Discrete Applied Mathematics 289 (2021) 219–229

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Approximability of the dispersed $\vec{p}$-neighbor k-supplier problem

Martijn van Ee

Faculty of Military Sciences, Netherlands Defence Academy, Den Helder, The Netherlands

ARTICLE INFO

Article history:
Received 6 February 2020
Received in revised form 21 September 2020
Accepted 11 October 2020
Available online 22 October 2020

Keywords:
k-supplier
k-center
Approximation algorithms
Inapproximability
Dispersion

ABSTRACT

We consider the dispersed $\vec{p}$-neighbor k-supplier problem. In the classical k-supplier problem, we have to select k suppliers in a metric space such that the maximum distance between a customer and its closest supplier is minimized. Here, we generalize this problem to the case where each customer possibly needs service from more than one supplier. Moreover, the selected suppliers should not be too close to each other, i.e., they need to be dispersed. For the classical k-supplier problem, and its special case the k-center problem, there is a 3- and a 2-approximation respectively. We show that these guarantees can also be given in the case when customers need service from multiple suppliers, without imposing dispersion constraints. If we generalize the problem to the dispersed case, without imposing neighboring constraints, we get inapproximability results depending on the measure of dispersion. We also show (almost) matching upper bounds. Finally, we show that adding both the neighbor requirement and the dispersion requirement leads to an inapproximable problem.

1. Introduction

In this paper, we discuss the approximability of a generalization of the k-supplier problem [10]. In this problem, we are given a metric space containing a set of customers and suppliers. We can choose k suppliers, and each customer will be served by its closest supplier. The goal is to find a set of suppliers that minimizes the maximum distance between any customer and its closest supplier. A special case of the k-supplier problem in which there is no distinction between suppliers and customers, i.e., every point in the metric space is a customer, and at each point a supplier can be selected, is the well-known k-center problem [9]. Here, the chosen suppliers are usually referred to as centers.

We generalize the k-supplier problem in two ways. We consider the generalization in which customer v needs service from multiple suppliers, say p_v. These neighbor requirements may differ over the customers. The objective now becomes the maximum distance between any customer v and its p_v-closest supplier. We also generalize the problem by adding a dispersion requirement, i.e., the distance between any two suppliers should be large enough. These two generalizations result in the dispersed $\vec{p}$-neighbor k-supplier and k-center problem, which will be formally defined in the next section. In this paper, we study to what extend the approximability results for the k-supplier (and k-center) problem can be generalized when neighbor or dispersion requirements are added.

The problem originates from the following military application. Suppose we are going to deploy several units of special forces. There are some predefined locations at which a unit can be placed. There is also a set of potential targets, from which we will engage exactly one. To successfully engage a target, a predefined number of units is needed at the same

E-mail address: M.v.Ee.01@mindef.nl.

https://doi.org/10.1016/j.dam.2020.10.007

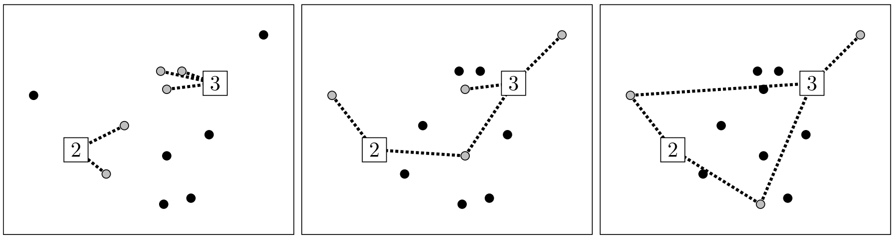

Fig. 1. Illustration of an instance of the dispersed $\vec{p}$-neighbor k-supplier problem, and feasible solutions satisfying increasing dispersion requirements, when we can choose at most five suppliers. There is a dotted line between any customer v and its p_v closest suppliers. The objective value is equal to the maximum distance taken over all dotted lines.

Table 1
Overview of results on upper bounds (UB) and lower bounds (LB) on the approximability.

		Non-dispersed		Dispersed	
		k-center	k-supplier	k-center	k-supplier
1-neighbor	UB	2 [5,6,9]	3 [10]	max{2, Δ} (Theorem 4)	max{5, $\Delta+1$} (Theorem 7)
	LB	2 [11]	3 [10]	Δ (Theorem 5)	$\Delta+1$ (Theorem 8)
p-neighbor	UB	2 [2,16]	3 [2,16]	Inapproximable (Theorem 11)	
$\vec{p}$-neighbor	UB	2 (Theorem 1)	3 (Theorem 3)		

time. The required number of units may differ over the targets. We want to place the units such that the time (assuming the units travel at unit speed) to successfully engage any target is minimized. However, if units are placed on locations close to each other, these units may become a target for the enemy. Therefore, the units should also be dispersed, i.e., the distance between any pair of units should be large enough. We can relate this application with the dispersed $\vec{p}$-neighbor k-supplier problem, by interpreting the locations at which the units can be placed as suppliers, and the targets as customers.

An illustration of an instance and feasible solutions is given in Fig. 1, where dots represent suppliers, and squares represent customers. The number in a square represents the number of suppliers that is needed to serve a customer. In Fig. 1, we illustrate feasible solutions (in gray) that satisfy increasing dispersion requirements. Note that the time needed to serve a customer is determined by the supplier that is farthest (out of the chosen suppliers that will serve this customer) from the customer. Here, it is clear that increasing the dispersion requirement increases the objective value.

The novelty of the problems considered in this paper lies in two elements. Firstly, the dispersion requirement has not been considered. Secondly, we consider the case that the number of required suppliers differs per customer. In this paper, we investigate the consequences of these elements on the approximability of the k-supplier and the k-center problem. Here, an α-approximation algorithm will, for each instance, produce a feasible solution that has an objective value that is within a factor α of the optimal value for the instance. It also has to run in polynomial time. Before giving formal definitions (Section 2) and the main results, we give a brief overview of our results and relevant literature.

1.1. Our results

In Table 1, we show our results on approximation algorithms and inapproximability for problem variations considered in this paper. In Section 3, we show a 2-approximation for the $\vec{p}$-neighbor k-center problem, and a 3-approximation for the $\vec{p}$-neighbor k-supplier problem. This is done by generalizing the results in [2] for the p-neighbor k-center and k-supplier problem, in which the neighbor requirement is equal for all points and customers respectively. Here, we note that in the $\vec{p}$-neighbor k-center problem the objective is the maximum distance, taken over all points without a center, from a point to a center. If the objective is the maximum distance, taken over all points, from a point to a center, we speak of the $\vec{p}$-reliable k-center problem, for which we also give a 2-approximation (Theorem 2).

Then, in Section 4, we investigate the approximability of the dispersed k-center and k-supplier problem. We show lower bounds depending on Δ, the dispersion requirement, and also give approximation algorithms with (almost) matching upper bounds. Finally, in Section 5, we show that for the dispersed p-neighbor k-center problem, it is NP-complete to decide whether there exists a feasible solution. As a corollary, the problem is inapproximable. This also settles the approximability of the harder problem variations.

We also consider the dispersed k-median problem (Section 4.3). In this problem, the objective is the average distance of a point to its closest center. For the dispersed k-median problem, we show lower bounds on the approximability of 1.36, 2, and $\frac{7}{8}+\frac{3}{16}\Delta$ (Theorems 9 and 10). For simplicity, the results for the $\vec{p}$-reliable k-center problem and the dispersed k-median problem are omitted in Table 1.

1.2. Related work

The k-center problem is a classical problem in combinatorial optimization. For clarity, we emphasize here that we consider the discrete version, and that we assume that the distances satisfy the triangle inequality. The problem is NP-hard, even on Euclidean instances [18]. For general metrics, there are several 2-approximations, for example the ones obtained by Feder and Greene [5], Gonzalez [6], and Hochbaum and Shmoys [9]. It was shown in [11] that there is no α-approximation algorithm for $\alpha < 2$, unless P = NP.

For the k-supplier problem, Hochbaum and Shmoys [10] gave a 3-approximation, and showed that this result is also tight. Recently, Nagarajan et al. [19] improved the approximation guarantee on Euclidean instances to $1 + \sqrt{3}$. This was the first algorithm with a guarantee below 3. For the k-center problem in the Euclidean case the best approximation guarantee is still equal to 2.

The k-median problem is NP-hard [13], even for Euclidean instances [18]. The approximability of the k-median has not been settled yet. Currently, the best guarantee is given by Byrka et al. [1], who designed a $(2.675 + \epsilon)$-approximation, for any given $\epsilon > 0$. It was shown by Jain and Vazirani [12] that the k-median problem cannot be approximated within a factor $1 + 2/e$.

The p-neighbor k-center problem was considered first by Krumke [17]. He gave a 4-approximation by generalizing the ideas of Hochbaum and Shmoys [10]. Later, Chaudhuri et al. [2] improved this to a 2-approximation. They also gave a 2-approximation for the p-reliable k-center problem, and a 3-approximation for the p-neighbor k-supplier problem. Independently, Khuller et al. [16] gave a 2-approximation for the p-neighbor k-center problem, a 3-approximation for the p-reliable k-center problem, and a 3-approximation for the k-supplier problem.

A related problem we like to mention is the (discrete) k-dispersion problem. Here, one has to place k units in a metric space such that the dispersion, i.e., the distance between any two units, is maximized. The problem was shown to be NP-hard in both the general and the Euclidean case by Erkut [4]. Both Tamir [21] and Ravi et al. [20] gave an algorithm that produces a solution with a dispersion of at least half the optimal dispersion. Ravi et al. [20] also showed that this result is tight. Grigoriev et al. [7] consider the problem of maximizing the number of placed units given a certain dispersion requirement, where units can also be placed along an edge. They consider the computational complexity for small dispersion requirements.

2. Formal definitions

In this section, we will formally define our problem variations. We will also mention graph-theoretical notions that will be needed in this article.

The input of the dispersed $\vec{p}$-neighbor k-supplier problem consists of a metric space containing a set of suppliers $\mathcal{S}$ and set of customers $\mathcal{C}$. We denote by $d(u, v)$ the distance between point u and v. We assume that the minimum non-zero distance is equal to 1, and that the distances satisfy the triangle inequality. Customer $v \in \mathcal{C}$ has a neighbor requirement of p_v, i.e., we need p_v suppliers to serve customer v. The input also contains an integer k, representing the maximum number of suppliers that can be chosen, and a rational number $\Delta \geq 1$, the dispersion requirement. A feasible solution is a subset of suppliers of size at most k and at least $\max_v p_v$ such that the distance between any two chosen suppliers is at least Δ. The objective is the maximum distance, over all v, between a customer v and its p_v-closest chosen supplier. Our goal is to minimize this objective.

Note that instances with a minimum non-zero distance unequal to 1 can be scaled such that the minimum non-zero distance becomes 1, without influencing the approximation guarantee. In this case, Δ can be interpreted as the normalized dispersion requirement. Also note that it is easy to check whether the optimal value, denoted by $\textsc{Opt}$, equals 0. Hence, we assume in the remainder of the paper that $\textsc{Opt} \geq 1$.

When the neighbor requirement is equal for all customers, say p, we refer to this problem as the dispersed p-neighbor k-supplier problem. In case all neighbor requirements equal 1, we refer to the problem as the dispersed k-supplier problem. If we drop the dispersion constraint, we refer to the problem as the $\vec{p}$-neighbor k-supplier problem.

In the dispersed $\vec{p}$-neighbor k-center problem there is no distinction between suppliers and customers. The points in the metric space are now referred to as vertices, and V is the set of all vertices. Any vertex in the metric space can be chosen as a center. A feasible solution is a set of at most k and at least $\max_v p_v$ centers such that the distance between any two centers is at least Δ. The objective is the maximum distance, over all v, between any vertex v, that is not a center, and its p_v-closest center. Our goal is to minimize this objective. Special cases with respect to neighbor or dispersion requirement are addressed similarly as for the dispersed $\vec{p}$-neighbor k-supplier problem. We also discuss the approximability of the $\vec{p}$-reliable k-center problem. In this problem, also the centers are considered in the objective value, i.e., the objective is the maximum distance, over all v, between any vertex v and its p_v-closest center. For vertices at which a center is placed, the closest center of this vertex is the center placed at itself.

We also briefly discuss the dispersed k-median problem in Section 4.3. Here, we have the same input as in the dispersed k-center problem. The objective is now the average distance between a vertex and its nearest center, and our goal is to minimize this value.

Finally, we like to mention some basic graph-theoretical notions that will be used in this paper. For a given graph $G = (V, E)$, we say that a set $S \subseteq V$ is independent if $(u, v) \notin E$ for all $u, v \in S$. A maximum independent set is an independent with the highest possible number of vertices. A maximal independent set is an independent set that cannot be extended by adding vertices, i.e., for all $u \notin S$ there is some $v \in S$ such that $(u, v) \in E$. We say that a set $S \subseteq V$ is a vertex cover if at least one of u and v is in S whenever $(u, v) \in E$. Note that when S is an independent set in G, then $V \setminus S$ is a vertex cover in G. We say that a set $S \subseteq V$ is a dominating set if for all v, either $v \in S$ or there is some $u \in S$ such that $(u, v) \in E$. Note that, by definition, a maximal independent set is also a dominating set.

3. The $\vec{p}$-neighbor k-center and k-supplier problem

Here, we give a 2-approximation for the $\vec{p}$-neighbor and the $\vec{p}$-reliable k-center problem and a 3-approximation for the $\vec{p}$-neighbor k-supplier problem by adjusting the algorithms presented in [2]. Recall that the objective of the $\vec{p}$-neighbor k-center problem is the maximum distance, over all v, between any vertex v, that is not a center, and its p_v-closest center. For the $\vec{p}$-reliable k-center problem the objective is the maximum distance, over all v, between any vertex v and its p_v-closest center.

3.1. The $\vec{p}$-neighbor k-center problem

We first give some essential definitions. For the algorithm, it is convenient to consider our metric space as a weighted graph $G = (V, E)$, where edge $e \in E$ has weight w_e. Rename the edges such that $w_1 \leq w_2 \leq \cdots \leq w_{|E|}$. Now, define G_i as the graph containing only edges with weight at most w_i. Further, for a given graph G, define the square of the graph G^2 as the graph that contains an edge (u, v) whenever there is a path between u and v in G of at most two edges.

We define $\vec{p}$-dominating and $\vec{p}$-independent sets as follows. We say that a set $S \subseteq V$ is $\vec{p}$-dominating if every vertex not in S satisfies the neighbor requirement, i.e., for all $v \notin S$, $\deg_S(v) \geq p_v$. We say that a set $S \subseteq V$ is $\vec{p}$-independent if every vertex in S has a number of neighbors in S that is less than its neighbor requirement, i.e., for all $v \in S$, $\deg_S(v) \leq p_v - 1$.

We first prove how the size of $\vec{p}$-dominating sets in G relates to the size of $\vec{p}$-independent sets in G and G^2. These will be used to obtain a 2-approximation. The proofs are generalizations of proofs in [17] and [2]. Hence, the same structure is used for convenience.

Lemma 1. *If G has a $\vec{p}$-dominating set D of size $|D|$, then for any $\vec{p}$-independent set I in G^2 we have $|I| \leq |D|$.*

Proof. Let D be a $\vec{p}$-dominating set in G, and I a $\vec{p}$-independent set in G^2. If $I \subseteq D$, then we clearly have $|I| \leq |D|$. Thus, we assume this is not the case, and we let v be a vertex in $I \setminus D$.

Let T_1 be the vertices in D that are neighbors of v, and T_2 the vertices in $V \setminus D$ that are neighbors of the vertices in T_1. Note that v is contained in T_2. Further, let $T = T_1 \cup T_2$. Since each vertex in T is either a neighbor of v in G^2 or v itself, the set I contains at most p_v vertices from T. On the other hand, D contains at least p_v vertices from T.

Consider the graph G', which is the graph induced by $V \setminus T$. The set $D \setminus T$ is a $\vec{p}$-dominating set in G'. To see this, consider an arbitrary vertex u in $V \setminus T$ that is not contained in $D \setminus T$. Since D is a $\vec{p}$-dominating set in G, u has at least p_u neighbors in D. None of these neighbors can be contained in T, because otherwise u would be contained in T_2, and hence u would not be contained in $V \setminus T$.

At the same time, $I \setminus T$ is clearly a $\vec{p}$-independent set in $(G')^2$. Thus, we can repeat the above construction with $\vec{p}$-dominating set $D \setminus T$ and $\vec{p}$-independent set $I \setminus T$, until we obtain that there is no vertex in the residual graph that belongs to the $\vec{p}$-independent set but not to the $\vec{p}$-dominating set. Since at each step the number of vertices deleted from I was at most the number of deleted vertices from D, we have that $|I| \leq |D|$. □

Lemma 2. *Given a graph $G = (V, E)$ and an integral vector $\vec{p}$, with $1 \leq p_v \leq |V| - 1$ for all v, there exists a $\vec{p}$-independent set $S \subseteq V$ that is also $\vec{p}$-dominating.*

Proof. Let S be a $\vec{p}$-independent set that is not $\vec{p}$-dominating. In particular, let $z \in V \setminus S$ be such that $\deg_S(z) = q < p_z$. Let U be the neighbors of z in S that are maximal in S with respect to $\vec{p}$-independence, i.e., for all $u \in U$, we have $\deg_S(u) = p_u - 1$. If $U \neq \emptyset$, we let $G[U]$ be the subgraph induced by U in G. Let I be a maximal independent set in $G[U]$. Since a maximal independent set is also a dominating set, we know that the set $S \setminus I \cup \{z\}$ is $\vec{p}$-independent. If $U = \emptyset$, we know that the set $S \cup \{z\}$ is $\vec{p}$-independent.

Define the potential of a $\vec{p}$-independent set S as $\psi(S) = \sum_{v \in S} p_v - |E(G[S])|$, where $E(G[S])$ denotes the edge set of the subgraph induced by S in G. Since

$$\sum_{v \in S} p_v - \sum_{v \in S \setminus I \cup \{z\}} p_v = \sum_{v \in I} p_v - p_z$$

$$|E(G[S])| - |E(G[S \setminus I \cup \{z\}])| = \sum_{v \in I} (p_v - 1) - (q - |I|)$$

we have

$$\psi(S) - \psi(S \setminus I \cup \{z\}) = q - p_z < 0$$

Given any $\vec{p}$-independent set that is not $\vec{p}$-dominating we can obtain another $\vec{p}$-independent set that has a strictly larger potential. Therefore the $\vec{p}$-independent set with maximum potential is also $\vec{p}$-dominating. $\square$

The proof of the lemma above yields a polynomial time algorithm for computing a $\vec{p}$-independent set that is also $\vec{p}$-dominating. We start with an arbitrary $\vec{p}$-independent set and if it is not $\vec{p}$-dominating we select a vertex z that has less than p_z neighbors in the set. Then we delete a maximal independent set of the neighbors of z which are maximal with respect to $\vec{p}$-independence, and add z to the set. Since the potential is at least 0 and at most $\sum_v p_v$, we will obtain a $\vec{p}$-independent set that is also $\vec{p}$-dominating in at most $\sum_v p_v$ steps. Since $\sum_v p_v \leq |V|^2$, this algorithm runs in polynomial time.

Let G_i be the first subgraph in the sequence $G_1, G_2, G_3, \ldots$ for which the $\vec{p}$-independent set found in G_i^2 by using the algorithm above is of cardinality at most k. It follows from the triangle inequality that the longest edge in G_i^2 is of length at most $2w_i$. Hence, we have a solution for the $\vec{p}$-neighbor k-center problem with value at most $2w_i$. Since in G_{i-1}^2 we found a $\vec{p}$-independent set of cardinality more than k, it follows from Lemma 1 that G_{i-1} does not have a $\vec{p}$-dominating set of size at most k. Hence, the optimal value is at least w_i, which results in the next theorem.

Theorem 1. *There is a 2-approximation for the $\vec{p}$-neighbor k-center problem.*

3.2. The $\vec{p}$-reliable k-center problem

For the $\vec{p}$-reliable k-center problem, we slightly adjust the definition of a $\vec{p}$-dominating set in such a way that also vertices in the set need to be dominated. Note that a vertex can also dominate itself in this problem. We say that a set $S \subseteq V$ is $\vec{p}$-dominating if every vertex satisfies the neighbor requirement, i.e., for all $v \notin S$, $\deg_S(v) \geq p_v$, and for all $v \in S$, $\deg_S(v) \geq p_v - 1$. We need the following lemmas to state our algorithm, and to analyze its performance.

Lemma 3. *If G has a $\vec{p}$-dominating set of size $|D|$, then any independent set I in G^2 has $\sum_{v \in I} p_v \leq |D|$.*

Proof. Let I be an independent set in G^2. Each vertex in G can dominate at most one vertex of I. Hence, any $\vec{p}$-dominating set in G has size at least $\sum_{v \in I} p_v$. Since G has a $\vec{p}$-dominating set of size $|D|$, we have $\sum_{v \in I} p_v \leq |D|$. $\square$

Lemma 4. *There exists a $\vec{p}$-dominating set S such that any vertex $v \in S$ with $\deg_S(v) > p_v - 1$ has a neighbor $u \in S$ with $\deg_S(u) = p_u - 1$.*

Proof. For the sake of contradiction, let S be a $\vec{p}$-dominating set, and z a vertex in S, not satisfying the lemma. Then, $\deg_S(z) > p_z - 1$ and $\deg_S(u) \geq p_u$ for every $u \in S$ that is a neighbor of z. Let U be the neighbors of z not in S that are tight with respect to their neighbor requirement, i.e., for all $u \in U$, we have $\deg_S(u) = p_u$. If $U \neq \emptyset$, let I be a maximal independent set in $G[U]$. Since I is a dominating set in $G[U]$, the set $S \cup I \setminus \{z\}$ is $\vec{p}$-dominating. If $U = \emptyset$, then $S \setminus \{z\}$ is $\vec{p}$-dominating.

Define the potential of a $\vec{p}$-dominating set S as $\psi(S) = \sum_{v \in S}(p_v - 1) - |E(G[S])|$. Since

$$\sum_{v \in S \cup I \setminus \{z\}} (p_v - 1) - \sum_{v \in S}(p_v - 1) = \sum_{v \in I}(p_v - 1) - (p_z - 1)$$
$$|E(G[S \cup I \setminus \{z\}])| - |E(G[S])| = \sum_{v \in I}(p_v - 1) - q$$

we have

$$\psi(S \cup I \setminus \{z\}) - \psi(S) = q - (p_z - 1) > 0$$

Given any $\vec{p}$-dominating set that does not satisfy the lemma, we can obtain a $\vec{p}$-dominating set with strictly larger potential. Hence, the $\vec{p}$-dominating set with maximum potential satisfies the lemma. $\square$

Lemma 5. *Any $\vec{p}$-dominating set S satisfying the statement in Lemma 4 contains an independent set with $\sum_{v \in I} p_v \geq |S|$.*

Proof. Arbitrarily select a vertex z from S with a degree in S of at most $p_z - 1$, and include it in the independent set I. Delete the vertex and its neighbors, and proceed in this manner, until no such vertex exists anymore. Since any vertex v with degree at least p_v has a neighbor u with degree $p_u - 1$, we stop only when we have deleted all vertices. When adding a vertex z to the independent set, we delete at most p_z vertices from S. Hence, the independent set I satisfies $\sum_{v \in I} p_v \geq |S|$. $\square$

Our algorithm starts with a $\vec{p}$-dominating set that may not satisfy the statement in Lemma 4. By applying the operations mentioned in the proof, we obtain a $\vec{p}$-dominating set that does satisfy the statement. Since in each step the potential strictly increases, and since the potential is at least $-|E|$ and at most $\sum_{v \in V}(p_v - 1)$, the maximum number of steps needed is at most $\sum_{v\in V}(p_v-1)+|E| \leq |V|^2+|E|$. Hence, this procedure only takes polynomial time. If the resulting $\vec{p}$-dominating set contains more than k vertices, we apply the procedure from Lemma 5 to obtain an independent set I with $\sum_{v\in I} p_v > k$.

Now, let G_i be the first subgraph in the sequence $G_1, G_2, \ldots$ for which the $\vec{p}$-dominating set in G_i^2, found by the procedure above, is of size at most k. Then, we have a feasible solution for the $\vec{p}$-reliable k-center problem with value at most $2w_i$. Since we found an independent set I with $\sum_{v\in I} p_v > k$ in G_{i-1}^2, it follows from Lemma 3 that there is no $\vec{p}$-dominating set in G_{i-1} of size at most k. Hence, the optimal value is at least w_i, and we obtained a 2-approximation.

Theorem 2. *There is a 2-approximation for the $\vec{p}$-reliable k-center problem.*

3.3. The $\vec{p}$-neighbor k-supplier problem

For the $\vec{p}$-neighbor k-supplier problem, it is again convenient to consider our metric space as a weighted graph $G=(V,E)$, where V contains both customers and suppliers. Further, we use roughly the same definitions as for the $\vec{p}$-neighbor and $\vec{p}$-reliable k-center problem. However, a $\vec{p}$-dominating set now is a set of suppliers such that each customer v has at least p_v chosen suppliers as a neighbor. Furthermore, an independent set is an independent set of customers. We further define the cube of the graph G^3 as the graph that contains an edge (u,v) whenever there is a path between u and v in G of at most three edges. The next lemma is similar to Lemma 3.

Lemma 6. *If G has a $\vec{p}$-dominating set of size $|D|$ then any independent set I in G^2 has $\sum_{v\in I} p_v \leq |D|$.*

Proof. Let I be an independent set in G^2. Each supplier in G can dominate at most one vertex of I. Hence, any $\vec{p}$-dominating set in G has size at least $\sum_{v\in I} p_v$. Since G has a $\vec{p}$-dominating set of size $|D|$, we have $\sum_{v\in I} p_v \leq |D|$. □

Firstly note that if there is a customer v with less than p_v neighbors in G_i, the optimal value should be larger than w_i. Hence, we start our algorithm with the smallest ℓ such that each customer v has at least p_v neighbors in G_ℓ. Then for each $j \geq \ell$, we select a maximal independent set I in G_j^2, where in each step we select the vertex with maximum neighbor requirement (among the vertices that do not violate independence). In this way, we satisfy $\sum_{v\in I} p_v \geq p_{max}$, which ensures that our solution is feasible. Let G_i be the first subgraph in the sequence $G_1, G_2, \ldots$ such that this maximal independent set I has the property that $\sum_{v\in I} p_v \leq k$. Since G_{i-1}^2 has an independent set I' such that $\sum_{v\in I'} p_v > k$, it follows from Lemma 6 that G_{i-1} does not have a $\vec{p}$-dominating set of size at most k. Hence, the optimal value is at least w_i.

For each customer v in the maximal independent set I we choose p_v suppliers adjacent to it in G_i. This can be done since each customer v has at least p_v neighbors in G_i. Since $\sum_{v\in I} p_v \leq k$, the total number of chosen suppliers is at most k. Moreover, by the construction of our independent set I, every customer $v \notin I$ has a neighboring customer $u \in I$ in G_i^2 with neighbor requirement $p_u \geq p_v$. Hence, the chosen suppliers are a $\vec{p}$-dominating set in G_i^3. The longest edge in G_i^3 has length at most $3w_i$. Hence, we obtained a 3-approximation for our problem.

Theorem 3. *There is a 3-approximation for the $\vec{p}$-neighbor k-supplier problem.*

4. The dispersed k-center and k-supplier problem

In this section, we look at the impact of the dispersion constraint on the approximability of the k-center, k-supplier, and k-median problem. For all problems, we show lower bounds depending on Δ. For the dispersed k-center and k-supplier problem, we also provide algorithms with (almost) matching upper bounds.

4.1. The dispersed k-center problem

For the dispersed k-center problem we adjust the algorithm for the k-center problem designed by Gonzalez [6]. The algorithm, which we call the Greedy algorithm, starts by selecting an arbitrary vertex as our first center. Then, we add a center at a vertex at maximum distance of the current centers, as long as this does not violate the dispersion constraint. The obtained solution is feasible, and is proven to be within a factor of $\max\{2, \Delta\}$ of the optimal value. The analysis is similar to the one in [22].

Algorithm 1 Greedy algorithm.

Place a center at an arbitrary point z, i.e., $S=\{z\}$
while $|S| < k$ and $\max_{v\in V\setminus S} \min_{u\in S} d(u,v) \geq \Delta$ **do**
  Determine $z \in V\setminus S$ that maximizes $\min_{u\in S} d(u,z)$
  $S := S \cup \{z\}$
end while
return S

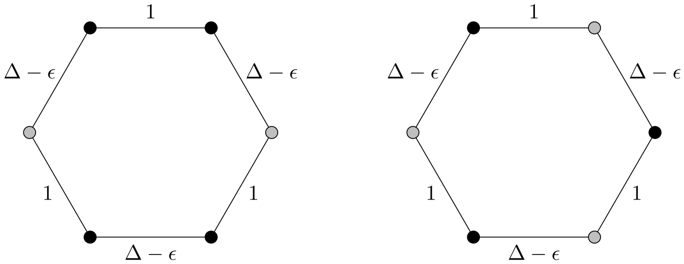

Fig. 2. Tight instance for Greedy algorithm. Left: the solution returned by the Greedy algorithm (in gray), right: the optimal solution (in gray).

Theorem 4. *For $\Delta \leq 2$, the Greedy algorithm is a 2-approximation for the dispersed k-center problem. For $\Delta > 2$, the Greedy algorithm is a Δ-approximation for the dispersed k-center problem.*

Proof. Firstly, consider the optimal solution. Define $\ell \leq k$ to be the number of centers used in the optimal solution, and denote by Opt the objective value of the optimal solution. By assigning each point to its nearest center (ties are broken arbitrarily), the optimal solution partitions the points into ℓ clusters. Within a cluster, each pair of points is within distance 2Opt. This follows from combining the fact that each point is within distance Opt of a center and the triangle inequality.

Suppose that the Greedy algorithm uses at least ℓ centers. We will show that the value of the algorithm is within a factor 2 of the optimal value. Two cases need to be considered. In the first case, a center is placed in each of the clusters defined by the optimal solution. Now, as observed above, each point is within distance 2Opt of its nearest center. In the other case, there is a cluster with at least two centers in it. Consider the center chosen latest in time. This center is within distance 2Opt of the other centers in the cluster. Moreover, when choosing this point as a center, we considered all remaining points and chose this one because it maximizes the distance to the current centers. This means that all other points are within distance 2Opt of a center. Hence, in both cases we are within a factor 2 of the optimal value.

Now, suppose that the Greedy algorithm uses $\ell' < \ell$ centers. Again, two cases need to be considered. In case there is a cluster with at least two centers in it, we can use the same reasoning as above to conclude that each point is within distance 2Opt of a center. In the other case, each cluster defined by the optimal solution contains at most one center. Since $\ell' < k$, the Greedy algorithm did not add more centers because adding a center would have violated the dispersion constraint. Hence, the distance between any point and its nearest center is smaller than Δ. Since Opt ≥ 1, the value of the solution produced by the Greedy algorithm is within a factor Δ of the optimal value. Combining all the above results in the theorem. □

Observe that if the Greedy algorithm is able to place all k centers, the value of the solution obtained is within a factor two of the optimal value. Furthermore, for $\Delta > 2$, the approximation guarantee is Δ and not 2, because the Greedy algorithm may fail to place the same number of centers as the optimal solution. We use this observation to show that the analysis is tight, and there is no polynomial time algorithm with a guarantee smaller than Δ.

To see that the analysis is tight, consider the cycle in Fig. 2. The cycle contains six vertices and six edges. The edge weights are 1 or $\Delta - \epsilon$ alternatingly, where $0 < \epsilon \leq 1$. Consider the instance obtained by taking the metric completion of this cycle, i.e., the other distances correspond to shortest paths in this weighted cycle graph. The Greedy algorithm may choose the left-most vertex first. Then, it will choose the right-most vertex next, since this one maximizes the distance to the left-most vertex. Unfortunately, this solution cannot be extended without violating the dispersion constraint. This solution has a value of $\Delta - \epsilon$. The optimal solution chooses the left-most vertex, and the two vertices adjacent to the right-most vertex. This is also a feasible solution, but with a value of 1. Hence, the gap is $\Delta - \epsilon$, and the analysis is tight.

Next, we show that there is no polynomial time algorithm for dispersed k-center with an approximation guarantee smaller than Δ. This is shown by a reduction from 3-Satisfiability (3-SAT) [14], and is based on a standard reduction from 3-SAT to the independent set problem. In the satisfiability problem, we are given a Boolean formula in conjunctive normal form. Equivalently, we are given variables $x_1, \ldots, x_n$, and clauses $C_1, \ldots, C_m$. Each clause is a disjunction of literals, where a literal can be a variable or its negation. The question is whether there exists a truth-assignment to the variables such that each clause is satisfied. In 3-SAT, each clause contains exactly three literals.

Theorem 5. *There is no α-approximation algorithm for the dispersed k-center problem for any $\alpha < \Delta$, unless P = NP.*

Proof. Given an instance of 3-SAT, we create a cycle on three vertices for each clause. The vertices correspond to the literals, and the distances are set equal to 1. Moreover, we connect each vertex corresponding to a literal to each vertex corresponding to the negation of this literal by an edge of length $\Delta - \epsilon$, where $0 < \epsilon \leq 1$. We call this graph H (see Fig. 3), and assume without loss of generality that it is connected. The instance of dispersed k-center is defined by taking the metric completion of this weighted graph H. Finally, we set k equal to m, the number of clauses.

Suppose we have a yes-instance of 3-SAT. In each cycle, we can choose one vertex corresponding to a literal with a true-value in the assignment. This solution is feasible, because the centers form an independent set in the created graph

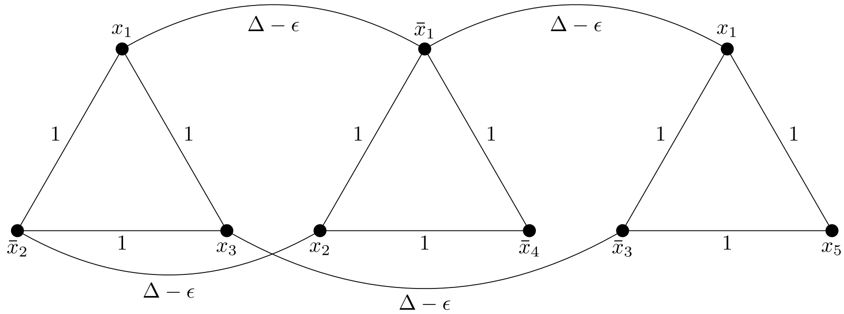

Fig. 3. Illustration of created instance for dispersed k-center when the 3-SAT-instance consists of the clauses $x_1 \vee \bar{x}_2 \vee x_3$, $\bar{x}_1 \vee x_2 \vee \bar{x}_4$, and $x_1 \vee \bar{x}_3 \vee x_5$.

H, and hence the distance between each pair of centers is at least Δ. Moreover, this solution has a value of 1, since we chose one center in each cycle.

Now, suppose we have a no-instance of 3-SAT. In this case, any solution choosing one vertex in each of the m cycles will violate the dispersion constraint. This follows because any such solution satisfying the dispersion constraint would correspond to a satisfying assignment. Hence, any dispersed solution has less than m centers. This implies that there are vertices at distance at least $\Delta - \epsilon$ from their nearest center. So, if there is no satisfying assignment, the value of the dispersed k-center instance is at least $\Delta - \epsilon$.

We showed that it is NP-complete to decide whether an instance of the dispersed k-center problem has a value of at most 1 or a value of at least $\Delta - \epsilon$. Hence, there is no α-approximation for any $\alpha < \Delta$, unless P = NP. $\square$

4.2. Dispersed k-supplier problem

For the dispersed k-supplier problem, we give two algorithms. The first algorithm is a $(\Delta + 3)$-approximation, but is within a factor 3 from the optimal value if it succeeds in selecting exactly k suppliers. The second algorithm improves upon the first one for $\Delta > 2$. Namely, it is a $\max\{5, \Delta + 1\}$-approximation. If this algorithm succeeds in selecting exactly k suppliers, it is within a factor 5 from the optimal value. Later, we show a lower bound on the approximability of $\Delta + 1$, which shows that the second algorithm is tight for $\Delta \geq 4$.

Our first algorithm runs the Greedy algorithm for the k-center problem (Algorithm 1) on the set of customers, i.e., choose a customer that maximizes the distance to the already chosen customers. Call this set $C = \{c_1, \ldots, c_k\}$. Then, for each i, we choose the supplier closest to c_i, and call this supplier s_i. In this way, we obtain the set $S = \{s_i | c_i \in C\}$. Since the suppliers in S may not be dispersed, we choose a maximal dispersed subset R of S as our solution. Here, a set is maximal dispersed if adding any vertex not in the set would violate the dispersion requirement. Next, we show that this algorithm is a $(\Delta + 3)$-approximation for the dispersed k-supplier problem.

Theorem 6. *The algorithm above is a $(\Delta + 3)$-approximation for the dispersed k-supplier problem.*

Proof. Firstly, note that $\textsc{Opt} \geq d(c_i, s_i)$ for all i. Consider an arbitrary customer $v \in \mathcal{C}$. We first show that there exists a customer $c_i \in C$ at distance at most $2\textsc{Opt}$ from v. If $v \in C$, then this clearly holds, so we assume $v \notin C$. For the sake of contradiction, suppose there is no customer $c_i \in C$ at distance at most $2\textsc{Opt}$ from v. Then, by construction of the algorithm, for any two customers $c_i, c_j \in C$ we have $d(c_i, c_j) > 2\textsc{Opt}$. But then the set $C \cup \{v\}$ consists of $k + 1$ customers such that any pair is at distance strictly more then $2\textsc{Opt}$. This is impossible since in that case at least $k + 1$ suppliers are needed for a solution with value $\textsc{Opt}$.

Hence, there exists a customer $c_i \in C$ at distance at most $2\textsc{Opt}$ from v. If s_i is chosen in the solution, we know by the triangle inequality that

$$d(v, s_i) \leq d(v, c_i) + d(c_i, s_i) \leq 2\textsc{Opt} + \textsc{Opt} = 3\textsc{Opt}.$$

If s_i is not chosen in the solution, by the maximality of R, there is some supplier $\widetilde{s_i}$ that is chosen and is within distance Δ of s_i. Again, by the triangle inequality we obtain

$$d(v, \widetilde{s_i}) \leq d(v, c_i) + d(c_i, s_i) + d(s_i, \widetilde{s_i}) \leq 2\textsc{Opt} + \textsc{Opt} + \Delta \leq (\Delta + 3)\textsc{Opt}. \quad \square$$

Observe that if the algorithm is able to choose all suppliers in S, it is a 3-approximation. For $\Delta > 2$, we can improve upon the algorithm above. For this, we first select for each customer a closest supplier, i.e., we select $S' = \{s_i | c_i \in \mathcal{C}\}$. Then, we perform Algorithm 1 on S', resulting in a set of suppliers called R'. We show that if the algorithm succeeds in selecting k suppliers in S', the algorithm is a 5-approximation. If the algorithm gets stuck, we prove an approximation guarantee of $\Delta + 1$. The proof is similar to the proof of Theorem 6.

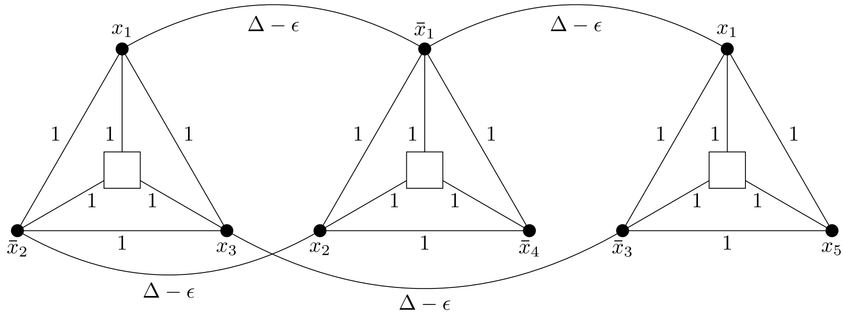

Fig. 4. Illustration of created instance for dispersed k-supplier when the 3-SAT-instance consists of the clauses $x_1 \vee \bar{x}_2 \vee x_3$, $\bar{x}_1 \vee x_2 \vee \bar{x}_4$, and $x_1 \vee \bar{x}_3 \vee x_5$. Here, white squares represent customers and black dots represent suppliers.

Theorem 7. *For $\Delta > 4$, the algorithm above is a $(\Delta + 1)$-approximation for the dispersed k-supplier problem. For $\Delta \leq 4$, it is a 5-approximation.*

Proof. Firstly, note that $\text{OPT} \geq d(c_i, s_i)$ for all i. Consider an arbitrary customer $v \in \mathcal{C}$, and name its closest supplier w. We first show that, if the algorithm is able to choose k suppliers, there exists a supplier $s'_i \in R'$ at distance at most 4OPT from w. For the sake of contradiction, suppose there is no supplier $s'_i \in R'$ at distance at most 4OPT from w. Then, by construction of the algorithm, for any two chosen suppliers s_i, s_j we have $d(s_i, s_j) > 4\text{OPT}$. Then, for two customers c_i, c_j corresponding to any two s_i, s_j respectively, we have $d(c_i, c_j) > 2\text{OPT}$. Similarly, because $d(w, s_i) > 4\text{OPT}$ for all $s_i \in R'$, we have that $d(v, c_i) > 2\text{OPT}$ for all i. But then there is a set consisting of $k+1$ customers such that any pair is at distance strictly more then 2OPT. This is impossible since in that case at least $k+1$ suppliers are needed for a solution with value OPT.

Note that if w is chosen, the distance between v and its closest chosen supplier is within OPT. If w is not chosen, we consider two cases. If the algorithm is able to choose k suppliers, there exists a supplier $s'_i \in R'$ at distance at most 4OPT from w. We know by the triangle inequality that

$$d(v, s'_i) \leq d(v, w) + d(w, s'_i) \leq \text{OPT} + 4\text{OPT} = 5\text{OPT}.$$

If the algorithm is not able to choose k suppliers, there is some supplier $\widetilde{w}$ that is chosen and is within distance Δ of w. Again, by the triangle inequality we obtain

$$d(v, \widetilde{w}) \leq d(v, w) + d(w, \widetilde{w}) \leq \text{OPT} + \Delta \leq (\Delta + 1)\text{OPT}. \quad \square$$

It is easy to obtain an instance with gap $\Delta + 1$ from the instance in Fig. 2. This is done by associating a supplier with each vertex in the cycle graph, and by adding a customer for each supplier at distance 1 from the supplier. By adjusting the proof of Theorem 5, we now obtain a $(\Delta+1)$-lower bound on the approximability of the dispersed k-supplier problem.

Theorem 8. *There is no α-approximation algorithm for the dispersed k-supplier problem for any $\alpha < \Delta + 1$, unless $P = NP$.*

Proof. Consider the same reduction as in Theorem 5. We define the vertices in graph H to be suppliers. Now, we add a client in each cycle at distance 1 of the suppliers in the cycle (see Fig. 4). As in the proof of Theorem 5, if the instance of 3-SAT is satisfiable, then the objective value equals 1. Otherwise, there is at least one client at distance $\Delta + 1 - \epsilon$ of its closest chosen supplier. $\square$

4.3. Dispersed k-median

We show two inapproximability results concerning the dispersed k-median problem. The first is a constant lower bound obtained by relating the problem to the vertex cover problem. The second result is a lower bound depending on Δ, which shows that a linear dependence of Δ is also unavoidable for the dispersed k-median problem. It also improves upon the first result for $\Delta > 6$. In the proofs, we use the total distance as our objective, i.e., the average distance multiplied by the number of vertices, instead of the average distance.

An instance of the vertex cover problem is a graph $G = (V, E)$, and an integer K. The question is whether there exists a vertex cover in G of size at most K. This problem is known to be NP-complete [14].

Theorem 9. *There is no α-approximation algorithm for dispersed k-median, for any $\alpha < 1.36$, unless $P = NP$. Assuming the Unique Games Conjecture holds, there is no α-approximation algorithm, for any $\alpha < 2$, unless $P = NP$.*

Proof. Given is an instance of the vertex cover problem, i.e., a graph $G = (V, E)$, and an integer K. We create an instance of the dispersed k-median problem as follows. If there is an edge between vertices u and v in G, then the distance between u and v is set to 1. Otherwise, the distance is 2. Further, we set $k = |V|$, and we let the dispersion requirement equal 2, i.e., $\Delta = 2$.

Because of the dispersion requirement, any feasible solution for this instance corresponds to an independent set in G. Also, an optimal solution always corresponds to a maximal independent set. This is true since $k = |V|$, and the objective value cannot increase when centers are added. If a set of centers corresponding to a maximal independent set contains r vertices, then the value of the solution in the dispersed k-median instance is equal to $|V| - r$. Namely, r vertices contribute 0, and $|V| - r$ vertices contribute 1, since a maximal independent set is a dominating set. Remember that the complement of an independent set is a vertex cover. Hence, the value of the solution is equal to the number of vertices in the corresponding vertex cover. So, the dispersed k-median problem on the created instance is equivalent to the vertex cover problem on G, i.e., there is a vertex cover of size K if and only if the instance of the dispersed k-median problem has a solution with value K. The result of the theorem now follows from the inapproximability results obtained for the vertex cover problem [3,15]. □

For the next theorem, we use the same reduction from 3-SAT as in Theorem 5. However, here we use that, for any $\delta > 0$, no polynomial time algorithm can distinguish between 3-SAT-instances in which all clauses are satisfiable, and instances in which at most a $(\frac{7}{8} + \delta)$-fraction of the clauses are satisfiable, unless P = NP [8].

Theorem 10. *For any $\delta > 0$, there is no approximation algorithm for dispersed k-median with approximation ratio $\frac{7}{8} + \frac{3}{16}\Delta - \delta$, unless P = NP.*

Proof. Consider the same reduction as in Theorem 5. If all clauses are satisfiable, we can choose exactly one vertex in each cycle in our solution. The other two vertices in the cycle each contribute 1 to the objective value. Hence, the solution has an objective value of $2m$. If a $(\frac{7}{8} + \delta')$-fraction of the clauses are satisfiable, then we can choose exactly one vertex in our solution in $(\frac{7}{8} + \delta')m$ cycles, and no vertex in the other $(\frac{1}{8} - \delta')m$ cycles. Cycles without a vertex in the solution contain three vertices at a distance of at least $\Delta - \epsilon$ to their closest center. Hence, the solution has an objective value of at least $2(\frac{7}{8} + \delta')m + 3(\Delta - \epsilon)(\frac{1}{8} - \delta')m$. The ratio of these values is equal to

$$\frac{2(\frac{7}{8} + \delta')m + 3(\Delta - \epsilon)(\frac{1}{8} - \delta')m}{2m} = \left(\frac{7}{8} + \delta'\right) + \left(\frac{3}{16} - \frac{3}{2}\delta'\right)(\Delta - \epsilon) = \frac{7}{8} + \frac{3}{16}\Delta - \delta,$$

where $\delta = \delta'(3(\Delta - \epsilon)/2 - 1)$. The hardness result for 3-SAT now gives the desired result. □

5. Dispersed *p*-neighbor *k*-center problem

In this section, we show that combining the neighbor and dispersion requirement leads to an inapproximable problem. That is, we show that the problem of deciding whether there exists a feasible solution for the dispersed p-neighbor (or p-reliable) k-center problem is NP-complete. Since any α-approximation algorithm should return a feasible solution, the problem is inapproximable. We give a reduction from the independent set problem. Here, we are given a graph $G = (V, E)$ and an integer K. The question is whether there is an independent set in G of size at least K. This is a well-known NP-complete problem [14].

Theorem 11. *Deciding whether an instance of the dispersed p-neighbor k-center problem has a feasible solution is NP-complete.*

Proof. Given an instance of the independent set problem, we create an instance of the dispersed p-neighbor k-center problem as follows. Set $d(u, v) = 1$ if $(u, v) \in E$, and $d(u, v) = 2$ otherwise. Further, let $p = K$, $\Delta = 2$, and $k = |V|$. We show that the created instance has a feasible solution if and only if the instance of the independent set problem has an independent set of size at least K.

If the independent set instance has an independent set of size at least K, we choose the vertices in the independent set as the solution for the dispersed p-neighbor k-center instance. This is a feasible solution, since every vertex has at least $p = K$ neighbors, the vertices satisfy the dispersion constraint, and at most $k = |V|$ centers are chosen. Similarly, if the created instance has a feasible solution, we see that G has an independent set of size at least K. □

6. Conclusion

In this paper, we extended the results of [2] for the p-neighbor k-center, the p-reliable k-center and the p-neighbor k-supplier problem to the version in which the neighbor requirement differs over the vertices or customers. For the first two, this gives a 2-approximation, for the last one a 3-approximation. Then, we showed that adding a constraint on the dispersion of the centers or suppliers gives lower bounds on approximability that depend on Δ, the dispersion requirement. This holds for the dispersed k-center, k-supplier, and k-median problem. For the dispersed k-center problem, we give a tight $\max\{2, \Delta\}$-approximation. For the k-supplier problem, the approximation guarantee of our algorithm is max{5, $\Delta + 1$}. Finally, we showed that adding both the neighbor and dispersion requirement leads to an inapproximable problem. That is, we showed that it is NP-complete to decide whether there exists a feasible solution for the dispersed p-neighbor k-center problem.

The main goal of this paper was to assess the influence of adding neighbor and dispersion requirements on the approximability of the classical k-center and k-supplier problem. Our results show that adding the dispersion requirement has enormous effects on the guarantees we can give. On the other hand, we showed that the same approximation guarantees can be achieved when just the neighbor requirement is added.

References

[1] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, Khoa Trinh, An improved approximation for k-median and positive correlation in budgeted optimization, ACM Trans. Algorithms (TALG) 13 (2) (2017) 23:1–23:31.
[2] Shiva Chaudhuri, Naveen Garg, R. Ravi, The p-neighbor k-center problem, Inform. Process. Lett. 65 (3) (1998) 131–134.
[3] Irit Dinur, Samuel Safra, On the hardness of approximating minimum vertex cover, Ann. Math. (2005) 439–485.
[4] Erhan Erkut, The discrete p-dispersion problem, European J. Oper. Res. 46 (1) (1990) 48–60.
[5] Tomás Feder, Daniel Greene, Optimal algorithms for approximate clustering, in: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, ACM, 1988, pp. 434–444.
[6] Teofilo F. Gonzalez, Clustering to minimize the maximum inter-city distance, Theoret. Comput. Sci. 38 (1985) 293–306.
[7] Alexander Grigoriev, Tim A. Hartmann, Stefan Lendl, Gerhard J. Woeginger, Dispersing obnoxious facilities on a graph, in: Proceedings of the 36th International Symposium on Theoretical Aspects of Computer Science, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019, pp. 33:1–33:11.
[8] Johan Håstad, Some optimal inapproximability results, J. ACM 48 (4) (2001) 798–859.
[9] Dorit S. Hochbaum, David B. Shmoys, A best possible heuristic for the k-center problem, Math. Oper. Res. 10 (2) (1985) 180–184.
[10] Dorit S. Hochbaum, David B. Shmoys, A unified approach to approximation algorithms for bottleneck problems, J. ACM 33 (3) (1986) 533–550.
[11] Wen-Lian Hsu, George L. Nemhauser, Easy and hard bottleneck location problems, Discrete Appl. Math. 1 (3) (1979) 209–215.
[12] Kamal Jain, Vijay V. Vazirani, Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation, J. ACM 48 (2) (2001) 274–296.
[13] Oded Kariv, Seifollah L. Hakimi, An algorithmic approach to network location problems. II: the p-medians, SIAM J. Appl. Math. 37 (3) (1979) 539–560.
[14] Richard M. Karp, Reducibility among combinatorial problems, Complexity Comput. Comput. (1972) 85–103.
[15] Subhash Khot, Oded Regev, Vertex cover might be hard to approximate to within 2- ε, J. Comput. System Sci. 74 (3) (2008) 335–349.
[16] Samir Khuller, Robert Pless, Yoram J. Sussmann, Fault tolerant k-center problems, Theoret. Comput. Sci. 242 (1–2) (2000) 237–245.
[17] Sven Oliver Krumke, On a generalization of the p-center problem, Inform. Process. Lett. 56 (2) (1995) 67–71.
[18] Nimrod Megiddo, Kenneth J. Supowit, On the complexity of some common geometric location problems, SIAM J. Comput. 13 (1) (1984) 182–196.
[19] Viswanath Nagarajan, Baruch Schieber, Hadas Shachnai, The Euclidean k-supplier problem, Math. Oper. Res. 45 (1) (2020) 1–14.
[20] Sekharipuram S. Ravi, Daniel J. Rosenkrantz, Giri Kumar Tayi, Heuristic and special case algorithms for dispersion problems, Oper. Res. 42 (2) (1994) 299–310.
[21] Arie Tamir, Obnoxious facility location on graphs, SIAM J. Discrete Math. 4 (4) (1991) 550–567.
[22] David P. Williamson, David B. Shmoys, The Design of Approximation Algorithms, Cambridge University Press, 2011.