show that there are $(3-\varepsilon)$-perturbation resilient instances for which this is not true, i.e. that there are local minima unequal to the optimal solution. To do so, we first define what a local minimum is for the FLP.

Definition. *A solution X to an instance of the FLP is a local minimum iff*

- $c(X \setminus \{i\}) \geq c(X)$ *for all* $i \in X$ *with* $|X| \geq 2$ *(dropping a facility), and*
- $c(X \cup \{i\}) \geq c(X)$ *for all* $i \in F \setminus X$ *(adding a facility), and*
- $c(X \setminus \{i\} \cup \{j\}) \geq c(X)$ *for all* $i \in X, j \in F \setminus X$ *(swapping a facility).*

This definition means that the local minimum X cannot be improved by opening a new facility, closing an open facility or switching between an open and closed facility.

The following theorem is a result for local search algorithms on any FLP instance.

Theorem 9 (Arya et al. 2004 [3]). *All local minima X for an instance of the FLP satisfy* $c_F(X) \leq c_F(X^*) + 2c_S(X^*)$ *and* $c_S(X) \leq c_F(X^*) + c_S(X^*)$*, where X^* is the optimal solution for the instance. Combined, this yields* $c(X) \leq 3c(X^*)$*.*

This theorem will be used to show that there are no local minima except for the optimal solution for 3-perturbation resilient FLP instances (Theorem 11). To prove this, we first need an extra result from the following lemma. This lemma means that, if some γ-perturbation resilient instance exists, then also such a γ-perturbation resilient instance with more restrictions exists: you can see this as removing everything from an instance that is not needed to remain γ-perturbation resilient or have a local minimum X besides the optimal solution.

Lemma 10. *Assume a γ-perturbation resilient instance (F, D, f, c) of the FLP exists with a local minimum (X, σ) not equal to the optimal solution (X^*, σ^*).*

Then an instance (F', D', f', c') of the FLP exists with the following properties:

- *the instance (F', D', f', c') is γ-perturbation resilient, and*
- *the instance (F', D', f', c') has a local minimum (X', σ') not equal to the optimal solution (X'^*, σ'^*), and*
- *$F' = X' \cup X'^*$, and*
- *$f'_i = 0$ for all $i \in X' \cap X'^*$, and*
- *for all $j \in D'$, ($\sigma'(j) \notin X' \cap X'^*$ or $\sigma'^*(j) \notin X' \cap X'^*$).*

Note that the optimal solution (X^*, σ^*) to any instance is always a local minimum, so this is explicitly excluded in the lemma.

Proof. Take any such instance (F, D, f, c). We transform the instance (F', D', f', c') using the following steps. We apply each step until its conditions are satisfied for all local minima $X \neq X^*$, possibly applying earlier steps again in the process if their conditions are not valid any more after applying a later step. Thus, at the beginning of every step, the conditions of all previous steps are satisfied.

Step 1. *Condition to satisfy: $F = X \cup X^*$.*
Drop all facilities not in X or X^*, i.e. $F' = X \cup X^*$. All customer assignments of σ and σ^* remain valid and the costs $c(X)$ and $c(X^*)$ are unchanged. Thus, the resulting instance (F', D, f, c) is still γ-perturbation resilient with optimal solution X^*. Because all subsets of F' also are a subset of F, $X \neq X^*$ is still a local minimum. After doing this, the conditions of step 1 are satisfied.

Step 2. *Condition to satisfy: $f_i = 0$ for all $i \in X \cap X^*$.*
Change the facility costs to the following:

$$\tilde{f}_i = \begin{cases} 0 & \text{if } i \in X \cap X^*, \text{ and} \\ f_i & \text{otherwise.} \end{cases}$$

Note that the resulting instance $(F, D, \tilde{f}, c)$ still has X as a local minimum, since the cost of adding a facility to X is identical, compared to instance (F, D, f, c), and the cost of dropping or swapping a facility from X is the equal or higher, as compared to instance (F, D, f, c). To show that the instance $(F, D, \tilde{f}, c)$ is γ-perturbation resilient with optimal solution X^*, consider all nonempty sets of open facilities $Y \subseteq F$ and customer assignments σ' in all perturbations of costs f'_i and c'_{ij} and equivalent perturbations of $\tilde{f}_i$ and c_{ij}:

$$\begin{aligned} c'(X^*, \sigma^*) &= c'_F(X^*) + c'_S(X^*, \sigma^*) \\ &= \sum_{i \in X^* \setminus X} f'_i + \sum_{i \in X \cap X^*} f'_i + c'_S(X^*, \sigma^*) \\ &= \tilde{c}'(X^*, \sigma^*) + \sum_{i \in X \cap X^*} f'_i, \\ c'(Y, \sigma') &= c'_F(Y) + c'_S(Y, \sigma') \\ &= \sum_{i \in Y \setminus (X \cap X^*)} f'_i + \sum_{i \in Y \cap X \cap X^*} f'_i + c'_S(Y, \sigma') \\ &= \tilde{c}'(Y, \sigma') + \sum_{i \in Y \cap X \cap X^*} f'_i. \end{aligned}$$

This implies that

$$\begin{aligned}
\tilde{c}'(X^*,\sigma^*) + \sum_{i\in X\cap X^*} f'_i &\leq \tilde{c}'(Y,\sigma') + \sum_{i\in Y\cap X\cap X^*} f'_i,\\
\tilde{c}'(X^*,\sigma^*) + \sum_{i\in (X\cap X^*)\setminus Y} f'_i &\leq \tilde{c}'(Y,\sigma'),\\
\tilde{c}'(X^*,\sigma^*) &\leq \tilde{c}'(Y,\sigma').
\end{aligned}$$

So the instance $(F, D, \tilde{f}, c)$ is indeed still γ-perturbation resilient. This satisfies the conditions for step 2.

Step 3. *Condition to satisfy: for all $j \in D$, $(\sigma(j) \notin X \cap X^*$ or $\sigma^*(j) \notin X \cap X^*)$ must be true.* Choose an arbitrary $j \in D$ with $\sigma(j) \in X \cap X^*$ and $\sigma^*(j) \in X \cap X^*$. By the definition of γ-perturbation resilience, j cannot be assigned to any other facility in X^* in all of the γ-perturbed costs c'. Thus $c_{\sigma^*(j)j} < \gamma c_{ij}$ for all $i \in X^* \setminus \{\sigma^*(j)\}$. Also, because $F = X \cap X^*$ and X is a local minimum, $c_{\sigma(j)j} \leq c_{ij}$ for $i \in F \setminus X^* = X \setminus X^*$. Thus, the assignment $\sigma(j) = \sigma^*(j)$ is the best possible assignment in F.

Let $\tilde{c}$ denote the costs in instance (F, D', f, c).

The new instance (F, D', f, c) is created by removing customer j, i.e. $D' = D \setminus \{j\}$. As a result, X still is a local minimum in (F, D', f, c):

- Dropping a facility $i \in X$ (if $|X| > 2$); If $i = \sigma(j)$, $f_i = 0$, so $\tilde{c}(X \setminus \{i\}) \geq \tilde{c}(X)$. If $i \neq \sigma(j)$, then customer j is not connected to facility i, so $\tilde{c}(X) - \tilde{c}(X \setminus \{i\}) = c(X) - c(X \setminus \{i\}) \geq 0$. Thus, dropping facility $i \in X$ does not result in a better solution.
- Adding a facility $i \in F \setminus X$; Removing customer j does not change the cost of adding a facility, since $\sigma^*(j) = \sigma(j)$ and $c_{\sigma(j)j} \leq c_{ij}$: $\tilde{c}(X) - \tilde{c}(X \cup \{i\}) = c(X) - c(X \cup \{i\}) \geq 0$. Thus, adding facility $i \in F \setminus X$ does not result in a better solution.
- Swapping an open facility $i \in X$ with closed facility $i' \in F \setminus X$; If $i \neq \sigma(j)$, the same reasoning as in adding a facility holds. If $i = \sigma(j)$, $f_i = 0$ and this swap is not better than just adding facility i', which did not improve the cost either. Thus, swapping open facility $i \in X$ with closed facility $i' \in F \setminus X$ does not result in a better solution.

To show that the new instance is γ-perturbation resilient with optimal solution (X^*, σ^*), consider all $Y \subseteq F$ and γ-perturbations (f', c'). Let $Y' = Y \cup (X \cap X^*)$ with optimal assignment (i.e. $\sigma'(x) = \arg\min_{i\in Y'} c'_{ix}$) and note that $c'(Y') \leq c'(Y)$ since $f'_i = 0$ for all $i \in X \cap X^*$, even in the new instance (F, D, f', c'). Let $\tilde{c}'$ denote the costs in instance (F, D', f', c'). Thus:

$$\begin{aligned}
c'(X^*,\sigma^*) &= c'_F(X^*) + c'_S(X^*,\sigma^*)\\
&= c'_F(X^*) + c'_{\sigma^*(j)j} + \sum_{x\in D'} c'_{\sigma^*(x)x}\\
&= \tilde{c}'(X^*) + c'_{\sigma^*(j)j}, \qquad (3)\\
c'(Y',\sigma') &= c'_F(Y') + c'_S(Y',\sigma')\\
&= c'_F(Y') + c'_{\sigma'(j)j} + \sum_{x\in D'} c'_{\sigma'(x)x}\\
&= \tilde{c}'(Y',\sigma) + c'_{\sigma'(j)j}. \qquad (4)
\end{aligned}$$

Note that because $\sigma^*(j) \in Y'$, $c'_{\sigma'(j)j} \leq c'_{\sigma^*(j)j}$. This and inequalities (3) and (4) imply

$$\begin{aligned}
\tilde{c}'(X^*,\sigma^*) + c'_{\sigma^*(j)j} &\leq \tilde{c}'(Y',\sigma') + c'_{\sigma'(j)j},\\
\tilde{c}'(X^*,\sigma^*) &\leq \tilde{c}'(Y',\sigma') \leq \tilde{c}'(Y),
\end{aligned}$$

so even after removal of customer j, the instance (F, D', f, c) is γ-perturbation resilient with optimal solution X^*. After doing this a couple of times, the condition for step 3 is satisfied.

After step 3, all conditions required for the lemma are satisfied. Note that all steps make the instance smaller in some way (less facilities, less facilities with nonzero cost, less customers), so this process terminates eventually. $\square$

This lemma can be interpreted as removing complications from the instance, except those which are necessary for either γ-perturbation resilience or the existence of the local minimum X. The following theorem uses Lemma 10 to show that any local minimum of 3-perturbation resilient FLP instances always is the global minimum.

Theorem 11. *All local minima (X, σ) of a γ-perturbation resilient instance (F, D, f, c) of the FLP with $\gamma \geq 3$ are equal to the optimal solution (X^*, σ^*) of the instance.*

Proof. Assume the contrary and use Lemma 10 to get an instance (F, D, f, c) with $F = X \cup X^*$, $f_i = 0$ for $i \in X \cap X^*$ and, for all $j \in D$, $(\sigma(j) \notin X \cap X^*$ or $\sigma^*(j) \notin X \cap X^*)$. Here (X^*, σ^*) is the optimal solution and $(X, \sigma) \neq (X^*, \sigma^*)$ the local minimum.

We perturb the costs as follows:

$$f'_i = \begin{cases} 0 & \text{if } i \in X \cap X^*, \text{ and} \\ 3f_i & \text{if } i \in X^* \setminus X, \text{ and} \\ f_i & \text{otherwise,} \end{cases}$$
$$c'_{ij} = \begin{cases} 3c_{ij} & \text{if } \sigma^*(j) = i, \text{ and} \\ c_{ij} & \text{otherwise.} \end{cases}$$

This is a valid γ-perturbation. Because the instance (F, D, f, c) is γ-perturbation resilient, the following holds:

$$\begin{aligned} c'(X^*, \sigma^*) &= c'_F(X^*) + c'_S(X^*, \sigma^*) \\ &= \sum_{i \in X^* \setminus X} 3f_i + \sum_{j \in D} 3c_{\sigma^*(j)j} && \text{(by choice of } f'_i \text{ and } c'_{ij}) \\ &= 3c(X^*, \sigma^*), \\ c'(X, \sigma) &= c_F(X) + c_S(X, \sigma) \\ &= \sum_{i \in X \setminus X^*} f_i + \sum_{j \in D} c_{\sigma(j)j}, && \text{(by the properties of } (F, D, f, c),\ f'_i \text{ and } c'_{ij}) \\ &= c(X, \sigma), \end{aligned}$$

so

$$3c(X^*, \sigma^*) < c(X, \sigma).$$

By Theorem 9, $c(X, \sigma) \leq 3c(X^*, \sigma^*)$, so $3c(X^*) < 3c(X^*)$ which is a contradiction. Thus, for γ-perturbation resilient instances of the FLP with $\gamma \geq 3$, no local minima exist except for the global optimum. $\square$

As a result of Theorem 11, there is an unique local, and thus global, minimum for all γ-perturbation resilient instances with $\gamma \geq 3$. Theorem 11 cannot be improved upon, as we will show in the following theorem.

Theorem 12. *There exist γ-perturbation resilient instances of the FLP for all $\gamma < 3$ with local minima $X \neq X^*$.*

Note that the local minimum $X = X^*$ always exists, as it is the global minimum.

Proof. The following example, inspired by Arya et al. [3], proves the theorem (see also Figure 2). Choose a $k \in \mathbb{N}$, whose value will be specified later.

$$
\begin{aligned}
F &= \{0_f, 1_f, \ldots, (k-1)_f, k_f, \xi_f\}, \\
D &= \{0_d, \ldots, k_d\}, \\
f_i &= \begin{cases} 2k & \text{if } i = \xi_f, \text{ and} \\ \frac{1}{k} & \text{otherwise,} \end{cases} \\
c_{ij} &= \begin{cases} 1 & \text{if } i = \xi_f, \text{ and} \\ 1 & \text{if } i = k_f, j = k_d \text{ for some } k, \text{ and} \\ 3 & \text{otherwise.} \end{cases}
\end{aligned}
$$

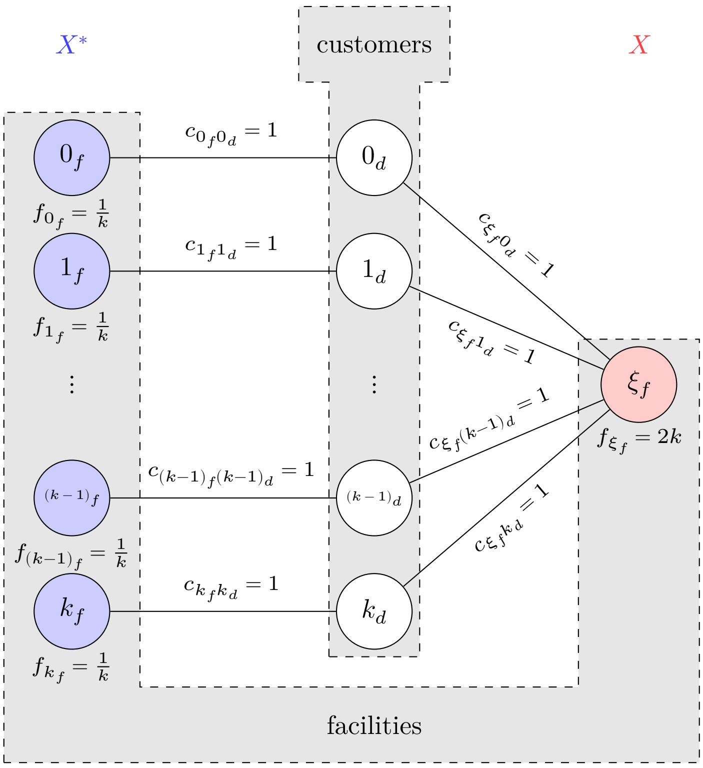

Figure 2: Example used in Theorem 12 (indirect service costs not shown)

The optimal solution X^* is $\{0_f, \ldots, k_f\}$, with $c(X^*, \sigma^*) = (k+1)(1 + \frac{1}{k})$. This instance is $\frac{3k}{(k+1)(1+1/k)}$-perturbation resilient, as we will show by comparing all solutions $X \neq X^*$ with X^* for all γ-perturbations (F, D, f', c'). By letting k go to infinity, the perturbation resilience comes arbitrarily close to 3, as required by the theorem.

Consider two cases: $\xi_f \in X$ and $\xi_f \notin X$. For the first case, assume $\xi_f \in X$. If so, $c(X, \sigma) \geq 3k + 1$. Thus, $c'(X^*, \sigma^*) \leq 3k < 3k + 1 \leq c'(X, \sigma)$, completing the first case.

The second case is when $\xi_f \notin X$. Without loss of generality, let $X = \{0_f, \ldots, (|X|-1)_f\}$. Since $c'_{i_f i_d} < 3 \leq c'_{i' i_d}$ for $i \in \{0, \ldots |X|-1\}, i' \in F \setminus \{i_f\}$, all customers $\{0_d, \ldots, (|X|-1)_d\}$ are assigned to the same facility in both X^* and X, i.e. $\sigma^*(i_d) = \sigma(i_d) = i_f$. For the other customers j_d with $j \in \{|X|, \dots, k\}$, look at the following quantity:

$$\begin{aligned} -f'_{j_f} + c'_{\sigma(j_d)j_d} - c'_{\sigma^*(j_d)j_d} &\geq -\frac{3}{(k+1)(1+\frac{1}{k})} + 3 - \frac{3k}{(k+1)(1+\frac{1}{k})} \\ &= \frac{3}{k+1} \\ &> 0. \end{aligned}$$

Using this quantity, it is possible to bound the difference between $c'(X^*, \sigma^*)$ and $c'(X, \sigma)$:

$$c'(X, \sigma) - c'(X^*, \sigma^*) = \sum_{j=|X|}^{k} c'_{\sigma(j_d)j_d} - c'_{\sigma^*(j_d)j_d} - f'_{j_f} > 0,$$

so $c'(X^*, \sigma^*) < c'(X, \sigma)$, completing the second case.

For both cases it holds that $c'(X^*, \sigma^*) < c'(X, \sigma)$, for all solutions $X \neq X^*$, so instance (F, D, f, c) is $\frac{3k}{(k+1)(1+1/k)}$-perturbation resilient. Also, $X = \{\xi_f\} \neq X^*$ is a local minimum, completing the proof. □

Note that the example used in this theorem has a local minimum $X = \{\xi_f\}$ with cost $c(X)$ which converges to $3c(X^*)$ as k goes to infinity. This shows that the 3-approximation bound of local search for FLP in general, as shown in Theorem 9, is tight.

3.2 Subsets of local minima

Knowing something about local minima, it is useful to relate other solutions to local minima. In the following theorem we show that solutions which are subsets of the local minima X can always be augmented with extra facilities without increasing the solution's cost. If the solution is a subset of the optimal solution X^* for a γ-perturbation resilient instance, then any addition of facilities in X^* but not in the solution will decrease the solution's cost.

Theorem 13. *For all instances (F, D, f, c) of the FLP, with any possible local minimum (X, σ_X), for all solutions Y, where Y is a strict subset of X, the following holds:*

- *For any added facility $i \in X \setminus Y$, the costs can only get lower, i.e. $c(Y) \geq c(Y \cup \{i\})$, and*
- *If $c(Y) \neq c(X)$, at least one facility $i \in X \setminus Y$ has $c(Y) > c(Y \cup \{i\})$.*

Additionally, if the instance is γ-perturbation resilient, all facilities $i \in X^* \setminus Y$ have $c(Y) > c(Y \cup \{i\})$.

Proof. Because X is a local minimum, dropping facility $i \in X \setminus Y$ leads to equal or higher costs. Thus

$$\begin{aligned} 0 &\leq -f_i + \sum_{\substack{j \in D \\ \sigma_X(j)=i}} \min_{i' \in X \setminus \{i\}} c_{i'j} - c_{ij} \qquad (5) \\ &\leq -f_i + \sum_{\substack{j \in D \\ \sigma_X(j)=i}} c_{\sigma_Y(j)j} - c_{ij} \\ &\leq -f_i + \sum_{j \in D} \max\{c_{\sigma_Y(j)j} - c_{ij}, 0\} \\ &= c(Y) - c(Y \cup \{i\}), \end{aligned}$$

so $c(Y) \geq c(Y \cup \{i\})$ for all $i \in X \setminus Y$, and $c(Y) > c(Y \cup \{i\})$ if $X = X^*$ and (F, D, f, c) is a γ-perturbation resilient instance, since inequality (5) is strict for γ-perturbation resilient instances and optimal solution X^*.

Now, assume $c(Y) \neq c(X)$ and define $g(i) = -f_i + \sum_{j \in D, \sigma_X(j)=i} c_{\sigma(j)j} - c_{ij}$. Because of the above inequalities, $0 \leq g(i) \leq c(Y) - c(Y \cup \{i\})$ for all facilities $i \in X \setminus Y$. Combining this for all such facilities:

$$\begin{aligned}
\sum_{i \in X \setminus Y} g(i) &= -\sum_{i \in X \setminus Y} f_i + \sum_{\substack{j \in D \\ \sigma_Y(j) \in X \setminus Y}} c_{\sigma_Y(j)j} - c_{\sigma_X(j)j} \\
&= -\sum_{i \in X \setminus Y} f_i + \sum_{j \in D} c_{\sigma_Y(j)j} - c_{\sigma_X(j)j} \\
&= c(Y) - c(X) \\
&> 0.
\end{aligned}$$

So at least one $i \in X \setminus Y$ has $g(i) > 0$ and thus $c(Y) > c(Y \cup \{i\})$. □

Thus, if you have a solution that is a subset of a local minimum, you can also end up in that local minimum by adding facilities while that improves the cost.

4 Greedy algorithms

In this section we will consider several greedy algorithms for γ-perturbation resilient instances. Greedy algorithms have the advantage of being efficient but they do not always give the optimal solution to an instance. The chosen greedy algorithms, greedy facility addition and greedy facility deletion, are an obvious choice for a greedy FLP algorithm because of their simplicity. These generally do not result in the optimal solution for the FLP, and we will show that this remains true for the γ-perturbation resilient FLP, even for high γ.

4.1 Greedy facility addition

One of the simplest greedy algorithms is greedy facility addition. In this algorithm, you start from some initial set of facilities X (potentially empty) and greedily add facilities until some condition is met. Here, we add the facility with maximum cost reduction (i.e. $-f_i + \sum_{j \in D} \max\{c_{\sigma(j)j} - c_{ij}, 0\}$), as long as that remains positive. This greedy algorithm seems promising, based on the results of Theorem 13.

The following $(\frac{7}{2} - \varepsilon)$-perturbation resilient FLP instance is a counterexample that shows this does not always yield the optimal solution:

$$\begin{aligned}
F &= \{1_f, 2_f, 3_f, 4_f\},\\
D &= \{1_d, 2_d, 3_d\},\\
f &= (0, 0, 0, 1)\,,\\
c &= \begin{pmatrix} 1 & 50 & 50 \\ 52 & 1 & 7 \\ 52 & 7 & 1 \\ 54 & 3 & 3 \end{pmatrix},\\
X &= \{1_f\},\\
X^* &= \{1_f, 2_f, 3_f\}.
\end{aligned}$$

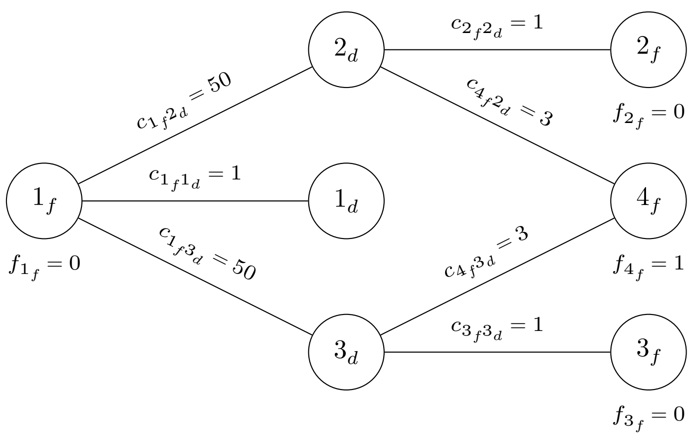

Figure 3: Counterexample to greedy facility addition (indirect service costs not shown)

In this example, the optimal solution is $X^* = \{1_f, 2_f, 3_f\}$, since out of all solutions, X^* has both minimal facility cost $c_F(X^*) = 0$ and minimal service cost $c_S(X^*) = 3$. To show that greedy facility addition does not yield X^* with this example, we start from $X = \{1_f\}$. This may not seem reasonable, but just adding a lot of copies of customer 1_d to the example makes this a good choice when starting from the empty set. In the next step, several facilities can be added, with the following cost reduction:

$$\begin{aligned}
&c(\{1_f\}) - c(\{1_f, 2_f\}) = 92,\\
&c(\{1_f\}) - c(\{1_f, 3_f\}) = 92,\\
&c(\{1_f\}) - c(\{1_f, 4_f\}) = 93.
\end{aligned}$$

So the greedy algorithm chooses to open facility 4_f and update X to $\{1_f, 4_f\}$, even though this is not optimal.

The example is $(\frac{7}{2} - \varepsilon)$-perturbation resilient, as switching from the optimal solution $X^* = \{1_f, 2_f, 3_f\}$ only starts making sense once both customer 2_d and 3_d switching to facility 4_f is advantageous enough, which happens when you have a $\frac{7}{2}$-perturbation of this example.

4.2 Greedy facility deletion

Another greedy algorithm is greedy facility deletion. In this algorithm, you start from some initial set of facilities X, e.g. $X = F$, and greedily drop facilities until some condition is met. Here, we drop the facility with maximum cost reduction (i.e. $f_i + \sum_{j \in D} c_{\sigma(j)j} - \min_{i' \in X \setminus \{i\}} c_{i'j}$), as long as that remains positive.

The following FLP instance, using $k \in \mathbb{N}, k \geq 4$, is a counterexample that shows this does not always yield the optimal solution. When k converges to infinity, the instance converges to 3-perturbation resilience.

$$\begin{aligned}
F &= \{0_f, 1_f, \dots, k_f\},\\
D &= \{1_d, \dots, k_d\},\\
f_i &= \begin{cases} \dfrac{3k-2}{k-2} & \text{if } i = 0_f \text{, and}\\ \dfrac{3k-3}{k-2} & \text{otherwise,} \end{cases}\\
c_{ij} &= \begin{cases} 1 & \text{if } i = 0_f \text{, and}\\ 1 & \text{if } i = k_f, j = k_d \text{ for some } k \text{, and}\\ 3 & \text{otherwise,} \end{cases}\\
X^* &= \{0_f\}.
\end{aligned}$$

The optimal solution is $X^* = \{0_f\}$, since opening less facilities always reduces $c_F(X)$ in this example, and letting 0_f stay open makes sure that the service costs $c_S(X^*)$ stay minimal. However, greedy facility deletion, when starting with $X = F$, drops facility 0_f first. This is because it is the most expensive facility and the service costs do not increase in the first deletion, no matter which facility in X you close.

For $k \geq 4$, X^* remains the optimal solution even when the instance is γ-perturbed with γ converging to 3 when k goes to infinity, since there are a lot of facilities $1_f, \dots, k_f$ and only one 0_f. At $\gamma = 3$ the costs of the more expensive facility 0_f cannot be offset by the service costs, since connecting all customers to e.g. 1_f has lower costs in that case.

5 Relation to approximation algorithms

In this section we will show one of the best approximation algorithms for the FLP, the Jain-Mahdian-Saberi algorithm [17], with approximation ratio 1.61. As approximation algorithms guarantee a solution X with cost $c(X) \leq \alpha c(X^*)$, with α depending on the algorithm, it seems obvious that these algorithms should give the optimal solution to γ-perturbation resilient instances, if $\gamma \geq \alpha$. After all, with γ-perturbation resilience you can multiply the costs of the optimal solution with γ and it is still the optimal solution. We will show that this conjecture is not true, even for a good algorithm like the Jain-Mahdian-Saberi algorithm.

5.1 The Jain-Mahdian-Saberi algorithm

To understand the Jain-Mahdian-Saberi algorithm, it is useful to first look at the integer linear program (LP) formulation of the facility location problem:

$$\begin{aligned} \min \quad & \sum_{i \in F} f_i y_i + \sum_{i \in F} \sum_{j \in D} c_{ij} x_{ij} & \\ \text{s.t.} \quad & x_{ij} \leq y_i & \text{for all } i \in F, j \in D, \\ & \sum_{i \in F} x_{ij} = 1 & \text{for all } j \in D, \\ & x_{ij} \in \{0,1\} & \text{for all } i \in F, j \in D, \\ & y_i \in \{0,1\} & \text{for all } i \in F. \end{aligned}$$

Of course, solving this LP is not simpler than solving the FLP. But if you relax the integer variables such that $x_{ij} \in [0,1]$ and $y_i \in [0,1]$, it is possible to solve the LP in polynomial time. This is the basis of LP-rounding algorithms, which calculate a FLP solution from the relaxed LP solution, which typically is not an integer solution. Better algorithms can be derived by also using the dual of the relaxed LP and the resulting complementary slackness. These algorithms are called primal-dual algorithms. Of course, the difficulty in this process is somehow getting a integral solution.

Definition. *The* Jain-Mahdian-Saberi algorithm *[17] is a primal-dual algorithm for the FLP.*

For every customer j and facility i, variables w_{ij} and v_j are introduced. Every customer starts unconnected *and every facility starts* closed. *The algorithm starts by maintaining $v_j = t$ and processes the following events while increasing t:*

- *$v_j = c_{ij}$ for some unconnected customer j and closed facility i. Start increasing w_{ij} such that $c_{ij} + w_{ij} = v_j$.*
- *$\sum_{j \in D} w_{ij} = f_i$ for some closed facility i. Open facility i and, for all customers j with $w_{ij} > 0$, stop increasing v_j and set $w_{i'j} = \max\{0, c_{ij} - c_{i'j}\}$ for all $i' \in F$ and mark customer j as connected.*
- *$v_j = c_{ij}$ for some unconnected customer j and open facility i. Stop increasing v_j and set $w_{i'j} = \max\{0, c_{ij} - c_{i'j}\}$ for all $i' \in F$ and mark customer j as connected.*

Ties between events, when occurring at the same t, are broken arbitrarily. When there are no more events to be processed, the algorithm will have connected all customers and yields a set of open facilities.

Intuitively, the Jain-Mahdian-Saberi algorithm lets customers pay for facilities to open, and lets cheap connections pay more than expensive connections. The solution of the algorithm has several approximation guarantees $c(X) \leq \alpha c_F(X^*) + \beta c_S(X^*)$ as a result from its so-called factor-revealing LP [17]. Examples of these guarantees are ($\alpha = 1, \beta = 2$), ($\alpha = 1.11, \beta = 1.78$) and ($\alpha = 1.61, \beta = 1.61$). This means that the solution of the Jain-Mahdian-Saberi algorithm satisfies *all* of these guarantees, you do not have to choose.

Note that better algorithms exist for the FLP (e.g. the 1.5-approximation algorithm by Byrka and Aardal [10]) and that the difference in approximation guarantee for the service costs $c_S(X^*)$ and facility costs $c_F(X^*)$ can be exploited to yield better approximation ratios (see Charikar and Guha [11], yielding a ratio of 1.51 when used with the Jain-Mahdian-Saberi algorithm).

5.2 Perturbation resilience and solutions of approximation algorithms

Since the γ-perturbation resilience property is almost equivalent to saying that the optimal solution stays unique and optimal even when its costs are multiplied by γ, it seems there must be some kind of relation to approximation algorithms for the FLP. After all, if some calculated solution cannot be worse than α times the optimum, then it likely must imply that it either is equal to the optimum or, if not, that $\gamma < \alpha$, for γ-perturbation resilient instances. This reasoning is wrong however, as we will show with the following example.

Define the following instance (see Figure 4):

$$\begin{aligned} F &= \{1_f, 2_f\}, \\ D &= \{1_d, 2_d\}, \\ f &= (3, 1), \\ c_{ij} &= \begin{pmatrix} 9 & 4 \\ 20 & 7 \end{pmatrix}. \end{aligned}$$

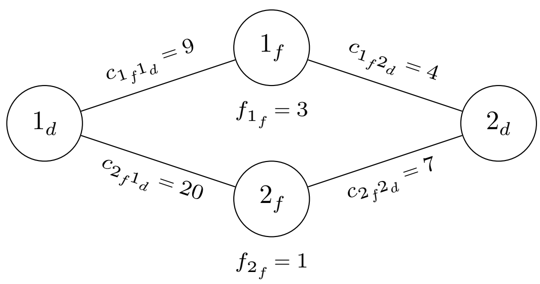

Figure 4: An α-perturbation resilient instance which has an α-approximate solution $X \neq X^*$

This example has $(\frac{7}{4} - \varepsilon)$-perturbation resilience. This can be seen by considering the cost of all solutions under the γ-perturbation defined in Theorem 5, where the following costs already assume that $\gamma < \frac{7}{4}$:

$$\begin{aligned} c'(X^*) = c'(\{1_f\}) &= 16\gamma, \\ c'(\{2_f\}) &= 28, \\ c'(\{2_f, 3_f\}) &= 1 + 16\gamma. \end{aligned}$$

So $c'(X^*) < c'(X)$ for all solutions $X \neq X^*$ iff $\gamma < \frac{7}{4}$, so the instance is $(\frac{7}{4} - \varepsilon)$-perturbation resilient.

The Jain-Mahdian-Saberi algorithm has an approximation ratio of 1.61, which is smaller than $\frac{7}{4}$, and approximates within a ratio of 1.03 on average for instances in practice [17]. When this algorithm is ran on the given example with pessimistic breaking of ties, the solution is

$X = \{1_f, 2_f\} \neq X^*$. So this example shows that α-approximation algorithms do not guarantee optimal solutions for γ-perturbation resilient instances with $\gamma \geq \alpha$.

Looking at the example, it is clear what causes this mismatch between the two concepts: adding a cheap facility (facility 2_f in the example) does not hurt the solution so much that it invalidates the approximation ratio. However, the solution is not optimal, since the original solution is extremely good, even after some perturbation.

6 Complexity of the perturbation resilient FLP

In this section we will look into the complexity of the perturbation resilient FLP. The FLP itself is NP-hard, so it is interesting to find out if the perturbation resilient FLP is as difficult as the FLP or easier. Here we look into the lower bound of the hardness of the perturbation resilient FLP.

6.1 Relation to complexity class RP

In this section we will show that the perturbation resilient FLP has a relation to the complexity class RP. First, let us define RP.

Definition. *A problem is contained in the complexity class* RP *iff a polynomial time algorithm for the problem exists which, given a source of random numbers, accepts an instance with probability at least $\frac{1}{2}$ if it is a yes-instance and always rejects if it is a no-instance.*

It is relatively easy to prove that P $\subseteq$ RP $\subseteq$ NP. An open question is if any of these inclusions are strict. A way of looking at a problem in RP is that, out of all its computation paths (as defined by every usage of a random number), at least half accept for a yes-instance. A problem in NP however, only requires one accepting computation path for a yes-instance. Thus, intuitively, problems in RP should be easier than problems in NP. This idea can be formalized, by changing all points where a random number is read in an algorithm for a problem in RP to an non-deterministic choice of numbers, which results in an algorithm with at least one accepting path for every yes-instance and no accepting path for every no-instance, showing that the problem is in NP too. An example of a problem in RP (but later proven to also be in P) is the primality problem [2].

In the following definition and theorem we shed some more light on the relation between RP and NP. First, we need the definition of Unambiguous-SAT, which is used in the Valiant–Vazirani theorem, upon which our proof for the relation between RP and NP is based.

Definition. Unambiguous-SAT *is a variant of* 3-SAT*. In* Unambiguous-SAT *we are given a 3-SAT instance, consisting of a boolean formula in 3-conjunctive normal form, with the promise that this instance has exactly one satisfying solution.*

If the promise holds for the given instance, the output of an algorithm for Unambiguous-SAT should be the solution for the instance. If not, the output can be anything, the algorithm is not even required to terminate.

Theorem 14 (Valiant–Vazirani [31]). *If a polynomial-time deterministic algorithm exists that gives a solution for all instances of Unambiguous-SAT (and can do anything on other instances of 3-SAT), then RP = NP.*

A similar relation between RP and NP can be found using the perturbation resilient FLP, as we will show below. First, a refinement of Theorem 8 is needed, since we need to solve γ-perturbation resilient FLP instances with arbitrarily small γ in Theorem 16, which relates the perturbation resilient FLP to RP = NP.

Lemma 15. *Assume a polynomial time deterministic algorithm* `A` *exists that decides FLP for all $g(|F|, |D|)$-perturbation resilient instances (F, D, f, c), where g is a function monotonically decreasing in $|F|$, i.e. $f(|F|, |D|) > f(|F'|, |D|)$ if $|F'| > |F|$.*

Then a polynomial time deterministic algorithm exists that finds the unique optimal solution X^* for those instances.

This lemma implies that if we have such an algorithm `A`, we can use it to find solutions to all $g(|F|, |D|)$-perturbation resilient instances.

Proof. Assume algorithm `A` exists. Take any possible γ-perturbation resilient FLP instance (F, D, f, c), with $\gamma = g(|F|, |D|)$, and modify it as follows:

$$F' = F \cup \{\xi\},$$
$$f'_i = \begin{cases} f_i & \text{if } i \in F \text{, and} \\ 1 & \text{otherwise,} \end{cases}$$
$$c'_{ij} = \begin{cases} c_{ij} & \text{if } i \in F \text{, and} \\ M & \text{otherwise,} \end{cases}$$

where M satisfies $M > \sum_{j \in D} \max_{i \in F} \gamma(f_i + c_{ij})$, i.e. it is so large that no customer will be assigned to facility ξ in the optimal solution to (F', D, f', c') or any of its γ-perturbed instances.

This modified instance (F', D, f', c') also is γ-perturbation resilient, since the optimal solution X'^* trivially is X^*. The optimal solution has cost $c(X'^*) = c(X^*)$. Opening ξ is never a good idea, meaning all customers assignments and facilities to open are the same as in X^*.

Since $|F'| > |F|$, $\gamma = g(|F|, |D|) > g(|F'|, |D|)$. Using this and the assumption about the existence of `A`, Theorem 8 implies that a deterministic polynomial time algorithm exists that finds the optimal solution X'^* to instance (F', D, f', c'). Because $X'^* = X^*$, finding the optimal solution X'^* means we found the optimal solution to instance (F, D, f, c).

Thus, a polynomial time deterministic algorithm exists that finds the optimal solution X^* for all $g(|F|, |D|)$-perturbation resilient FLP instances. $\square$

With this lemma we have all the tools we need for proving that, if an algorithm exists that solves $(1 + \frac{2}{2|D|+|F|})$-perturbation resilient FLP instances in deterministic polynomial time, then RP = NP.

Theorem 16. *If a polynomial time deterministic algorithm* `A` *exists that decides FLP for* $(1 + \frac{2}{2|D|+|F|})$*-perturbation resilient instances, then RP = NP.*

So if RP $\neq$ NP – which, while unproven, is not an uncommon belief – then no such algorithm `A` exists.

Proof. Assume algorithm `A` exists. By Lemma 15, a polynomial time deterministic algorithm `A` exists which finds the solution of $(1 + \frac{2}{2|D|+|F|})$-perturbation resilient FLP instances.

By transforming an instance of Unambiguous-SAT to a FLP instance and running algorithm `A`, a solution can be found for the Unambiguous-SAT instance. Then transform this solution back to the Unambiguous-SAT instance. By the Valiant–Vazirani theorem this will prove that RP = NP.

Let (C, Y) be an instance of Unambiguous-SAT, with $Y = \{x_1, \ldots, x_n\}$ and $C = c_1 \wedge \ldots \wedge c_m$, in which $c_k = (c_{k,1} \vee c_{k,2} \vee c_{k,3})$ and $c_{k,l} = x_i$ or $c_{k,l} = \bar{x}_i$ for some $x_i \in Y$. Now we define the following FLP instance:

$$F = \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\},$$
$$D = \{x_1, \ldots x_n\} \cup \{c_1, \ldots, c_m\},$$
$$f = 1,$$
$$c = \begin{cases} 1 & \text{if } j = x_k \text{ for some } k \text{ and } i = x_k \text{ or } i = \bar{x}_k \text{, and} \\ 1 & \text{if } j = c_k \text{ for some } k \text{ and } i \in \{c_{k,1}, c_{k,2}, c_{k,3}\} \text{, and} \\ 3 & \text{otherwise.} \end{cases}$$

The optimal solution X^* to this instance will never have a customer j connected to a facility i with $c_{ij} = 3$, since there always is some facility i' (potentially unopened) such that $c_{i'j} = 1$.

Opening this facility i', with facility cost $f_{i'} = 1$, will lower the total cost, so $c_{\sigma^*(j)j} = 1$ and $c_S(X^*) = n + m$.

Since $c_{\sigma^*(j)j} = 1$ for $j = x_k \in \{x_1, \dots, x_n\}$, either $x_k \in X^*$ or $\bar{x}_k \in X^*$. Thus, $|X^*| \geq n$ and $c_F(X^*) \geq n$.

Now suppose that $c(X) = 2n + m$ for some solution X. By the above inequalities, $c(X) = c(X^*)$ and exactly one of $(x_k \in X), (\bar{x}_k \in X)$ is true for every $x_k \in Y$. This is a assignment for the boolean formula in Unambiguous-SAT. Also because of the above inequalities, every clause c_k is connected to an open facility with cost 1. By construction, this means the clause is satisfied. Thus, X induces the solution for Unambiguous-SAT.

Going in the reverse direction, the unique solution for Unambiguous-SAT induces an optimal solution to the FLP instance, by adding x_k to X^* if $x_k = 1$ and adding $\bar{x}_k$ if $x_k = 0$. This solution has cost $c(X^*) = 2n + m$.

Thus, the constructed instance has an unique optimal solution X^* with $c(X^*) = 2n + m$. If another solution X would exist, a corresponding solution would exist for the Unambiguous-SAT instance too, which cannot happen.

Since all costs are integer, all solutions $X \neq X^*$ have $c(X) \geq 2n + m + 1$, making the instance $(1 + \frac{2}{2|D|+|F|} - \varepsilon)$-perturbation resilient. If, as by assumption, an algorithm exists which can solve these instances in deterministic polynomial time, then all Unambiguous-SAT instances can be solved in deterministic polynomial time. By the Valiant–Vazirani theorem this means RP = NP. $\square$

6.2 Hardness for constant γ

Although Theorem 16 shows some hardness-related results for the perturbation resilient FLP, it only shows it for sub-constant γ-perturbation resilience. Proving that FLP with γ-perturbation resilient instances is NP-hard or something similar for small but constant γ seems to be difficult. The first reason is that uniqueness of the solution is difficult to achieve in a reduction, since that usually requires the original instance to be unique. This problem can be worked around by using RP instead of only trying to prove NP-hardness, as was done in the previous section.

The second reason is that γ-perturbation resilient requires the unique solution to have costs a factor γ lower than other solutions, if you use the obvious approach. A reduction thus must introduce a gap between the optimal solution and other solutions, without knowing the solution. The trivial γ-perturbation resilience is, in that case, the cost of the second-best solution divided by cost of the optimal solution. The requirement that instances are metric makes sure that distances cannot differ too much, making this hard to achieve.

Despite the above indications that proving NP-hardness for small but constant γ is difficult, there also is an indication that it might be possible: Awasthi et al. have proven that k-means clustering, with an assumption similar to γ-perturbation resilience, is NP-hard when $\gamma < 3$, and they give a polynomial time algorithm for solving instances with $\gamma \geq 3$ [4]. This cannot be directly translated to the FLP however, since the freedom of number of facilities and facility costs are hard to incorporate into their core assumption. This does however indicate that the FLP with perturbation resilience may be NP-hard for some small γ, despite the difficulty in proving that.

7 Conclusion

In this thesis we introduced the γ-perturbation resilience assumption for the facility location problem. It seems likely that many instances in practice already satisfy or come close to this assumption, at least for small γ. An equivalent but less intuitive definition was also given, as that definition is easier to use in some proofs. Using the decision variant of the facility location problem is enough to solve the search variant even with γ-perturbation resilience, as is also true for the normal FLP.

Local search algorithms are guaranteed to produce the optimal solution for γ-perturbation resilient instances with $\gamma \geq 3$. For all $\gamma < 3$ there are γ-perturbation resilient instances which will not result in the optimal solution when using a local search algorithm. We also showed that several greedy algorithms can also give non-optimal solutions for γ-perturbation resilient instances with high γ.

The same happened with a well-performing approximation algorithm, the Jain-Mahdian-Saberi algorithm. This algorithm failed on γ-perturbation resilient instances with moderately high γ. This was surprising, since there seemed to be an obvious relation between approximation algorithms and the γ-perturbation resilience property. The counterexample for the Jain-Mahdian-Saberi algorithm shows that this conjecture is false.

Finally, we also showed that, if an efficient algorithm for solving γ-perturbation resilient instances with low γ exists, then RP = NP.

In conclusion, it looks as if the γ-perturbation resilience assumption does not make the facility location problem much easier. All results indicate that a relatively high γ for γ-perturbation resilient instances is needed to make the problem easier to solve, whereas such instances are not likely to appear in practice.

To make further progress on solving the facility location problem using this assumption, we recommend looking into custom algorithms exploiting the γ-perturbation resilience assumption, since exploiting the structure inherent in such instances may yield better results than using existing algorithms. Other natural assumptions can also be used to further analyze the facility location problem. An obvious example of such an assumption is optimal solution stability under removal of some (fraction of) customers.

Finally, as the γ-perturbation resilience assumption has already been used successfully for the k-means clustering problem and MAX-CUT, there may be other problems where this assumption works well. As an example, the knapsack problem may work well with this assumption.

8 References

[1] M. Ackerman and S. Ben-David. “Clusterability: A theoretical study”. In: *Journal of Machine Learning Research* 5 (2009), pp. 1–8.

[2] M. Agrawal, N. Kayal, and N. Saxena. “PRIMES is in P”. In: *Annals of Mathematics* 160.2 (2004), pp. 781–793.

[3] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. “Local search heuristics for k-median and facility location problems”. In: *SIAM Journal on Computing* 33.3 (2004), pp. 544–562. DOI: 10.1137/S0097539702416402.

[4] P. Awasthi, A. Blum, and O. Sheffet. “Center-based clustering under perturbation stability”. In: *Information Processing Letters* 112.1-2 (2012), pp. 49–54. DOI: 10.1016/j.ipl.2011.10.006.

[5] M.F. Balcan, A. Blum, and A. Gupta. “Clustering under approximation stability”. In: *Journal of the ACM* 60.2 (2013). DOI: 10.1145/2450142.2450144.

[6] M.F. Balcan, H. Röglin, and S.-H. Teng. “Agnostic clustering”. In: *Lecture Notes in Computer Science* 5809 (2009), pp. 384–398. DOI: 10.1007/978-3-642-04414-4_31.

[7] S. Ben-David and L. Reyzin. “Data stability in clustering: A closer look”. In: *Theoretical Computer Science* 558.C (2014), pp. 51–61. DOI: 10.1016/j.tcs.2014.09.025.

[8] Y. Bilu, A. Daniely, N. Linial, and M. Saks. “On the practically interesting instances of MAXCUT”. In: vol. 20. 2013, pp. 526–537. DOI: 10.4230/LIPIcs.STACS.2013.526.

[9] Y. Bilu and N. Linial. “Are stable instances easy?” In: *Combinatorics Probability and Computing* 21.5 (2012), pp. 643–660. DOI: 10.1017/S0963548312000193.

[10] J. Byrka and K. Aardal. “An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem”. In: *SIAM Journal on Computing* 39.6 (2010), pp. 2212–2231. DOI: 10.1137/070708901.

[11] M. Charikar and S. Guha. “Improved combinatorial algorithms for the facility location and k-median problems”. In: 1999, pp. 378–388. DOI: 10.1109/SFFCS.1999.814609.

[12] M. Charikar, S. Khuller, D.M. Mount, and G. Narasimhan. “Algorithms for facility location problems with outliers”. In: 2001, pp. 642–651.

[13] S.A. Cook. “The Complexity of Theorem-proving Procedures”. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. STOC ’71. Shaker Heights, Ohio, USA: ACM, 1971, pp. 151–158. DOI: 10.1145/800157.805047.

[14] L. Engebretsen. “An explicit lower bound for TSP with distances one and two”. In: *Algorithmica (New York)* 35.4 (2003), pp. 301–319. DOI: 10.1007/s00453-002-1001-6.

[15] M.R. Garey and D.S. Johnson. *Computers and intractability*. W.H. Freeman, 1979. ISBN: 0-7167-1045-5.

[16] S. Guha and S. Khuller. “Greedy Strikes Back: Improved Facility Location Algorithms”. In: *Journal of Algorithms* 31.1 (1999), pp. 228–248.

[17] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani. “Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP”. In: *Journal of the ACM* 50.6 (2003), pp. 795–824. DOI: 10.1145/950620.950621.

[18] D.S. Johnson and Lyle A. McGeoch. “The traveling salesman problem: A case study”. In: John Wiley & Sons, 1997. Chap. 8.

[19] R.M. Karp. “Reducibility among combinatorial problems”. In: *Complexity of computer computations*. Springer, 1972, pp. 85–103. DOI: 10.1007/978-1-4684-2001-2_9.

[20] J. Krarup and P.M. Pruzan. “The simple plant location problem: Survey and synthesis”. In: *European Journal of Operational Research* 12.1 (1983), pp. 36–57, 60–81. DOI: 10.1016/0377-2217(83)90181-9.

[21] J. Kun and L. Reyzin. “On coloring resilient graphs”. In: *Lecture Notes in Computer Science* 8635 LNCS.PART 2 (2014), pp. 517–528. DOI: 10.1007/978-3-662-44465-8_44.

[22] M. Künnemann and B. Manthey. “Towards understanding the smoothed approximation ratio of the 2-opt heuristic”. In: *Lecture Notes in Computer Science* 9134 (2015), pp. 859–871. DOI: 10.1007/978-3-662-47672-7_70.

[23] S. Li. “A 1.488 approximation algorithm for the uncapacitated facility location problem”. In: *Information and Computation* 222 (2013), pp. 45–58. DOI: 10.1016/j.ic.2012.01.007.

[24] K. Liao and D. Guo. “A Clustering-based approach to the capacitated facility location problem”. In: *Transactions in GIS* 12.3 (2008), pp. 323–339. DOI: 10.1111/j.1467-9671.2008.01105.x.

[25] M.T. Melo, S. Nickel, and F. Saldanha-da-Gama. “Facility location and supply chain management - A review”. In: *European Journal of Operational Research* 196.2 (2009), pp. 401–412. DOI: 10.1016/j.ejor.2008.05.007.

[26] M. Mihalák, M. Schöngens, R. Šrámek, and P. Widmayer. “On the complexity of the metric tsp under stability considerations”. In: *Lecture Notes in Computer Science* 6543 LNCS (2011), pp. 382–393. DOI: 10.1007/978-3-642-18381-2_32.

[27] C.H. Papadimitriou. *Computational Complexity*. Addison Wesley Publishing Company, 1994. ISBN: 0-201-53082-1.

[28] Ran Raz and Shmuel Safra. “Sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP”. In: 1997, pp. 475–484. DOI: 10.1145/258533.258641.

[29] F. Schalekamp, M. Yu, and A. Van Zuylen. “Clustering with or without the approximation”. In: *Journal of Combinatorial Optimization* 25.3 (2013), pp. 393–429. DOI: 10.1007/s10878-011-9382-6.

[30] D.A. Spielman and S.-H. Teng. “Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time”. In: *Journal of the ACM* 51.3 (2004), pp. 385–463. DOI: 10.1145/990308.990310.

[31] L.G. Valiant and V.V. Vazirani. “NP is as easy as detecting unique solutions”. In: *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*. 1985, pp. 458–463. DOI: 10.1145/22145.22196.

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