Squared Metric Facility Location Problem[*]

C.G. Fernandes[†], L.A.A. Meira[‡], F.K. Miyazawa[§], and L.L.C. Pedrosa[§]

November 7, 2011

Abstract

Jain *et al.* proposed two well-known algorithms for the Metric Facility Location Problem (MFLP), that achieve approximation ratios of 1.861 and 1.61. Mahdian *et al.* combined the latter algorithm with scaling and greedy augmentation techniques, obtaining a 1.52-approximation for the MFLP. We consider a generalization of the Squared Euclidean Facility Location Problem, when the distance function is a squared metric, which we call Squared Metric Facility Location Problem (SMFLP). We show that the algorithms of Jain *et al.* and of Mahdian *et al.*, when applied to this variant of the facility location, achieve approximation ratios of 2.87, 2.43, and 2.17, respectively. It is shown that, for the SMFLP, there is no 2.04-approximation algorithm, assuming P $\neq$ NP. In our analysis, we used nonlinear factor-revealing programs to obtain both lower and upper bounds on the approximation factors, and propose a systematic way to derive such factor-revealing programs.

---

[*] This research was partially supported by CNPq and FAPESP.

[†] Institute of Mathematics and Statistics, University of São Paulo, Brazil `cris@ime.usp.br`.

[‡] Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, Brazil, `augusto.meira@unifesp.br`.

[§] Institute of Computing, University of Campinas, Brazil, `fkm@ic.unicamp.br`, `lehilton@ic.unicamp.br`.

1 Introduction

Let C and F be finite disjoint sets. We call *cities* the elements of C and *facilities* the elements of F. For each facility i and city j, let c_{ij} be a non-negative number representing the cost to connect i to j. Additionally, let f_i be a non-negative number representing the cost to open facility i. For each city j and subset F' of F, let $c(F', j) = \min_{i \in F'} c_{ij}$. The FACILITY LOCATION PROBLEM (FLP) consists of the following: given sets C and F, and c and f as above, find a subset F' of F such that $\sum_{i \in F'} f_i + \sum_{j \in C} c(F', j)$ is minimum. Hochbaum [7] presented an $\mathrm{O}(\log n)$-approximation for the FLP. Archer [1] showed that this result is asymptotically tight, unless $\mathrm{NP} \subseteq \mathrm{DTIME}[n^{\mathrm{O}(\log \log n)}]$, by presenting a reduction from the Set Cover problem.

A well-studied particular case of the FLP is its so called metric variant. We say that an instance (C, F, c, f) of the FLP is *metric* if $c_{ij} \leq c_{ij'} + c_{i'j'} + c_{i'j}$, for all facilities i and i', and cities j and j'. In the context of the FLP, this inequality is called the *triangle inequality*. The METRIC FLP, denoted by MFLP, is the particular case of the FLP that considers only metric instances. Several algorithms were proposed in the literature for the MFLP [3, 4, 5, 9, 10, 12, 13, 15]. In particular, the best known algorithm for the MFLP is a 1.488-approximation proposed by Li [12]. Also, there is an inapproximability result that states that there is no approximation algorithm for the MFLP with a ratio smaller than 1.463, unless $\mathrm{NP} \subseteq \mathrm{DTIME}[n^{\mathrm{O}(\log \log n)}]$ [6]. This last result was strengthened by Sviridenko, who showed that the lower bound holds unless $\mathrm{P} = \mathrm{NP}$ [16].

The EUCLIDEAN FLP is a particular case of the MFLP also considered in the literature. In the Euclidean FLP, one is given a position in an Euclidean space for each city and for each facility, and the cost c_{ij} is the Euclidean distance between the position of facility i and the position of city j. There is a PTAS for the Euclidean FLP in 2-dimensional space, by Arora, Raghavan, and Rao [2].

Yet another variant considered in the literature is the so called SQUARED EUCLIDEAN FLP, denoted here by E^2FLP. In this variant, as in the Euclidean case, one is given a position in an Euclidean space for each city and for each facility. Here, the cost c_{ij} is the square of the Euclidean distance between the position of facility i and the position of city j. This choice of costs discourages excessive distances in the solution. This cost measure is known as ℓ_2^2, and was, for instance, considered by Jain and Vazirani [10, pp. 292–293] in the context of the FLP. Their approach implies a 9-approximation for the E^2FLP.

We consider instances (C, F, c, f) of the FLP such that a relaxed version of the triangle inequality is satisfied. We say that a cost function c is a *squared metric*, if, for all facilities i and i', and cities j and j', we have $\sqrt{c_{ij}} \leq \sqrt{c_{ij'}} + \sqrt{c_{i'j'}} + \sqrt{c_{i'j}}$. The particular case of the FLP that only considers instances with a squared metric is called SQUARED METRIC FLP, and is denoted by SMFLP. Notice that the SMFLP is a generalization of the E^2FLP and of the MFLP. Thus any approximation for the SMFLP is also an approximation for the E^2FLP or the MFLP, and the inapproximability results for the MFLP are also valid for the SMFLP.

Although there are several algorithms for the MFLP in the literature, there are very few works on the SMFLP. Nevertheless, one may try to solve an instance of the SMFLP using good algorithms designed for the MFLP. Since these algorithms and their analysis are based on the assumption of the triangle inequality, it is reasonable to expect that they generate good solutions also for the SMFLP. However, there is no trivial way to derive an approximation factor from the MFLP to the SMFLP, so each algorithm must be reanalyzed individually. In this paper, we analyze the 1.861 and the 1.61-approximation algorithms of Jain *et al.* [9], and the 1.52-approximation of Mahdian *et al.* [13], when applied to squared metric FLP instances. We show that these algorithms achieve ratios of 2.87, 2.43, and 2.17, respectively. We remark that these factors are close to the 2.04-inapproximability limit for the SMFLP, that was obtained by extending the hardness results of Guha and Khuller [6] for the metric case.

The original analysis of the three algorithms are based on the so called families of *factor-revealing linear programs* [9, 13]. For a certain number of cities, the approximation ratio is given by a computer calculated solution of the corresponding linear program. This gives a lower bound on the approximation factor. The upper bound, however, is obtained analytically by usually long and tedious proofs. In this paper, we propose a way to obtain upper bound factor-revealing programs for the SMFLP, as an alternative technique to achieve an upper bound. Therefore, both upper and lower bounds on the approximation factor are obtained by computer calculated solutions of optimization programs. We note that, in our case, the factor-revealing programs are nonlinear, since the squared metric constraints contain square roots. We tackle this by replacing these constraints with an infinite set of linear constraints.

Recently, we became aware of the *strongly factor-revealing linear programs*, proposed by Mahdian and Yan [14]. Any optimization program that gives an upper bound on the optimal value of a family of maximization programs $LP(k)$ can be thought of as a strongly factor-revealing program. In this sense, our upper bound factor-revealing program is also a strongly factor-revealing program. The techniques involved in obtaining such programs, however, are different. To obtain a strongly factor-revealing linear program, one projects a solution of $LP(md)$ in $LP(m)$, and tries to adjust the restrictions to obtain a feasible solution. In our approach, we define a candidate dual solution for $LP(k)$ from a fixed number of variables, and obtain an upper bound factor-revealing program directly in the form of a minimization program. For the case of the SMFLP, we observed that calculating the dual upper bound program seems easier than projecting the solutions on the primal. Also, we have considered the case of the MFLP, for which the obtained lower and upper bound factor-revealing programs converge.

2 Preliminaries

Observe that the E^2FLP (and therefore the SMFLP) is not a particular case of the MFLP. For example, consider an instance of the E^2FLP consisting of two facilities i and i' at positions $(0,0)$ and $(0,2)$, and two cities j and j' at $(0,1)$ and $(0,3)$. Thus, the cost function c is such that $c_{ij'} = 9$, and $c_{ij} = c_{i'j} = c_{i'j'} = 1$, so c does not satisfy the triangle inequality.

Although the constraints over the cost function c from an SMFLP instance are defined by square roots, the next lemma shows that a squared metric can be expressed by an infinite set of linear inequalities. As a consequence, for any cost function not satisfying the squared metric inequality, there exists some linear inequality, as defined in Lemma 1, that is violated.

Lemma 1. *Let A, B, C, and D be non-negative numbers. Then $\sqrt{A} \leq \sqrt{B} + \sqrt{C} + \sqrt{D}$ if and only if $A \leq (1 + \beta + \frac{1}{\gamma})B + (1 + \gamma + \frac{1}{\delta})C + (1 + \delta + \frac{1}{\beta})D$ for every positive numbers β, γ, and δ. In particular, if $\sqrt{A} \leq \sqrt{B} + \sqrt{C} + \sqrt{D}$, then $A \leq 3B + 3C + 3D$.*

Proof. Suppose $\sqrt{A} \leq \sqrt{B} + \sqrt{C} + \sqrt{D}$. As $(\sqrt{\beta B} - \sqrt{D/\beta})^2 \geq 0$, we have that $2\sqrt{BD} \leq \beta B + D/\beta$. Similarly, $2\sqrt{CB} \leq \gamma C + B/\gamma$ and $2\sqrt{DC} \leq \delta D + C/\delta$. Therefore, if $\sqrt{A} \leq \sqrt{B} + \sqrt{C} + \sqrt{D}$, then

$$\begin{aligned} A &\leq (\sqrt{B} + \sqrt{C} + \sqrt{D})^2 \\ &= B + C + D + 2\sqrt{BD} + 2\sqrt{CB} + 2\sqrt{DC} \\ &\leq B + C + D + \beta B + D/\beta + \gamma C + B/\gamma + \delta D + C/\delta \\ &= (1 + \beta + \frac{1}{\gamma})B + (1 + \gamma + \frac{1}{\delta})C + (1 + \delta + \frac{1}{\beta})D. \end{aligned}$$

Choosing $\beta = \gamma = \delta = 1$, we obtain $A \leq 3B + 3C + 3D$.

Now suppose $\sqrt{A} > \sqrt{B} + \sqrt{C} + \sqrt{D}$. Let $d > 0$ be such that $A = B + C + D + 2\sqrt{BD} + 2\sqrt{CB} + 2\sqrt{DC} + d$. Then, $A > (1 + \beta + \frac{1}{\gamma})B + (1 + \gamma + \frac{1}{\delta})C + (1 + \delta + \frac{1}{\beta})D$ is equivalent to $(\beta + \frac{1}{\gamma})B + (\gamma + \frac{1}{\delta})C + (\delta + \frac{1}{\beta})D < 2\sqrt{BD} + 2\sqrt{CB} + 2\sqrt{DC} + d$. We will analyze the cases in which none, one, two or all numbers B, C and D are zero. Let ξ and ξ' be positive numbers such that $\xi + \xi' < 1$.

Case 1: $B, C, D > 0$. Let $\beta = \sqrt{\frac{D}{B}}$, $\gamma = \sqrt{\frac{B}{C}}$ and $\delta = \sqrt{\frac{C}{D}}$. Then $(\beta + \frac{1}{\gamma})B + (\gamma + \frac{1}{\delta})C + (\delta + \frac{1}{\beta})D = 2\sqrt{BD} + 2\sqrt{CB} + 2\sqrt{DC} < 2\sqrt{BD} + 2\sqrt{CB} + 2\sqrt{DC} + d$.

Case 2: $B = 0$ and $C, D > 0$. Let $\beta = \frac{D}{\xi d}$, $\gamma = \frac{\xi' d}{C}$ and $\delta = \sqrt{\frac{C}{D}}$. Then $(\beta + \frac{1}{\gamma})B + (\gamma + \frac{1}{\delta})C + (\delta + \frac{1}{\beta})D = 2\sqrt{DC} + (\xi + \xi')d < 2\sqrt{BD} + 2\sqrt{CB} + 2\sqrt{DC} + d$.

Case 3: $B, C = 0$ and $D > 0$. Let $\beta = \frac{D}{\xi d}$, $\gamma = 1$ and $\delta = \frac{\xi' d}{D}$. Then $(\beta + \frac{1}{\gamma})B + (\gamma + \frac{1}{\delta})C + (\delta + \frac{1}{\beta})D = (\xi + \xi')d < 2\sqrt{BD} + 2\sqrt{CB} + 2\sqrt{DC} + d$.

Case 4: $B, C, D = 0$. Let $\beta = 1$, $\gamma = 1$ and $\delta = 1$. Then $(\beta + \frac{1}{\gamma})B + (\gamma + \frac{1}{\delta})C + (\delta + \frac{1}{\beta})D = 0 < 2\sqrt{BD} + 2\sqrt{CB} + 2\sqrt{DC} + d$. □

We also use the concept of a bifactor approximation algorithm, usually adopted in the context of the FLP. Consider an algorithm that generates a solution with possibly distinct approximation factors for facility and connection costs. A bifactor approximation for the FLP, as defined by Mahdian *et al.* [13], is described in the following:

Definition 2 (Bifactor approximation [13])**.** *An algorithm is called a (γ_f, γ_c)-approximation algorithm for the* FLP *if, for every instance $\mathcal{I} = (C, F, c, f)$ of the* FLP*, and for every solution $S \subseteq F$ for $\mathcal{I}$ with facility cost $f(S) = \sum_{i \in S} f_i$ and connection cost $c(S) = \sum_{j \in C} c(S, j)$, the cost of the solution produced by the algorithm is at most $\gamma_f f(S) + \gamma_c c(S)$.*

Observe that a (γ_f, γ_c)-approximation algorithm for a problem is also a γ-approximation for the problem, where $\gamma = \max(\gamma_f, \gamma_c)$. In some situations, one may take advantage of a more discriminative approximation ratio. For instance, a $(1, \alpha)$-approximation for the MFLP is a 2α-approximation for the metric k-median problem [11]. Jain *et al.* described such 2α-approximations for $\alpha = 3$ [8] and $\alpha = 2$ [9]. Mahdian *et al.* [13] used a scaling and greedy augmentation algorithm to balance a $(1.11, 1.78)$-approximation and obtain a 1.52-approximation for the MFLP. Byrka [3] combined different bifactor approximations to obtain a 1.5-approximation for the MFLP.

For the MFLP, Jain *et al.* showed that no algorithm is a (γ_f, γ_c)-approximation, with $\gamma_c < 1 + 2e^{-\gamma_f}$, unless $\mathrm{NP} \subseteq \mathrm{DTIME}[n^{\mathrm{O}(\log\log n)}]$ [9]. As the SMFLP is a generalization of the MFLP, these negative results apply also to the SMFLP. We extend these results by adapting the proof of Guha and Khuller [6] to the SMFLP as follows:

Theorem 3. *Let γ_f and γ_c be positive constants with $\gamma_c < 1 + 8e^{-\gamma_f}$. If there is a (γ_f, γ_c)-approximation for the* SMFLP*, then* $\mathrm{NP} \subseteq \mathrm{DTIME}[n^{O(\log\log n)}]$.

Proof (adapted from [6]). Assume A is a (γ_f, γ_c)-approximation for the SMFLP with $\gamma_c < 1 + 8e^{-\gamma_f}$. Let $\mathcal{J} = (\mathcal{U}, \mathcal{S})$ be an instance of the Set Cover, with $\mathcal{U}$ being a set of elements, $\mathcal{S}$ a collection of subsets of $\mathcal{U}$ and $n = |\mathcal{U}|$. We will derive a $(d' \ln n)$-approximation algorithm for the Set Cover problem, for some $d' < 1$.

Let k be the optimal value of $\mathcal{J}$. If k is not known, one can run this algorithm for $k = 1, \dots, n$ and output the best solution found.

The algorithm will find a solution for $\mathcal{J}$ by iteratively solving a sequence of instances of the SMFLP of the form $\mathcal{I}^{(j)} = (C^{(j)}, F, c, f^{(j)})$, where $F = \mathcal{S}$ and the inicial set $C^{(1)} = \mathcal{U}$. For each element $x_j \in S_i$, set $c_{ij} = 1$, and for each $x_j \notin S_i$, set $c_{ij} = 9$. Note that such c is a squared metric. Let $n_j = |C^{(j)}|$. In the jth instance, every facility cost is $f^{(j)} = \gamma \frac{n_j}{k}$, for some positive γ to be fixed later. For each j, let $S^{(j)}$ denote the solution for $\mathcal{I}^{(j)}$ produced by algorithm A and let $C^{(j+1)}$ be the elements of $C^{(j)}$ not covered by any set in $S^{(j)}$. This process stops when $C^{(j+1)} = \emptyset$ and yields the solution $S^{(1)} \cup \dots \cup S^{(j)}$ for $\mathcal{J}$.

Observe that an optimal solution for $\mathcal{J}$ is a solution for each $\mathcal{I}^{(j)}$ with total facility cost $k\, f^{(j)}$ and connection cost one for each of the n_j cities. Therefore, $S^{(j)}$ has cost at most $\gamma_f k f^{(j)} + \gamma_c n_j = (\gamma_f \gamma + \gamma_c) n_j$, because $f^{(j)} = \gamma \frac{n_j}{k}$. Let $\beta_j = |S^{(j)}|/k$ and d_j be such that $d_j n_j$ is the number of elements covered in iteration j, that is, the number of elements of $C^{(j)}$ in the union of the sets in $S^{(j)}$. Then the total facility cost of $S^{(j)}$ is $\beta_j k f^{(j)} = \beta_j \gamma n_j$. Moreover, $d_j n_j$ cities are connected with cost one and the other $n_j - d_j n_j = (1 - d_j) n_j$ cities are connected with cost nine. Hence the total cost of $S^{(j)}$ is $\beta_j \gamma n_j + d_j n_j + 9(1 - d_j) n_j = (\beta_j \gamma + 9 - 8 d_j) n_j$. We conclude that $\gamma_f \gamma + \gamma_c \;\geq\; \beta_j \gamma + 9 - 8 d_j$. So we have that $\gamma_c \;\geq\; (\beta_j - \gamma_f)\gamma + 9 - 8 d_j$.

Let $d < 1$ be such that $1 + 8e^{-\gamma_f/d} > \gamma_c$. Suppose, for the sake of contradiction, that $d_j \leq 1 - e^{-\beta_j/d}$ for some j. Then

$$\gamma_c \geq (\beta_j - \gamma_f)\gamma + 9 - 8(1 - e^{-\beta_j/d}).$$

Considering γ_f, γ and d fixed, the minimum value of the right hand side is achieved when $\beta_j = d \ln \frac{8}{d\gamma}$. Substituting β_j above, we get

$$\gamma_c \geq (d \ln \frac{8}{d\,\gamma} - \gamma_f)\gamma + 1 + d\,\gamma.$$

Considering d and γ_f fixed, we choose the value of γ that maximizes the right hand side, that is $\gamma = \frac{8}{d} e^{-\frac{\gamma_f}{d}}$. Replacing in the inequality, we obtain $\gamma_c \geq 1 + 8\, e^{-\frac{\gamma_f}{d}} > \gamma_c$, a contradiction. So $d_j > 1 - e^{-\beta_j/d}$ for every j, for this $d < 1$.

Following the lines of Guha and Khuller [6], one can prove that the algorithm described above for Set Cover is a $(d' \ln n)$-approximation for some $d' < 1$. This implies that NP $\subseteq$ DTIME$[n^{\mathrm{O}(\log \log n)}]$. $\square$

Let $\alpha < 1 + W_0(8e^{-1})$, where $W_0(z)$ is the solution of equation $z = xe^x$. We observe that there is no α-approximation for the SMFLP, unless NP $\subseteq$ DTIME$[n^{\mathrm{O}(\log \log n)}]$. To see this, notice that an α-approximation implies a (γ_f, γ_c)-approximation where $\gamma_f = 1 + W_0(8e^{-1})$ and $\gamma_c = \alpha < 1 + W_0(8e^{-1}) = 1 + 8e^{-\gamma_f}$; so the above theorem states that NP $\subseteq$ DTIME$[n^{\mathrm{O}(\log \log n)}]$. In particular, $2.04 < 1 + W_0(8e^{-1})$, so there is no 2.04-approximation for the SMFLP unless NP $\subseteq$ DTIME$[n^{\mathrm{O}(\lg \lg n)}]$. We may strengthen this result [16] to show that there is no 2.04-approximation for the SMFLP unless P = NP.

3 Constant ratio approximation for the Squared Metric FLP

The metric property is used only in the analysis of the algorithms of Jain *et al.* [9]. In other words, their algorithms can be applied to general FLP instances. However, the perfomance guarantee is only proved to hold for the MFLP instances. In this section, we will prove that their first algorithm, that we denote by A1, is a 2.87-approximation for the SMFLP. For the sake of completeness, their algorithm is described next.

Algorithm A1 (C, F, c, f) [9]

1. Set $U := C$, meaning that every facility starts unopened, and every city unconnected. Each city j has some budget α_j, initially 0, and, at every moment, the budget that an unconnected city j offers to some unopened facility i equals to $\max(\alpha_j - c_{ij}, 0)$.

2. While $U \neq \emptyset$, the budget of each unconnected city is increased continuously until one of the following events occur:

   (a) For some unconnected city j and some open facility i, $\alpha_j = c_{ij}$. In this case, connect city j to facility i and remove j from U.

   (b) For some unopened facility i, $\sum_{j \in U} \max(\alpha_j - c_{ij}, 0) = f_i$. In this case, open facility i and, for every unconnected city j with $\alpha_j \geq c_{ij}$, connect j to i and remove it from U.

The analysis presented by Jain *et al.* [9] uses the dual fitting method. That is, their algorithms produce not only a solution for the MFLP, but also a vector $\alpha = (\alpha_1, \ldots, \alpha_{|C|})$ such that the value of the solution produced is equal to $\sum_j \alpha_j$. Moreover, for the first algorithm, following the dual fitting method, Jain *et al.* [9] proved that the vector $\alpha/1.861$ is a feasible solution for the dual linear program presented as (3) in [9], concluding that the algorithm is a 1.861-approximation for the MFLP. To present a similar analysis for the

SMFLP, we use the same definitions and follow closely the steps of Jain *et al.* analysis. We start by adapting Lemma 3.2 from [9] for a squared metric.

Lemma 4. *For every facility i, cities j and j', and vector α obtained by the first algorithm of Jain* et al. *[9] given an instance of the* SMFLP*,*

$$\sqrt{\alpha_j} \leq \sqrt{\alpha_{j'}} + \sqrt{c_{ij'}} + \sqrt{c_{ij}}.$$

Proof. If $\alpha_j \leq \alpha_{j'}$, the inequality obviously holds. So assume $\alpha_j > \alpha_{j'}$. Let i' be the facility to which the algorithm connects city j'. Thus $\alpha_{j'} \geq c_{i'j'}$ and facility i' is open at time $\alpha_{j'} < \alpha_j$. If $\alpha_j > c_{i'j}$, then city j would have been connected to facility i' at some time $t \leq \max(\alpha_{j'}, c_{i'j}) < \alpha_j$, and α_j would have stopped growing then, a contradiction. Hence $\alpha_j \leq c_{i'j}$. Furthermore, by the squared metric constraint, $\sqrt{c_{i'j}} \leq \sqrt{c_{i'j'}} + \sqrt{c_{ij'}} + \sqrt{c_{ij}}$. Therefore $\sqrt{\alpha_j} \leq \sqrt{\alpha_{j'}} + \sqrt{c_{ij'}} + \sqrt{c_{ij}}$. ☐

From Lemmas 1 and 4, we derive the following.

Corollary 5. *For every positive β, γ, and δ, and for every facility i, cities j and j', and the vector α produced by the first algorithm of Jain* et al. *[9] given an instance of the* SMFLP*, $\alpha_j \leq (1+\beta+\frac{1}{\gamma})\alpha_{j'} + (1+\gamma+\frac{1}{\delta})c_{ij'} + (1+\delta+\frac{1}{\beta})c_{ij}$.*

A facility i is said to be γ-*overtight* for some positive γ if, at the end of the algorithm,

$$\sum_j \max\left(\frac{\alpha_j}{\gamma} - c_{ij}, 0\right) \ \leq \ f_i. \tag{1}$$

Observe that, if every facility is γ-overtight, then the vector α/γ is a feasible solution for the dual linear program presented as (3) in [9]. Jain *et al.* proved that, for the MFLP, every facility is 1.861-overtight. We want to find a γ for the SMFLP, as close to 1 as possible, for which every facility is γ-overtight.

Fix a facility i. Let us assume without loss of generality that $\alpha_j \geq \gamma c_{ij}$ only for the first k cities. Following the lines of Jain *et al.* [9], we want to obtain the so called *factor-revealing* program. We define a set of variables f, d_j, and α_j, corresponding to facility cost f_i, distance c_{ij}, and city contribution α_j. Then, we capture the intrinsic properties of the algorithm using constraints over these variables. We assume without loss of generality that $\alpha_1 \leq \cdots \leq \alpha_k$. Also, we use Lemma 3.3 from [9], that states that the total contribution offered to a facility at any time is at most its cost, that is, $\sum_{l=j}^{k} \max(\alpha_j - d_l, 0) \leq f$. Besides these, we have the constraints from Lemma 4. Subject to all of these constraints, we want to find the minimum γ such that the facility is γ-overtight. In terms of the defined variables, we want the maximum ratio $\sum_{j=1}^{k} \alpha_j/(f + \sum_{j=1}^{k} d_j)$. We obtain the following maximization program:

$$\begin{array}{rcll} z_k^{A1} = & \max & \frac{\sum_{j=1}^{k}\alpha_j}{f+\sum_{j=1}^{k} d_j} & \\ & \text{s.t.} & \alpha_j \leq \alpha_{j+1} & \forall 1 \leq j < k \\ & & \sqrt{\alpha_j} \leq \sqrt{\alpha_l} + \sqrt{d_j} + \sqrt{d_l} & \forall\ 1 \leq j, l \leq k \\ & & \sum_{l=j}^{k} \max(\alpha_j - d_l, 0) \leq f & \forall\ 1 \leq j \leq k \\ & & \alpha_j, d_j, f \geq 0 & \forall\ 1 \leq j \leq k. \end{array} \tag{2}$$

The next lemma has the same statement of Lemma 3.4 in [9], but it refers to program (2). Since the proof is the same, we omit it.

Lemma 6. *Let* $\gamma = \sup_{k\geq 1} z_k^{A1}$*. Every facility is* γ*-overtight.*

Therefore $\sup_{k\geq 1} z_k^{A1}$ is an upper bound on the approximation factor of the algorithm for the SMFLP. A slight modification of the example presented in Theorem 3.5 of [9] shows that this upper bound is tight.

3.1 A first analysis

Our first step is to relax (2) into a linear program. For that, we adjust the objective function as in [9], and we approximate the squared metric property by using inequalities given by Corollary 5. First, we will use only the inequalities corresponding to $\beta = \gamma = \delta = 1$. With this, we will prove that $\sup_{k\geq 1} z_k^{A1}$ is not greater than 3.236. Later, we will improve the obtained result by using a whole set of inequalities from Corollary 5, and using a more standard factor-revealing analysis for the SMFLP. The first relaxed factor-revealing linear program is:

$$\begin{array}{rlll} \dot{w}_k = & \max & \sum_{j=1}^{k} \alpha_j & \\ & \text{s.t.} & f + \sum_{j=1}^{k} d_j \leq 1 & \\ & & \alpha_j \leq \alpha_{j+1} & \forall\, 1 \leq j < k \\ & & \alpha_j \leq 3\alpha_l + 3d_j + 3d_l & \forall\, 1 \leq j, l \leq k \\ & & x_{jl} \geq \alpha_j - d_l & \forall\, 1 \leq j \leq l \leq k \\ & & \sum_{l=j}^{k} x_{jl} \leq f & \forall\, 1 \leq j \leq k \\ & & \alpha_j, d_j, f \geq 0 & \forall\, 1 \leq j \leq k. \end{array} \tag{3}$$

As (3) is a relaxation of (2), we have that $z_k^{A1} \leq \dot{w}_k$ and thus an upper bound on $\sup_{k\geq 1} \dot{w}_k$ is also an upper bound on $\sup_{k\geq 1} z_k^{A1}$. Solving linear program (3) using CPLEX for $k = 540$, we obtain the next lemma.

Lemma 7. $\sup_{k\geq 1} \dot{w}_k \geq 3.220$*.*

To obtain an upper bound to their factor-revealing linear program, Jain *et al.* [9] presented a general solution to the dual of their factor-revealing linear program. This dual solution is deduced from computational experiments and empirical results for small values of k. First, they relaxed each constraint corresponding to the cities contribution, that is, $\sum_{l=j}^{k} \max(\alpha_j - d_l, 0) \leq f$, obtaining $\sum_{l=j}^{l_j} (\alpha_j - d_l) \leq f$, for some value l_j estimated computationally. Then, they multiplied each of these inequalities by a variable θ_j and added them up, combining them with the other inequalities to obtain a bound on the optimal value. Variables θ_j play the role of dual variables. They also argue that doubling (and scaling down) a solution of their factor-revealing linear program for k gives rise to a feasible solution for $2k$ with the same value. Thus one can obtain an upper bound on the optimal value of this linear program assuming k is large enough.

We have replayed the analysis of their algorithm for the SMFLP. In their approach, they defined l_j and θ_j as 2- and 3-step functions specified by parameters p_1 and p_2. Experimentally one can see that such choices are natural. Then, with straightforward calculations, they obtained an upper bound as a function of p_1 and p_2, and adjusted these parameters to prove the best possible bound on the approximation factor. Using a similar approach for the squared metric case, that is, using step functions for l_j and θ_j with small number of steps, we obtained a factor not better than 3.625 for SMFLP. We managed to improve the obtained factor to 3.512 by using a piecewise function for θ_j whose pieces are either constant or hyperboles on j. Different choices of functions l_j and θ_j lead to different approximation guarantees. We have tried several choices based on empirical observations. For instance, observing the primal general solution may suggest that defining l_j and θ_j as 3- and 4-step functions is a good choice, but this does not improve the 3.625 bound.

Inspired on this process, we looked for an alternative analysis of program (3). Instead of defining explicitly the value of each θ_j, we multiply each inequality of this program by variables and add them up. Then, we try to adjust the value of these variables to obtain the best upper bound. This is done through a linear program subject to some desired constraints, with the objective of achieving the smallest upper bound on the approximation factor. Unfortunately, this linear program can be arbitrarily large. We deal with this situation by choosing an appropriate value of k and exploiting a special property of program (3). The next lemma shows that $\dot{w}_k$ does not decrease for multiples of k.

Lemma 8. *For every k and every t, $\dot{w}_k \leq \dot{w}_{tk}$.*

Proof. It is enough to make t replicas of an optimal solution for k, and then scale down the variables by $1/t$, to obtain a feasible solution of the linear program for tk with objective value $\dot{w}_{tk} = \dot{w}_k$. □

In the following, we use a linear program to give a very tight bound on $\sup_{k\geq 1} \dot{w}_k$.

Lemma 9. *For every k, $\dot{w}_k \leq 3.236$.*

Proof. In what follows, we deduce an upper bound on $\dot{w}_k$ by deriving a linear minimization program whose feasible solutions are upper bounds on $\dot{w}_k$. Then we present a feasible solution of value less than 3.236 for this program. The idea is to determine a conical combination of the inequalities of (3) that imply inequality (1) for a γ as small as possible. The linear minimization program will help us to choose the coefficients of such conical combination.

Let us rewrite the third inequality of program (3), so that the right-hand side is zero. For each j and l, we multiply the corresponding inequality by φ_{jl}. Denote by A the sum of all these inequalities, that is,

$$\sum_{j=1}^{k}\sum_{l=1}^{k} \varphi_{jl}(\alpha_j - 3\alpha_l - 3d_l - 3d_j) \leq 0.$$

The fourth and fifth inequalities of program (3) can be relaxed to the set of inequalities $\sum_{i=j}^{l_j}(\alpha_j - d_i) \leq f$, one for each l_j such that $j \leq l_j \leq k$. For each j and l_j, we multiply the corresponding inequality by θ_{jl_j} and denote by B the inequality resulting of summing them up, that is,

$$\sum_{j=1}^{k}\sum_{l=j}^{k} \theta_{jl} \textstyle\sum_{i=j}^{l}(\alpha_j - d_i) \leq \left(\displaystyle\sum_{j=1}^{k}\sum_{l=j}^{k}\theta_{jl}\right) f.$$

The coefficients of α_j in A and B are, respectively,

$$\text{coeff}_A[\alpha_j] = \sum_{l=1}^{k}(\varphi_{jl} - 3\varphi_{lj}) \qquad \text{and} \qquad \text{coeff}_B[\alpha_j] = \sum_{l=j}^{k}(l - j + 1)\theta_{jl},$$

and the coefficients of $-d_j$ in A and B are, respectively,

$$\text{coeff}_A[-d_j] = \sum_{l=1}^{k} 3(\varphi_{jl} + \varphi_{lj}) \qquad \text{and} \qquad \text{coeff}_B[-d_j] = \sum_{i=1}^{j}\sum_{l=j}^{k} \theta_{il}.$$

Now, we sum inequalities A and B and obtain a new inequality C:

$$\sum_{j=1}^{k} \text{coeff}_C[\alpha_j]\ \alpha_j - \sum_{j=1}^{k} \text{coeff}_C[-d_j]\ d_j \leq \text{coeff}_C[f]\ f. \tag{4}$$

We want to find γ, θ_{jl}, and φ_{jl} so that the corresponding coefficients of C are such that inequality (4) implies, for suficiently large k, that

$$\sum_{j=1}^{k} \alpha_j - \gamma \sum_{j=1}^{k} d_j \leq \gamma f. \tag{5}$$

Moreover, we want γ as small as possible. To obtain inequality (5) from inequality (4), it is enough that, for each j, coefficient $\text{coeff}_C[\alpha_j] \geq 1$, $\text{coeff}_C[-d_j] \leq \gamma$, and $\text{coeff}_C[f] \leq \gamma$. Hence, this can be expressed by the following linear program.

$$\begin{array}{rlll} y_k = & \min & \gamma & \\ & \text{s.t.} & \text{coeff}_C[\alpha_j] \geq 1 & \forall\ 1 \leq j \leq k \\ & & \text{coeff}_C[-d_j] \leq \gamma & \forall\ 1 \leq j \leq k \\ & & \text{coeff}_C[f] \leq \gamma & \\ & & \varphi_{jl} \geq 0 & \forall\ 1 \leq j, l \leq k \\ & & \theta_{jl} \geq 0 & \forall\ 1 \leq j \leq l \leq k. \end{array} \tag{6}$$

The interested reader may observe that this program is the dual of a relaxed version of the factor-revealing linear program (3). Therefore, its optimal value is an upper bound on the optimal value of (3).

Using Lemma 8, we may assume that k has the form $k = pt$ with p and t positive integers. We will use a scaling argument to create a linear minimization program with a small number of variables, and obtain a feasible solution for program (6) from a solution of the former program. Then, we will show that the value of the generated solution is bounded by the value of the small solution.

Consider variables $\gamma' \in \mathbb{R}_+$, $\varphi'_{jl} \in \mathbb{R}_+$ for $1 \leq j, l \leq t$, and $\theta'_{jl} \in \mathbb{R}_+$ for $1 \leq j \leq l \leq t$. For an arbitrary n, let $\hat{n} = \lceil \frac{n}{p} \rceil$. We obtain a candidate solution for program (6) by taking

$$\varphi_{jl} = \frac{\varphi'_{\hat{j}\hat{l}}}{p}, \qquad \theta_{jl} = \frac{\theta'_{\hat{j}\hat{l}}}{p^2} \qquad \text{and} \qquad \gamma = \gamma'. \tag{7}$$

Let us calculate each coefficient of C for this solution.

$$\begin{aligned}
\mathrm{coeff}_C[\alpha_j] &= \sum_{l=1}^{k}(\varphi_{jl} - 3\varphi_{lj}) + \sum_{l=j}^{k}(l-j+1)\theta_{jl} \\
&= \sum_{l=1}^{k}(\frac{\varphi'_{\hat{j}\hat{l}}}{p} - 3\frac{\varphi'_{\hat{l}\hat{j}}}{p}) + \sum_{l=j}^{k}(l-j+1)\frac{\theta'_{\hat{j}\hat{l}}}{p^2} \\
&\geq \sum_{l=1}^{pt}(\frac{\varphi'_{\hat{j}\hat{l}}}{p} - 3\frac{\varphi'_{\hat{l}\hat{j}}}{p}) + \sum_{\boldsymbol{l=p\,\hat{j}+1}}^{pt}(l-p\,\hat{j})\frac{\theta'_{\hat{j}\hat{l}}}{p^2} \\
&= \sum_{l'=1}^{t} p(\frac{\varphi'_{\hat{j}l'}}{p} - 3\frac{\varphi'_{l'\hat{j}}}{p}) + \sum_{l'=\hat{j}+1}^{t}\frac{\theta'_{\hat{j}l'}}{p^2}\sum_{i=0}^{p-1}(p\,l' - i - p\,\hat{j}) \\
&= \sum_{l'=1}^{t}(\varphi'_{\hat{j}l'} - 3\varphi'_{l'\hat{j}}) + \sum_{l'=\hat{j}+1}^{t}\frac{\theta'_{\hat{j}l'}}{p^2}(p^2\,l' - \frac{p(p-1)}{2} - p^2\,\hat{j}) \\
&\geq \sum_{l'=1}^{t}(\varphi'_{\hat{j}l'} - 3\varphi'_{l'\hat{j}}) + \sum_{l'=\hat{j}+1}^{t}(l' - \hat{j} - \frac{1}{2})\theta'_{\hat{j}l'}.
\end{aligned}$$

$$\begin{aligned}
\mathrm{coeff}_C[-d_j] &= \sum_{l=1}^{k} 3(\varphi_{jl} + \varphi_{lj}) + \sum_{i=1}^{j}\sum_{l=j}^{k}\theta_{il} \\
&= \sum_{l=1}^{pt} 3(\frac{\varphi'_{\hat{j}\hat{l}}}{p} + \frac{\varphi'_{\hat{l}\hat{j}}}{p}) + \sum_{i=1}^{j}\sum_{l=j}^{pt}\frac{\theta'_{\hat{i}\hat{l}}}{p^2} \\
&\leq \sum_{l'=1}^{t} p\cdot 3(\frac{\varphi'_{\hat{j}l'}}{p} + \frac{\varphi'_{l'\hat{j}}}{p}) + \sum_{i'=1}^{\boldsymbol{\hat{j}}} p\cdot \sum_{\boldsymbol{l'=\hat{j}}}^{t} p\cdot \frac{\theta'_{i'l'}}{p^2} \\
&= \sum_{l'=1}^{t} 3(\varphi'_{\hat{j}l'} + \varphi'_{l'\hat{j}}) + \sum_{i'=1}^{\hat{j}}\sum_{l'=\hat{j}}^{t}\theta'_{i'l'}.
\end{aligned}$$

$$\mathrm{coeff}_C[f] = \sum_{j=1}^{k}\sum_{l=j}^{k}\theta_{jl} = \sum_{j=1}^{pt}\sum_{l=j}^{pt}\frac{\theta'_{\hat{j}\hat{l}}}{p^2} \leq \sum_{j'=1}^{t} p\cdot \sum_{\boldsymbol{l'=\hat{j}}}^{t} p\cdot \frac{\theta'_{j'l'}}{p^2} = \sum_{j'=1}^{t}\sum_{l'=\hat{j}}^{t}\theta'_{j'l'}.$$

Now, we want to find the minimum value of γ' and variables φ'_{jl} and θ'_{jl} such that the candidate solution for program (6) is feasible. We may define the following linear program.

$$\begin{array}{rlll}
\dot{x}_t = & \min & \gamma' & \\
& \text{s.t.} & \sum_{l=1}^{t}(\varphi'_{jl} - 3\varphi'_{lj}) + \sum_{l=j+1}^{t}(l-j-\frac{1}{2})\theta_{jl} \geq 1 & \forall\, 1 \leq j \leq t \\
& & \sum_{l=1}^{t} 3(\varphi'_{jl} + \varphi'_{lj}) + \sum_{i=1}^{j}\sum_{l=j}^{t}\theta'_{il} \leq \gamma' & \forall\, 1 \leq j \leq t \\
& & \sum_{j=1}^{t}\sum_{l=j}^{t}\theta'_{jl} \leq \gamma' & \\
& & \varphi'_{jl} \geq 0 & \forall\, 1 \leq j, l \leq t \\
& & \theta'_{jl} \geq 0 & \forall\, 1 \leq j \leq l \leq t.
\end{array} \tag{8}$$

Consider an optimal solution for program (6). Replacing it in inequality (4), that is, inequality C, we obtain $\sum_{j=1}^{k} \alpha_j - \gamma \sum_{j=1}^{k} d_j \le \gamma f$. Therefore, $\dot{w}_k \le \gamma = y_k$. Now, consider an optimal solution for program (8) and the corresponding generated solution for program (6). We obtain $y_k \le \gamma = \gamma' = \dot{x}_t$ and conclude that $\dot{w}_k \le \dot{x}_t$.

Using CPLEX to solve program (8), we obtained $\dot{x}_{800} \approx 3.23586 < 3.236$, and this concludes the proof of the theorem. $\square$

3.2 An improved factor-revealing analysis

In Lemma 9, we obtained the minimization program (8) from a conical combination of constraints from program (3). The optimal value of this minimization program is an upper bound on the approximation factor. The calculations involved are very similar to those used to obtain the corresponding dual program. We propose a new standard factor-revealing technique, which provides a more straightforward way to obtain a bound on the approximation factor.

Consider the traditional maximization factor-revealing linear program. First, the dual program is obtained for some k. Take k in the form $k = pt$, for a fixed t. We will create a minimization program that mimics the constraints of the dual, but depends only on t and bounds its optimal value for every k. The idea is to constrain the variables of the small program to obtain a feasible solution for the dual program. To obtain a linear program independent of k, we will scale the variables by p. Since any solution of such a program reveals an upper bound on the approximation factor, we call it an *upper bound factor-revealing program.*

In order to derive a better upper bound factor-revealing linear program, we will use a whole set of linear inequalities to approximate the nonlinear constraint in (2). Consider tuples $(\beta_i, \gamma_i, \delta_i)$ of positive real numbers and $B_i = 1 + \beta_i + \frac{1}{\gamma_i}$, $C_i = 1 + \gamma_i + \frac{1}{\delta_i}$, $D_i = 1 + \delta_i + \frac{1}{\beta_i}$ for $1 \le i \le m$. Using Corollary 5, we insert inequalities corresponding to the given tuples, replacing the nonlinear constraint, and obtain $z_k^{A1} \le w_k^{A1}$, where w_k^{A1} is given by

$$
\begin{array}{rll}
w_k^{A1} = & \max & \sum_{j=1}^{k} \alpha_j \\
& \text{s.t.} & f + \sum_{j=1}^{k} d_j \le 1 \\
& & \alpha_j \le \alpha_{j+1} \qquad \forall\, 1 \le j < k \\
& & \alpha_j \le B_i \alpha_l + C_i d_j + D_i d_l \qquad \forall\, 1 \le j, l \le k, \quad 1 \le i \le m \\
& & x_{jl} \ge \alpha_j - d_l \qquad \forall\, 1 \le j \le l \le k \\
& & \sum_{l=j}^{k} x_{jl} \le f \qquad \forall\, 1 \le j \le k \\
& & \alpha_j, d_j, f \ge 0 \qquad \forall\, 1 \le j \le k.
\end{array}
\tag{9}
$$

The following lemma gives a lower bound on the approximation factor of the algorithm for the SMFLP.

Lemma 10. $\sup_{k \ge 1} z_k^{A1} \ge 2.86$.

Proof. Although program (2) contains nonlinear constraints, we may use linear program packages to solve it. We start by solving program (9) with a fixed number of inequalities. Then, we employ a cutting plane insertion strategy: if the obtained solution violates the squared metric property, we derive a cutting plane using Lemma 1, and resolve the linear program with this additional constraint. Using CPLEX with the cutting plane strategy we obtained $z^{A1}_{700} \approx 2.86099 > 2.86$. $\square$

Now, we can bound the approximation factor of the algorithm using an upper bound factor-revealing program.

Lemma 11. $\sup_{k \geq 1} z^{A1}_k \leq 2.87$.

Proof. It is easy to see that the proof of Lemma 8 is also valid for program (9). So, from Lemma 1, we have that $z^{A1}_k \leq w^{A1}_{kt}$, for every positive integer t. We assume that k has the form $k = pt$, with p and t positive integers. The dual of the linear program (9) is

$$
\begin{array}{rlll}
w^{A1}_k = \min & \gamma & & \\
\text{s.t.} & a_j - a_{j-1} + \sum\limits_{i=1}^{m} \sum\limits_{l=1}^{k} c_{jli} - \sum\limits_{i=1}^{m} B_i \sum\limits_{l=1}^{k} c_{lji} + \sum\limits_{l=j}^{k} e_{jl} \geq 1 & \forall & 1 \leq j \leq k \\
& \sum\limits_{i=1}^{m} C_i \sum\limits_{l=1}^{k} c_{jli} + \sum\limits_{i=1}^{m} D_i \sum\limits_{l=1}^{k} c_{lji} + \sum\limits_{l=1}^{j} e_{lj} \leq \gamma & \forall & 1 \leq j \leq k \\
& \sum\limits_{j=1}^{k} h_j \leq \gamma & & \\
& e_{jl} \leq h_j & \forall & 1 \leq j \leq l \leq k \\
& a_0 = a_k = 0, a_j, h_j, e_{jl}, c_{jli} \geq 0 & \forall & \begin{array}{l} 1 \leq j, l \leq k \\ 1 \leq i \leq m. \end{array}
\end{array}
\tag{10}
$$

We can derive the upper bound factor-revealing linear program. We would like to define variables as in equation (7). Just using a scale factor is not sufficient to preserve the variables a_j in program (10). The variables a_j correspond to the ordering restrictions of primal variables α_j in program (9), and computational experiments have indicated that removing such restrictions does not change the optimal value significantly, for large values of k. So, we could just set $a_j = 0$ for all j. However, we want to preserve such restrictions, as they will shortly be needed to prove Lemma 13. To do this, we can interpolate the variables of the upper bound factor-revealing program to obtain the variables of the lower bound program. Again, we group sets of variables based on their indices. For that, we denote the group of a variable of index n as $\hat{n}$. Let $\hat{n} = \lceil \frac{n}{p} \rceil$ and consider variables $\gamma', a'_j, c'_{jli}, e'_{jli}, h'_j$. We obtain a candidate solution for program (10) by defining

$$
\gamma = \gamma', \quad a_j = p\, a'_{\hat{j}}(p\,\hat{j} - j)(a'_{\hat{j}} - a'_{\hat{j}-1}), \quad c_{jli} = \frac{c'_{\hat{j}\hat{l}i}}{p}, \quad e_{jl} = \frac{e'_{\hat{j}\hat{l}}}{p}, \quad \text{and} \quad h_j = \frac{h'_{\hat{j}}}{p}. \tag{11}
$$

In the following, we will use definition (11) to obtain a candidate solution for program (10) from a small set of prime variables. Then, for each constraint of program (10), we obtain the expression formed by the dependent terms, and calculate it as a function of the considered variables. Notice that there is an expression for each primal variable of program (9). These expressions are analogous to the primal variables coefficients used in Lemma 9, thus, for each primal variable x, we say that this is the *coefficient expression* for x, and we will denote it by coeff$[x]$.

Now we create the minimization upper bound factor-revealing program. The objective value is obtained by applying definition (11) to the objective value of program (10). Then, for each group of coefficient expressions that has the same value, we include a constraint in the upper bound program that bounds the expression by the independent term. Notice that each constraint of the upper bound factor-revealing linear program may correspond to an arbitrarily large number of constraints of the factor-releaving linear program. In the following, we calculate and bound each coefficient expression.

First notice that $a_j - a_{j-1} = a'_{\hat{j}} - a'_{\hat{j}-1}$. To see this, it is enough to use definition (11) and consider the cases $\hat{j} = \hat{(j-1)}$, and $\hat{j} = \hat{(j-1)} + 1$. Now we have:

$$
\begin{aligned}
\text{coeff}[\alpha_j] &= a_j - a_{j-1} + \sum_{i=1}^{m} \sum_{l=1}^{k} c_{jli} - \sum_{i=1}^{m} B_i \sum_{l=1}^{k} c_{lji} + \sum_{l=j}^{k} e_{jl} \\
&= a'_{\hat{j}} - a'_{\hat{j}-1} + \sum_{i=1}^{m} \sum_{l=1}^{pt} \frac{c'_{\hat{j}\hat{l}i}}{p} - \sum_{i=1}^{m} B_i \sum_{l=1}^{pt} \frac{c'_{\hat{l}\hat{j}i}}{p} + \sum_{l=j}^{pt} \frac{e'_{\hat{j}\hat{l}}}{p} \\
&\geq a'_{\hat{j}} - a'_{\hat{j}-1} + \sum_{i=1}^{m} \sum_{l'=1}^{t} p \frac{c'_{\hat{j}l'i}}{p} - \sum_{i=1}^{m} B_i \sum_{l'=1}^{t} p \frac{c'_{l'\hat{j}i}}{p} + \sum_{\boldsymbol{l'=\hat{j}+1}}^{t} p \frac{e'_{\hat{j}l'}}{p} \\
&= a'_{\hat{j}} - a'_{\hat{j}-1} + \sum_{i=1}^{m} \sum_{l'=1}^{t} c'_{\hat{j}l'i} - \sum_{i=1}^{m} B_i \sum_{l'=1}^{t} c'_{l'\hat{j}i} + \sum_{l'=\hat{j}+1}^{t} e'_{\hat{j}l'} \geq 1.
\end{aligned}
$$

$$
\begin{aligned}
\text{coeff}[d_j] &= \gamma - \sum_{i=1}^{m} C_i \sum_{l=1}^{k} c_{jli} - \sum_{i=1}^{m} D_i \sum_{l=1}^{k} c_{lji} - \sum_{l=1}^{j} e_{lj} \\
&= \gamma' - \sum_{i=1}^{m} C_i \sum_{l=1}^{pt} \frac{c'_{\hat{j}\hat{l}i}}{p} - \sum_{i=1}^{m} D_i \sum_{l=1}^{pt} \frac{c'_{\hat{l}\hat{j}i}}{p} - \sum_{l=1}^{j} \frac{e'_{\hat{l}\hat{j}}}{p} \\
&\geq \gamma' - \sum_{i=1}^{m} C_i \sum_{l'=1}^{t} p \frac{c'_{\hat{j}l'i}}{p} - \sum_{i=1}^{m} D_i \sum_{l'=1}^{t} p \frac{c'_{l'\hat{j}i}}{p} - \sum_{l'=1}^{\boldsymbol{\hat{j}}} p \frac{e'_{l'\hat{j}}}{p} \\
&= \gamma' - \sum_{i=1}^{m} C_i \sum_{l'=1}^{t} c'_{\hat{j}l'i} - \sum_{i=1}^{m} D_i \sum_{l'=1}^{t} c'_{l'\hat{j}i} - \sum_{l'=1}^{\hat{j}} e'_{l'\hat{j}} \geq 0.
\end{aligned}
$$

$$
\text{coeff}[f] = \gamma - \sum_{j=1}^{k} h_j = \gamma' - \sum_{j=1}^{pt} \frac{h'_{\hat{j}}}{p} = \gamma' - \sum_{j'=1}^{t} p \frac{h'_{j'}}{p} = \gamma' - \sum_{j'=1}^{t} h'_{j'} \geq 0.
$$

$$
\text{coeff}[x_{jl}] = h_j - e_{jl} = \frac{h'_{\hat{j}}}{p} - \frac{e'_{\hat{j}\hat{l}}}{p} \geq 0.
$$

We notice that for each primal variable, the constraint for its coefficient expression is equivalent to the constraint of any other primal variable in the same group. For example, for any pair α_j and α_l, such that $\hat{j} = \hat{l}$, we need to add only one constraint to the upper bound factor-revealing program; therefore, we need only t constraints for this kind of primal variable. We remark that the constraint obtained for coeff$[x_{jl}]$ does not depend on p. Conjoining all different constraints, and fixing variables as a'_1 and a'_t to zero, we obtain program (12).

$$
\begin{array}{rlll}
x_t^{A1} = \min & \gamma & & \\
\text{s.t.} & a_j - a_{j-1} + \sum\limits_{i=1}^{m}\sum\limits_{l=1}^{t} c_{jli} - \sum\limits_{i=1}^{m} B_i \sum\limits_{l=1}^{t} c_{lji} + \sum\limits_{\boldsymbol{l=j+1}}^{t} e_{jl} \geq 1 & \forall & 1 \leq j \leq t \\
& \sum\limits_{i=1}^{m} C_i \sum\limits_{l=1}^{t} c_{jli} + \sum\limits_{i=1}^{m} D_i \sum\limits_{l=1}^{t} c_{lji} + \sum\limits_{l=1}^{j} e_{lj} \leq \gamma & \forall & 1 \leq j \leq t \\
& \sum\limits_{j=1}^{t} h_j \leq \gamma & & \\
& e_{jl} \leq h_j & \forall & 1 \leq j \leq l \leq t \\
& a_0 = a_k = 0, a_j, h_j, e_{jl}, c_{jli} \geq 0 & \forall & \begin{array}{l} 1 \leq j, l \leq t \\ 1 \leq i \leq m. \end{array}
\end{array}
\tag{12}
$$

Now, we want to use Lemma 1 and choose a set of tuples (β, γ, δ), so that the squared metric is minimally relaxed. To accommodate the premises of Lemma 1, we solve the dual of the upper bound factor-revealing LP, so we may use the same cutting plane strategy used in Lemma 10. The dual is given in the following.

$$
\begin{array}{rlll}
x_t^{A1} = & \max & \sum_{j=1}^{t} \alpha_j & \\
& \text{s.t.} & f + \sum_{j=1}^{t} d_j \leq 1 & \\
& & \alpha_j \leq \alpha_{j+1} & \forall\, 1 \leq j < t \\
& & \alpha_j \leq B_i \alpha_l + C_i d_j + D_i d_l & \forall\, 1 \leq j, l \leq t, \quad 1 \leq i \leq m \\
& & x_{jl} \geq \alpha_j - d_l & \forall\, 1 \leq j < l \leq t \\
& & \sum_{l=j}^{t} x_{jl} \leq f & \forall\, 1 \leq j \leq t \\
& & \alpha_j, d_j, f, x_{jl} \geq 0 & \forall\, 1 \leq j, l \leq t.
\end{array}
\tag{13}
$$

Using the cutting plane strategy with CPLEX we obtain $x_{700}^{A1} \approx 2.8697 < 2.87$. $\square$

If we apply this analysis for the metric case, we obtain an upper bound factor-revealing program similar to program (13). The only difference is that, for the metric case, there are no coefficients B_l, C_l and D_l. We use this modified linear program to tighten the approximation factor for the metric case.

Lemma 12. *For the* MFLP*, the approximation factor of A1 [9] is between* 1.814 *and* 1.816*.*

Proof. Let $\hat{z}_k^{A1}$ be the optimal value of the lower bound factor-revealing program (5) in [9]. The corresponding upper bound factor-revealing program is:

$$
\begin{array}{rlll}
\hat{x}_t^{A1} = & \max & \sum_{j=1}^{t} \alpha_j & \\
& \text{s.t.} & f + \sum_{j=1}^{t} d_j \leq 1 & \\
& & \alpha_j \leq \alpha_{j+1} & \forall\, 1 \leq j < t \\
& & \alpha_j \leq \alpha_l + d_j + d_l & \forall\, 1 \leq j, l \leq t \\
& & x_{jl} \geq \alpha_j - d_l & \forall\, 1 \leq j < l \leq t \\
& & \sum_{l=j}^{t} x_{jl} \leq f & \forall\, 1 \leq j \leq t \\
& & \alpha_j, d_j, f, x_{jl} \geq 0 & \forall\, 1 \leq j, l \leq t.
\end{array}
\tag{14}
$$

Numerical computations using CPLEX show that $\hat{z}_{1000}^{A1} \approx 1.81412 > 1.814$, and that $\hat{x}_{1000}^{A1} \approx 1.81584 < 1.816$. $\square$

We notice that the only difference between the upper and lower bound factor-revealing programs is that the upper bound factor-revealing program does not contain the restrictions $\alpha_j - d_j \leq x_{jj}$ for all j. We exploit the similarity between these programs to bound the gap between their optimal values. The following lemma is valid for both the metric and squared metric cases.

Lemma 13. *Let* z_k^{A1} *be the optimal value of the lower bound factor-revealing program* (9) *(program (5) in [9]) and let* $(\boldsymbol{\alpha}, \mathbf{d}, \mathbf{x}, \mathbf{f})$ *be an optimal solution for program* (13) *(respectively program* (14)*) with cost value* x_k^{A1}*. If* $\varepsilon = \max_j\{\alpha_j - d_j\}$*, then* $z_k^{A1} \geq \frac{1}{1+\varepsilon} x_k^{A1}$*.*

Proof. Consider a candidate solution for the lower bound factor-revealing program formed by identical $(\boldsymbol{\alpha}, \mathbf{d}, \mathbf{x}', \mathbf{f}')$, such that $f' = f + \varepsilon$, $x'_{jl} = x_{jl}$ if $j \neq l$, and $x'_{jj} = \alpha_j - d_j$. Clearly, this solution has an objective value of x_k^{A1}, and it violates only the first restriction of program (9) (program (5) in [9], respectively). For that, we get $f' + \sum_{j=1}^{k} d_j = 1 + \varepsilon > 1$. Now, it is enough to multiply each variable by $\frac{1}{1+\varepsilon}$, and obtain a feasible solution. □

From the last lemma, it is clear that the upper and lower bound factor-revealing programs yield close values, except for an error factor that depends only on the variable values of an optimal solution for the upper bound factor-revealing program. Since the optimal value decreases as the number of variables k increases, it is reasonable to expect that the value of both factor-revealing programs get very close as k tends to infinity. Indeed, for the metric case, it is easy to show that this error vanishes as k goes to infinity and, therefore, the upper bound and the lower bound factor-revealing programs converge to the same value, as k goes to infinity.

Theorem 14. *Let* $\hat{z}_k^{A1}$ *be as in program (5) in [9] and let* $\hat{x}_k^{A1}$ *be as in program* (14)*. Then* $\sup_{k\geq 1} \hat{z}_k^{A1} = \inf_{k \geq 1} \hat{x}_k^{A1}$*.*

Proof. First notice that, if we double a dual solution of program (14), then the obtained solution for the corresponding minimization upper bound factor-revealing is still feasible. Therefore, we may assume that k is arbitrarily large. Consider an optimal solution of program (14). We have that $\alpha_j - d_j \leq \alpha_l + d_l$, for every j and l. Let j be such that $\varepsilon = \alpha_j - d_j$ is maximum and add up these inequalities for all l. We get $k\varepsilon = k(\alpha_j - d_j) = \sum_{l=1}^{k}(\alpha_j - d_j) \leq \sum_{l=1}^{k}(\alpha_l + d_l) \leq \hat{x}_k^{A1} + 1 \leq 1.816 + 1$. From Lemmas 12 and 13, we get that $\hat{x}_k^{A1} \geq \hat{z}_k^{A1} \geq \frac{1}{1+\varepsilon}\hat{x}_k^{A1} \geq \frac{1}{1+2.816/k}\hat{x}_k^{A1}$. Taking the limit as k goes to infinity, we get that $\sup_{z\geq 1} \hat{z}_k^{A1} = \inf_{k\geq 1} \hat{x}_k^{A1}$. □

It would be nice to bound the values of the variables of program (13), as this would suffice to show that the factor-revealing programs also converge for the squared metric case. Since the coefficients of the squared triangle inequality involved in program (13) are all greater than one, we cannot use the same approach as in Theorem 14. Although experiments suggest that the value of variable α_k in an optimal solution decreases as k increases, it does not seem trivial to determine whether α_k vanishes when k goes to infinity.

4 Analysis of the second algorithm

In this section we analyze the second algorithm of Jain *et al.* [9] for the squared metric case. The algorithm is essentially the same as Algorithm A1, but each connected city keeps contributing to unopened facilities. The contribution of a connected city j to an unopened facility i is the budget the city would save if facility i were opened. The algorithm, that is denoted by A2, is described in the following.

Algorithm A2 (C, F, c, f) [9]

1. Set $U := C$, meaning that every facility starts unopened, and every city unconnected. Each city j has some budget α_j, initially 0. At every moment, for each unopened facility i, if city j is unconnected, then j offers $\max(\alpha_j - c_{ij}, 0)$ to i, and, if city j is connected to facility i', then j offers $\max(c_{i'j} - c_{ij}, 0)$ to i.
2. While $U \neq \emptyset$, the budget of each unconnected city is increased continuously until one of the following events occur:
    (a) For some unconnected city j and some open facility i, $\alpha_j = c_{ij}$. In this case, connect city j to facility i and remove j from U.
    (b) For some unopened facility i, the total offer i receives from the cities equals the cost f_i of opening i. In this case, open facility i, connect to i each city j with a positive offer to i, and remove each connected city from U.

For the metric case, the approximation factor is 1.61. With a completely analogous reasoning, we obtain the corresponding factor-revealing program (15). The variables are the same as in program (2). The new variable r_{jl} corresponds to the budget α_j if city j is connected at the same time as city l, or corresponds to the distance from j to the facility to which j is connected just before l is connected.

$$\begin{array}{rlll} z_k^{A2} = & \text{maximize} & \frac{\sum_{j=1}^k \alpha_j}{f + \sum_{j=1}^k d_j} & \\ & \text{subject to} & \alpha_j \leq \alpha_{j+1} & \forall\, 1 \leq j < k \\ & & r_{jl} \geq r_{j,l+1} & \forall\, 1 \leq j < l < k \\ & & \sqrt{\alpha_l} \leq \sqrt{r_{jl}} + \sqrt{d_l} + \sqrt{d_j} & \forall\, 1 \leq j < l \leq k \\ & & \sum_{j=1}^{l-1} \max(r_{jl} - d_j, 0) + \sum_{j=l}^{k} \max(\alpha_l - d_j, 0) \leq f & \forall\, 1 \leq l \leq k \\ & & \alpha_j, d_j, f, r_{j,l} \geq 0 & \forall\, 1 \leq j \leq l \leq k. \end{array} \tag{15}$$

We repeat the previous analysis to give lower and upper bounds on the approximation factor of the second algorithm for the SMFLP.

Lemma 15. $2.415 \leq \sup_{k \geq 1} z_k^{A2} \leq 2.425$.

Proof. First, we obtain an upper bound factor-revealing program. See details in Appendix A. This program is exactly the same as program (15), except the fourth constraint is replaced with

$$\sum_{j=1}^{l-1} \max(r_{jl} - d_j, 0) + \sum_{\boldsymbol{j=l+1}}^{k} \max(\alpha_l - d_j, 0) \leq f.$$

Let x_k^{A2} be the optimal value of such a program. With CPLEX we get that $z_{500}^{A2} \approx 2.41565 > 2.415$, and that $x_{500}^{A2} \approx 2.42473 < 2.425$. □

Solving the upper bound factor-revealing LP obtained for the MFLP for $k = 500$ we may show that the approximation factor of A2 [9] is 1.602. The lower bound factor-revealing program and the maximization upper bound factor-revealing program are essentially the same, except for the extra terms of the kind $\max(\alpha_l - d_l)$. Therefore, Lemma 13 also holds for such programs. For the metric case, using a similar analysis to that of Theorem 14, one can show that the lower and the upper bound factor-revealing programs converge.

Theorem 16. *Let $\hat{z}_k^{A2}$ be as in program (25) in [9] and let $\hat{x}_k^{A2}$ be the optimal value of the corresponding upper bound factor-revealing program obtained by removing the terms of the kind* $\max(\alpha_l - d_l)$ *from the fourth restriction. Then* $\sup_{k \geq 1} z_k^{A2} = \inf_{k \geq 1} x_k^{A2}$.

5 Scaling and greedy augmentation

Algorithm A2 can also be analyzed as a bifactor approximation algorithm. The analysis uses a factor-revealing linear program, and is similar to the previous analysis. Mahdian *et al.* [13] observed that, due to the asymmetry between the approximation guarantee for the opened facilities cost and the connections cost, Algorithm A2 may be used to open facilities that are very economical. This gives rise to a two-phase algorithm, denoted here by A3, based on scaling the cost of facilities by a constant $\delta \geq 1$, and on the greedy augmentation technique introduced by Guha and Khuller [5]. The first phase opens the most economical facilities, and the second phase greedily includes facilities that reduce the cost of the solution.

Algorithm A3 (C, F, c, f) [**13**]

1. *Scaling*:
   - (a) Scale the facility costs by a factor δ.
   - (b) Run Algorithm A2 on the scaled instance.
2. *Greedy augmentation*: While there are facilities that reduce the total cost:
   - (a) Compute the gain g_i of opening each unopened facility i.
   - (b) Open a facility i that maximizes the ratio $\frac{g_i}{f_i}$.

Mahdian *et al.* [13] used a factor-revealing linear program to analyze Algorithm A3 using a somewhat different, but equivalent, greedy augmentation procedure. This was used to balance a bifactor from Algorithm A2 for the MFLP. As noticed by Byrka [3], this analysis is not restricted to Algorithm A2, and applies to any bifactor approximation for the FLP. Therefore, since it does not depend on the cost function being a metric, we can use it to balance a bifactor approximation for the squared metric case. This result is precisely stated as follows.

Lemma 17 ([13])**.** *Consider a (γ_f, γ_c)-approximation for the* FLP*. Then, for every $\delta \geq 1$, Algorithm A3 is a $(\gamma_f + \ln \delta + \varepsilon, 1 + \frac{\gamma_c}{\delta})$-approximation for the* FLP.

For the metric case, it has been shown that Algorithm A2 is a $(1.11, 1.78)$-approximation. This and Lemma 17 give a 1.52-approximation for the MFLP. For the SMFLP, we present an analysis based on an upper bound factor-revealing program. Using straightforward calculations, we may obtain the following:

Lemma 18. *Let $\gamma_f \geq 1$ be a fixed value and let $\gamma_c = x_k^{A2_c}$, where*

$$\begin{array}{rlll} x_k^{A2_c} = & \max & \frac{\sum_{j=1}^{k} \alpha_j - \gamma_f f}{\sum_{j=1}^{k} d_j} & \\ & \text{s.t.} & \alpha_l \leq \alpha_{l+1} & \forall\, 1 \leq l < k \\ & & r_{jl} \geq r_{j,l+1} & \forall\, 1 \leq j < l < k \\ & & \sqrt{\alpha_l} \leq \sqrt{r_{jl}} + \sqrt{d_l} + \sqrt{d_j} & \forall\, 1 \leq j < l \leq k \\ & & \sum\limits_{j=1}^{l-1} \max(r_{jl} - d_j, 0) + \sum\limits_{j=l+1}^{k} \max(\alpha_l - d_j, 0) \leq f & \forall\, 1 \leq l \leq k \\ & & \alpha_j, d_j, f, r_{jl} \geq 0 & \forall\, 1 \leq j \leq l \leq k. \end{array} \tag{16}$$

Then, if $\gamma_c < \infty$, Algorithm A2 is a (γ_f, γ_c)-approximation for the SMFLP.

The only difference between program (16) and the corresponding lower bound factor-revealing program is the extra term $\max(\alpha_l - d_l, 0)$ in the lower bound program, which is not in the fourth constraint of program (16). Again, having a bound for this term is sufficient to show convergence of the upper and lower bound factor-revealing programs. For the metric case, this can be done easily. Notice that we may assume $r_{jl} \leq \alpha_j$, so, using a similar analysis to that of Theorem 16, one can show that, if z_k and x_k are solutions for the lower and upper bound programs respectively, then $x_k - \gamma_f \varepsilon \leq z_k \leq x_k$, for some $\varepsilon = \mathrm{O}(\frac{1}{k})$.

We observe that program (16) is unbounded for values of γ_f close to one. This happens also for the corresponding lower bound factor-revealing program. This is in contrast to the factor-revealing programs obtained for the metric case, for which we know that Algorithm A2 is a $(1, 2)$-approximation. In this case, the lower bound program is always bounded, but the upper bound program is unbounded for $\gamma_f = 1$, or for values close to one. It would be interesting if it was possible to strengthen this upper bound factor-revealing program, so that it could also be used in the analysis also for $\gamma_f = 1$.

Theorem 19. *Algorithm A3 is a* 2.173*-approximation for the* SMFLP.

Proof. Consider program (16) for $\gamma_f = 1.45$. Numerical computations using CPLEX show that $x_{300}^{A2_c} \approx 3.40339 < 3.4034$. From Lemma 18, we get that Algorithm A2 is a $(1.45, 3.4034)$-approximation for the SMFLP. Now, for $\delta = 2.0543$, Lemma 17 states that Algorithm A3 is a $(2.169\ldots, 2.169\ldots)$-approximation for the SMFLP. □

6 Concluding remarks and future works

In this paper we considered the SMFLP, a generalization of the E^{2}FLP. The squared metric property captures the widely used l_2^2 distance function. We analyzed three known algorithms [9, 13] for the MFLP, and proved that they achieve 2.87, 2, 43 and 2.17 approximation factors for the SMFLP. Also, we showed that there is no 2.04-approximation, unless $\mathrm{P} = \mathrm{NP}$, by extending the hardness results for the MFLP.

We presented a technique for deriving upper bound factor-revealing programs that can be solved by computer, as an alternative way to obtain an upper bound on the approximation factors. This analysis allowed us to tighten the obtained approximation factors, and to simplify the analysis of the three algorithms, when used for to both SMFLP and MFLP instances. These upper bounds factor-reveling programs are analogous to the strongly factor-revealing linear programs presented recently by Mahdian and Yan [14], but were developed independently and are obtained using a different approach. The programs derived for the SMFLP have particular nonlinear constraints involving square roots. To solve these programs, we used a cutting plane strategy to replace such constraints with linear constraints.

It would be interesting to investigate whether the techniques used to obtain the upper bound factor-revealing programs for the FLP can be used in the analysis of other factor-revealing linear programs (or even nonlinear programs, if we may employ the cutting plane strategy). Also, the 1.5-approximation of Byrka [3], and the more recent 1.488-approximation of Li [12] for the MFLP are not directly based on factor-revealing programs. In future works, we want to analyze these algorithms when applied to instances with squared metrics.

References

[1] A. Archer. Inapproximability of the asymmetric facility location and k-median problems. Manuscript, 2000.

[2] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 106–113, 1998.

[3] J. Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 29–43, 2007.

[4] F.A. Chudak and D.B. Shmoys. Improved approximation algorithms for the capacitated facility location problem. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 875–876, 1999.

[5] S. Guha and S. Khuller. Greedy strikes back: improved facility location algorithms. In *Proceedings of the ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 649–657, 1998.

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

[7] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(2):148–162, 1982.

[8] K. Jain. A factor 2-approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[9] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM*, 50(6):795–824, 2003.

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

[11] W. Jain and V. V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, page 2, 1999.

[12] Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In *Automata, Languages and Programming*, pages 77–88, 2011.

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

[14] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing LPs. In *Proceedings of the 43rd annual ACM Symposium on Theory of computing*, pages 597–606, 2011.

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

[16] J. Vygen. Approximation algorithms for facility location problems (lecture notes). Technical Report 05950-OR, Research Institute for Discrete Mathematics, University of Bonn, 2005.

A Upper Bound Factor-Revealing Program for Algorithm A2

Consider tuples $(\beta_i, \gamma_i, \delta_i) \in {\mathbb{R}_+^*}^3$ and $B_i = 1 + \beta_i + \frac{1}{\gamma_i}$, $C_i = 1 + \gamma_i + \frac{1}{\delta_i}$, $D_i = 1 + \delta_i + \frac{1}{\beta_i}$ for $1 \le i \le m$. Using Lemma 1, we insert inequalities corresponding to these tuples, replacing the nonlinear constraint, and obtain $z_k^{A2} \le w_k^{A2}$, where w_k^{A2} is given by

$$\begin{array}{rlll}
w_k^{A2} = & \max & \sum_{j=1}^{k} \alpha_j & \\
& \text{s.t.} & f + \sum_{j=1}^{k} d_j \le 1 & \\
& & \alpha_j \le \alpha_{j+1} & \forall\, 1 \le j < k \\
& & r_{jl} \ge r_{j,l+1} & \forall\, 1 \le j < l < k \\
& & \alpha_l \le B_i r_{jl} + C_i d_l + D_i d_j & \forall\, 1 \le j < l \le k, 1 \le i \le m \\
& & r_{jl} - d_j \le x_{jl} & \forall\, 1 \le j < l \le k \\
& & \alpha_l - d_j \le x_{jl} & \forall\, 1 \le l \le j \le k \\
& & \sum_{j=1}^{k} x_{jl} \le f & \forall\, 1 \le l \le k \\
& & \alpha_j, d_j, f, r_{jl} \ge 0 & \forall\, 1 \le j \le l \le k \\
& & x_{jl} \ge 0 & \forall\, 1 \le j, l \le k.
\end{array} \tag{17}$$

Now, we calculate the dual of program (17) to derive the upper bound factor-revealing linear program. After that, we calculate its dual program (21), in order to use Lemma 1, and solve the upper bound factor-revealing program inserting cutting planes. We proceed the same way as done in Lemma 11. With similar arguments, we may see that $z_k^{A2} \leq z_{kt}^{A2}$, for any t, and we assume that k has the form $k = pt$, for some integer t. The dual of linear program (17) is given in the following.

$$
\begin{array}{rlll}
w_k^{A2} = & \min & \gamma & \\
& \text{s.t.} & a_l - a_{l-1} + \sum_{i=1}^{m} \sum_{j=1}^{l-1} c_{jli} + \sum_{j=l}^{k} e_{jl} \geq 1 & \forall\, 1 \leq l \leq k \\
& & \gamma - \sum_{i=1}^{m} C_i \sum_{j=1}^{l-1} c_{jli} - \sum_{i=1}^{m} D_i \sum_{j=l+1}^{k} c_{lji} - \sum_{j=1}^{k} e_{lj} \geq 0 & \forall\, 1 \leq l \leq k \\
& & \gamma - \sum_{l=1}^{k} h_l \geq 0 & \\
& & b_{j,l-1} - b_{jl} + e_{jl} - \sum_{i=1}^{m} B_i c_{jli} \geq 0 & \forall\, 1 \leq j < l \leq k \\
& & h_l - e_{jl} \geq 0 & \forall\, 1 \leq j, l \leq k \\
& & a_0 = a_k = b_{ll} = b_{lk} = 0 & \forall\, 1 \leq l \leq k \\
& & a_l, h_l, e_{jl} \geq 0 & \forall\, 1 \leq l, j \leq k \\
& & b_{jl}, c_{jli} \geq 0 & \forall \begin{array}{l} 1 \leq j < l \leq k \\ 1 \leq i \leq m. \end{array}
\end{array}
\tag{18}
$$

Now, we may derive the upper bound factor-revealing linear program. Let $\hat{n} = \lceil \frac{n}{p} \rceil$ and consider prime variables $\gamma', a'_l, b'_{jl}, c'_{jli}, e'_{jl}, h'_l$. We obtain a candidate solution for program (18) by defining:

$$
\begin{gathered}
\gamma = \gamma', \quad a_l = p\, a'_{\hat{l}} - (p\, \hat{l} - l)(a'_{\hat{l}} - a'_{\hat{l}-1}), \quad b_{jl} = b'_{\hat{j},\hat{l}} - \tfrac{p\,\hat{l}-l}{p}(b'_{\hat{j}\hat{l}} - b'_{\hat{j},\hat{l}-1}), \\
c_{jll} = \tfrac{c'_{\hat{j}\hat{l}l}}{p}, \qquad e_{jl} = \tfrac{e'_{\hat{j}\hat{l}}}{p} \qquad \text{and} \qquad h_l = \tfrac{h'_{\hat{l}}}{p}.
\end{gathered}
\tag{19}
$$

In the following, we apply definition (19) and calculate each coefficient expression for program (18). Again, notice that $a_l - a_{l-1} = a'_{\hat{l}} - a'_{\hat{l}-1}$, and that $b_{j,l-1} - b_{jl} = (b'_{\hat{j},\hat{l}-1} - b'_{\hat{j}\hat{l}})/p$. Also, fix variables c'_{lli} at zero.

$$
\begin{aligned}
\text{coeff}[\alpha_l] &= a_l - a_{l-1} + \sum_{i=1}^{m} \sum_{j=1}^{l-1} c_{jli} + \sum_{j=l}^{k} e_{jl} \\
&= a'_{\hat{l}} - a'_{\hat{l}-1} + \sum_{i=1}^{m} \sum_{j=1}^{l-1} \frac{c'_{\hat{j}\hat{l}i}}{p} + \sum_{j=l}^{pt} \frac{e'_{\hat{j}\hat{l}}}{p} \\
&\geq a'_{\hat{l}} - a'_{\hat{l}-1} + \sum_{i=1}^{m} \sum_{j'=1}^{\boldsymbol{\hat{l}-1}} p \frac{c'_{j'\hat{l}i}}{p} + \sum_{\boldsymbol{j'=\hat{l}+1}}^{t} p \frac{e'_{j'\hat{l}}}{p} \\
&= a'_{\hat{l}} - a'_{\hat{l}-1} + \sum_{i=1}^{m} \sum_{j'=1}^{\hat{l}-1} c'_{j'\hat{l}i} + \sum_{j'=\hat{l}+1}^{t} e'_{j'\hat{l}} \geq 1.
\end{aligned}
$$

$$
\begin{aligned}
\text{coeff}[d_l] &= \gamma - \sum_{i=1}^{m} \sum_{j=1}^{l-1} C_i c_{jli} - \sum_{i=1}^{m} \sum_{j=l+1}^{k} D_i c_{lji} - \sum_{j=1}^{k} e_{lj} \\
&= \gamma' - \sum_{i=1}^{m} C_i \sum_{j=1}^{l-1} \frac{c'_{\hat{j}\hat{l}i}}{p} - \sum_{i=1}^{m} D_i \sum_{j=l+1}^{k} \frac{c'_{\hat{l}\hat{j}i}}{p} - \sum_{j=1}^{k} \frac{e'_{\hat{l}\hat{j}}}{p} \\
&\geq \gamma' - \sum_{i=1}^{m} C_i \sum_{j'=1}^{\boldsymbol{\hat{l}}} p \frac{c'_{j'\hat{l}i}}{p} - \sum_{i=1}^{m} D_i \sum_{\boldsymbol{j'=\hat{l}}}^{t} p \frac{c'_{\hat{l}j',i}}{p} - \sum_{j'=1}^{t} p \frac{e'_{\hat{l}j'}}{p} \\
&= \gamma' - \sum_{i=1}^{m} C_i \sum_{j'=1}^{\boldsymbol{\hat{l}-1}} c'_{j'\hat{l}i} - \sum_{i=1}^{m} D_i \sum_{\boldsymbol{j'=\hat{l}+1}}^{t} c'_{\hat{l}j'i} - \sum_{j'=1}^{t} e'_{\hat{l}j'} \geq 0.
\end{aligned}
$$

$$
\text{coeff}[f] = \gamma - \sum_{l=1}^{k} h_l = \gamma' - \sum_{l=1}^{k} \frac{h'_{\hat{l}}}{p} = \gamma' - \sum_{l'=1}^{t} p \cdot \frac{h'_{l'}}{p} = \gamma' - \sum_{l'=1}^{t} h'_{l'} \geq 0.
$$

$$
\text{coeff}[r_{j,l}] = b_{j,l-1} - b_{jl} + e_{jl} - \sum_{i=1}^{r} B_i \, c_{jli} = \frac{b'_{\hat{j},\hat{l}-1} - b'_{\hat{j}\hat{l}}}{p} + \frac{e'_{\hat{j}\hat{l}}}{p} - \sum_{i=1}^{r} B_i \frac{c'_{\hat{j}\hat{l}i}}{p} \geq 0.
$$

$$
\text{coeff}[x_{jl}] = h_l - e_{j,l} = \frac{h'_{\hat{l}}}{p} - \frac{e'_{\hat{j}\hat{l}}}{p} \geq 0.
$$

Conjoining all constraints, the obtained upper bound factor-revealing linear program is:

$$
\begin{array}{rlll}
x_t^{A2} = & \min & \gamma & \\
& \text{s.t.} & a_l - a_{l-1} + \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{l-1} c_{jli} + \sum\limits_{\boldsymbol{j=l+1}}^{t} e_{jl} \geq 1 & \forall\, 1 \leq l \leq t \\
& & \gamma - \sum\limits_{i=1}^{m} C_i \sum\limits_{j=1}^{l-1} c_{jli} - \sum\limits_{i=1}^{m} D_i \sum\limits_{j=l+1}^{t} c_{lji} - \sum\limits_{j=1}^{t} e_{lj} \geq 0 & \forall\, 1 \leq l \leq t \\
& & \gamma - \sum\limits_{l=1}^{t} h_l \geq 0 & \\
& & b_{j,l-1} - b_{jl} + e_{jl} - \sum\limits_{i=1}^{m} B_i c_{jli} \geq 0 & \forall\, 1 \leq j < l \leq t \\
& & h_l - e_{jl} \geq 0 & \forall\, 1 \leq j, l \leq t \\
& & a_0 = a_t = b_{ll} = b_{lt} = 0 & \forall\, 1 \leq l \leq t \\
& & a_l, h_l, e_{jl} \geq 0 & \forall\, 1 \leq l, j \leq t \\
& & b_{jl}, c_{jli} \geq 0 & \forall \begin{array}{l} 1 \leq j < l \leq k \\ 1 \leq i \leq m. \end{array}
\end{array}
\tag{20}
$$

Finally, calculating the dual of program (20), we obtain program (21).

$$
\begin{array}{rlll}
x_t^{A2} = & \max & \sum_{j=1}^{t} \alpha_j & \\
& \text{s.t.} & f + \sum_{j=1}^{t} d_j \leq 1 & \\
& & \alpha_j \leq \alpha_{j+1} & \forall\, 1 \leq j < t \\
& & r_{jl} \geq r_{j,l+1} & \forall\, 1 \leq j < l < t \\
& & \alpha_l \leq B_i r_{jl} + C_i d_l + D_i d_j & \forall\, 1 \leq j < l \leq t, 1 \leq i \leq m \\
& & r_{jl} - d_j \leq x_{jl} & \forall\, 1 \leq j < l \leq t \\
& & \alpha_l - d_j \leq x_{jl} & \forall\, 1 \leq l < j \leq t \\
& & \sum_{j=1}^{t} x_{jl} \leq f & \forall\, 1 \leq l \leq t \\
& & \alpha_j, d_j, f, r_{jl} \geq 0 & \forall\, 1 \leq j \leq l \leq t \\
& & x_{jl} \geq 0 & \forall\, 1 \leq j, l \leq t.
\end{array}
\tag{21}
$$

B Experimental results

In Table 1, we present computational results using CPLEX for the lower bound (column z_k^{A1}) and upper bound (column x_k^{A1}) for the approximation factor of Algorithm A1. In Table 2, we present lower and upper bounds on the approximation factor of Algorithm A2 (columns z_k^{A2} and x_k^{A2}, respectively). In Table 3, we present computational results for program (15) when $\gamma_f = 1.45$, and the approximation factor obtained from Lemma 17. The chosen δ is given by the solution of equation $\gamma_f + \ln \delta = 1 + \frac{\gamma_c}{\delta}$, that is, $\delta = e^{W_0((\gamma_c - 1)e^{\gamma_f - 1}) - (\gamma_f - 1)}$. Figure 1 shows the trade-off between connection and facility costs approximation guarantees for the Algorithm A2, and Figure 2 shows the trend of obtained factor for Algorithm A3 as we vary the value of γ_f, when $k = 50$.

Table 1: Solutions of factor-revealing programs for A1.

k	z_k^{A1}	x_k^{A1}
10	2.57261	3.18162
20	2.71704	3.01717
50	2.80540	2.92579
100	2.83534	2.89553
200	2.85034	2.88046
300	2.85532	2.87543
400	2.85782	2.87292
500	2.85930	2.87142
600	2.86029	2.87041
700	2.86099	2.86970

Table 2: Solutions of factor-revealing programs for A2.

k	z_k^{A2}	x_k^{A2}
10	2.20702	2.65131
20	2.30987	2.53301
50	2.37551	2.46544
100	2.39773	2.44278
200	2.40894	2.43150
300	2.41267	2.42775
400	2.41453	2.42586
500	2.41565	2.42473

Table 3: Solutions of connection factor-revealing programs for A2, and obtained factor for A3.

k	x_k^{A2c}	best δ	factor
10	4.02931	2.33433	2.29772
20	3.64790	2.16561	2.22270
50	3.48465	2.09159	2.18792
100	3.43524	2.06895	2.17704
200	3.41127	2.05793	2.17170
300	3.40339	2.05430	2.16993

Figure 1: Trade-off between connection and facility approximation factors.

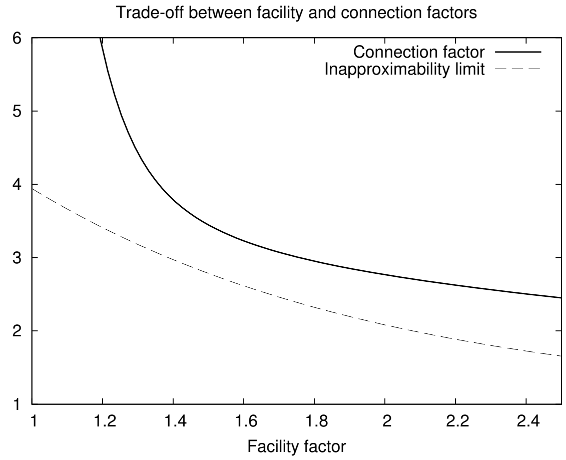

Figure 2: Trend of the obtained balanced approximation factors.