Diversity Maximization via Composable Coresets

Sepideh Aghamolaei* Majid Farhadi* Hamid Zarrabi-Zadeh*

Abstract

Given a set S of points in a metric space, and a diversity measure $\mathrm{div}(\cdot)$ defined over subsets of S, the goal of the *diversity maximization* problem is to find a subset $T \subseteq S$ of size k that maximizes $\mathrm{div}(T)$. Motivated by applications in massive data processing, we consider the composable coreset framework in which a coreset for a diversity measure is called α-composable, if for any collection of sets and their corresponding coresets, the maximum diversity of the union of the coresets α-approximates the maximum diversity of the union of the sets. We present composable coresets with near-optimal approximation factors for several notions of diversity, including remote-clique, remote-cycle, and remote-tree. We also prove a general lower bound on the approximation factor of composable coresets for a large class of diversity maximization problems.

1 Introduction

The *diversity maximization* problem—finding a subset of k points to maximize some function of the inter-point distances—is a fundamental problem in location theory [20,21] and has received considerable attention over the past few years, due to its application to search result diversification [5, 6, 14]. Various notions of diversity have been studied in the literature, most of which are proved to be NP-hard in both metric and geometric settings, and hence, the focus has been on providing efficient approximation algorithms. Among the most well-studied diversity problems are *remote-edge*, whose objective is to maximize the minimum distance in the k-subset [7, 11, 22], and the *remote-clique* problem, whose aim is to maximize the average distance [8, 12, 13, 18]. There are also some results on maximizing other combinatorial structures such as minimum spanning trees and minimum-weight tours [10, 16].

Motivated by applications in massive data processing, we consider the coreset framework, which is a fundamental tool for designing approximation algorithms, especially for large data sets [4]. In this framework, a small subset of input data set, called a "coreset", is extracted in such a way that solving the optimization problem on the coreset yields a solution to the whole data set with a guaranteed approximation factor. Many coresets considered in the literature are "decomposable" in the sense that taking the union of two coresets computed for two given sets yields a coreset for the union of those two sets with the same approximation guarantee. This property is essentially useful for designing streaming algorithms [9, 17], as it allows to maintain a coreset for the points recently inserted, and merge it to the coreset maintained for the rest of the points.

In [24], Zarrabi-Zadeh introduced a special class of decomposable coresets, called "core-preserving", having an additional property that taking a coreset of a coreset yields a coreset with the same size and approximation factor. Such coresets are in particular useful for obtaining streaming algorithms whose working space is independent of the size of input. The idea was used to obtain efficient streaming algorithms for problems such as k-center [24] and maintaining ε-kernels of fat point sets [25]. A similar idea was coined as "mergeable coresets" by Agarwal *et al.* [3], and was used to obtain better algorithms for maintaining statistical data summaries in the data stream model.

Very recently, Indyk *et al.* [19] introduced the notion of "composable coresets" in which the union of a collection of coresets gives a coreset for the points in the union of the sets within a guaranteed approximation factor. All decomposable coresets (and hence, core-preserving and mergeable coresets) are composable by definition. However, in composable coresets, the approximation factor may be increased after taking union, though it is still guaranteed to be within a certain factor. Composable coresets are in particular useful for distributed settings and MapReduce computation, in which a massive point set is partitioned among a set of machines/mappers, and each machine maps its input data into a composable coreset. A single reducer then takes the union of all the coresets received from the mappers, and computes a solution to the union, which is guaranteed to be within a good approximation factor.

Our contributions. In this paper, we revisit the composable coresets framework of Indyk *et al.* [19], and further refine it to the notion of "disjoint composable coresets", in which input data sets are assumed to be disjoint. We present improved composable coresets for several diversity maximization problems in both disjoint and non-disjoint settings. The problems studied in this

*Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. Email: {aghamolaei, m_farhadi}@ce.sharif.edu, zarrabi@sharif.edu.

Problem	Diversity Measure	Approximation Factor		
		Previous [19]	New	
			Disjoint	General
Remote-edge	$\min_{p,q\in S} d(p,q)$	3	$3^\dagger$	$3^\dagger$
Remote-clique	$\sum_{p,q\in S} d(p,q)$	51	$6+\varepsilon$	$7+4\sqrt{2}+\varepsilon$
Remote-star	$\min_{p\in S} \sum_{q\in S\setminus\{p\}} d(p,q)$	102	12	26
Remote-bipartition	$\min_{Q\subset S,\lvert Q\rvert=k/2} \sum_{p\in Q,q\in S\setminus Q} d(p,q)$	255	18	38
Remote-tree	$w(\mathrm{MST}(S))$	6	4	4
Remote-cycle	$w(\mathrm{TSP}(S))$	12	$3^\dagger$	$3^\dagger$
Remote t-trees	$\min_{S=S_1\vert\cdots\vert S_t} \sum_{i=1}^{t} w(\mathrm{MST}(S_i))$	6	4	4
Remote t-cycles	$\min_{S=S_1\vert\cdots\vert S_t} \sum_{i=1}^{t} w(\mathrm{TSP}(S_i))$	12	5	5

Table 1: Summary of the new results. In this table, S denotes the input set, d is the distance function in the underlying metric space, $\varepsilon > 0$ is an arbitrarily small constant, and $S = S_1|\cdots|S_t$ denotes a partition of S into t subsets. Tight factors are marked with † sign.

paper are listed and formally defined in Table 1. Here is a brief summary of our results.

- For the remote-clique problem, a factor-51 composable coreset was presented in [19]. When input sets are disjoint, we show that a much better approximation factor of $6+\varepsilon$ (for any $\varepsilon > 0$) is achievable for the problem. In general non-disjoint case, we provide an approximation factor of $7+4\sqrt{2}+\varepsilon \approx 12.66+\varepsilon$, greatly improving over the best previous factor of 51.

- For the remote-edge problem, a factor-3 composable coreset was presented in [1, 19]. Indyk *et al.* [19] left this question open whether a better approximation factor is possible. We settle this question in negative by showing that 3 is the best factor possible for the remote-edge problem. Our proof is indeed very general, and implies a lower bound of 3 for all notions of diversity listed in Table 1.

- We show that for any point set, the weight of its clique approximates the weight of its minimum partition to within a factor of $3-\frac{4}{k}$, improving upon the previous bound of 5 proved in [19]. Combined with our new factor for the remote-clique problem, this yields improved factors of 18 and 38 for the remote partition problem in the disjoint and non-disjoint settings, respectively, substantially improving over the previous bound of 255 available for the problem.

- We prove a tight upper bound of $2-\frac{2}{k}$ on the ratio of the weight of the minimum star of a point set and the weight of its clique. This yields improved factors of 12 and 26 for the remote-star problem in the disjoint and non-disjoint settings, respectively, greatly improving over the previous bound of 102 available for the problem.

- For the remote-cycle problem, we present a factor-3 composable coreset, improving the best previous bound of 12 available for the problem. Our coreset is indeed optimal, considering the general lower bound of 3 that we have presented in this paper.

- For the remote-tree and remote t-trees problems, we provide an approximation factor of 4, improving over the best previous factor of 6 obtained in [19]. We also improve the approximation factor of the remote t-cycles problem from 12 to 5.

As with many other approximation algorithms, our algorithms for extracting the coresets are simple, and are based on two known off-line algorithms, namely the Gonzalez's algorithm and the local search. However, the analyses of the approximation factors are non-trivial, and are based on finding a careful mapping from the points in the optimal solution to the points in the coreset, while keeping the error incurred as small as possible.

2 Preliminaries

Let (X,d) be a metric space, and f be a measure defined over subsets of X. A function $c(\cdot)$ that maps a set $S \subseteq X$ into one of its subsets is called an *α-composable coreset* for f, if for any collection of sets $S_1,\ldots,S_\ell$, with $S = \cup_{i=1}^{\ell} S_i$ and $T = \cup_{i=1}^{\ell} c(S_i)$,

$$\max\left\{\frac{f(S)}{f(T)}, \frac{f(T)}{f(S)}\right\} \leqslant \alpha.$$

The value $\alpha \geqslant 1$ is called the *approximation factor* of the coreset. A *disjoint α-composable coreset* is analogously defined, with an additional property that the input sets S_i are assumed to be disjoint.

Given a point set S in a metric space (X,d), we denote by $G[S]$ a complete graph over vertex set S, with edge weights specified by the metric distance d. Let Π denote a specific graph structure (e.g., a clique or a spanning

Algorithm 1 GMM(S, k)

$T \leftarrow$ {an arbitrary point $p \in S$}
for $i = 2, \ldots, k$ **do**
    find a point $p \in S \setminus T$ maximizing $d(p, T)$
    $T \leftarrow T \cup \{p\}$
return T

Algorithm 2 LocalSearch(S, k)

$T \leftarrow$ a k-subset of S containing the two farthest pts
while $\exists\, p \in T,\ q \in S \setminus T$ s.t.
    $\operatorname{div}(T \setminus \{p\} \cup \{q\}) > (1 + \frac{\varepsilon}{k}) \operatorname{div}(T)$ **do**
    $T \leftarrow T \setminus \{p\} \cup \{q\}$
return T

tree). Following the terminology of [10], we define the remote-Π problem as follows. For a point set $S \subseteq X$, the *diversity* of S (with respect to Π), denoted by $\operatorname{div}(S)$, is the weight of a Π structure in $G[S]$ whose total edge weight is minimum. The *k-diversity* of S, denoted by $\operatorname{div}_k(S)$, is the maximum diversity over all k-subsets of S, i.e., $\operatorname{div}_k(S) = \max_{P \subseteq S, |P|=k} \operatorname{div}(P)$. The *remote-*Π problem is then to compute, for a given point set S and a parameter k, the k-diversity of S with respect to Π. For example, the remote-tree problem involves finding a k-subset of S whose minimum spanning tree has maximum weight. Note that $\operatorname{div}_k(S)$ is undefined when $|S| < k$.

For a weighted graph G, we denote by $w(G)$ the total weight of the edges in G. Given a set S, we denote by $S = S_1 | \cdots | S_t$ the partition of S into t disjoint subsets $S_1, \ldots, S_t$.

2.1 Algorithms

The two offline algorithms that we will use for computing the coresets are the Gonzalez's algorithm and the local search. The Gonzalez's algorithm [15], presented in Algorithm 1, starts from an arbitrary point, and iteratively adds a point whose distance to the points already chosen is maximized. If r denotes the minimum pairwise distance in the set $T = \text{GMM}(S, k)$, then the following two properties, known as *anti-cover* properties, hold:

- $\forall p \in T : d(p, T \setminus \{p\}) \geqslant r$
- $\forall p \in S : d(p, T) \leqslant r$

The local search algorithm [2], presented in Algorithm 2, starts with an arbitrary subset of size k containing the two farthest points, and then, at each iteration tries to locally improve its current solution by exchanging a single point. The total number of iterations of this algorithm is at most $\log_{1+\frac{\varepsilon}{k}}(k^2) = O(\frac{k}{\varepsilon} \log k)$.

3 Composable Coresets for Diversity Problems

Consider a collection of sets $S_1, \ldots, S_\ell$, and let $S = \cup_{i=1}^{\ell} S_i$. For each set S_i, we compute a coreset $T_i = c(S_i)$, and set $T = \cup_{i=1}^{\ell} T_i$. Let O be an optimal solution for S, i.e. a k-subset of S for which $\operatorname{div}(O) = \operatorname{div}_k(S)$. We denote by O_i the portion of O lying inside S_i, but not in any other S_j $(j < i)$, i.e., $O_i = O \cap S_i \setminus \cup_{j<i} S_j$. This partitions O into ℓ disjoint subsets O_i.

In the following, we obtain upper bounds on the approximation factor of composable coresets designed for various notions of diversity. More precisely, we show how to compute coresets T_i such that their union T is a good representation of S, i.e., its diversity is within a guaranteed factor of $\operatorname{div}_k(S)$. We accomplish this by comparing the k-diversity of T with that of O, which is in turn equal to the k-diversity of S.

3.1 Remote Clique

In this section, we show that the local search algorithm computes a factor $6 + \varepsilon$ composable coreset for the remote-clique problem when input sets are disjoint. Throughout this subsection, $\operatorname{div}(\cdot)$ refers to the remote-clique diversity.

Let $T_i = \text{LocalSearch}(S_i, k)$. We denote by r_i the average weight of edges in T_i, i.e., $r_i = \operatorname{div}(T_i)/\binom{k}{2}$, and set $r = \max_i \{r_i\}$. Note that, for $i = \arg\max_i \{r_i\}$, $\operatorname{div}_k(T) \geqslant \operatorname{div}_k(T_i) = \binom{k}{2} r_i = \binom{k}{2} r$. We first prove the following lemma.

Lemma 1 *For any point $o \in O_i \setminus T_i$,*

$$\sum_{t \in T_i} d(o, t) \leqslant (1 + \varepsilon) k r.$$

Proof. For any $a \in T_i$, the termination condition of local search implies that

$$\operatorname{div}(T_i \setminus \{a\} \cup \{o\}) \leqslant (1 + \frac{\varepsilon}{k}) \operatorname{div}(T_i).$$

By the definition of remote-clique diversity we have

$$\sum_{p,q \in T_i} d(p, q) - \sum_{t \in T_i} d(a, t) + \sum_{t \in T_i} d(o, t) - d(o, a) \leqslant (1 + \frac{\varepsilon}{k}) \operatorname{div}(T_i).$$

Summing over all points $a \in T_i$, we get

$$k \operatorname{div}(T_i) - 2 \operatorname{div}(T_i) + k \sum_{t \in T_i} d(o, t) - \sum_{t \in T_i} d(o, t) \leqslant (k + \varepsilon) \operatorname{div}(T_i),$$

which simplifies to

$$(k - 1) \sum_{t \in T_i} d(o, t) \leqslant (2 + \varepsilon) \operatorname{div}(T_i).$$

Replacing $r_i = \text{div}(T_i)/\binom{k}{2}$, we get

$$\sum_{t \in T_i} d(o,t) \leqslant (1+\frac{\varepsilon}{2}) \times k r_i \leqslant (1+\varepsilon) k r.$$

Hence, the proof. □

Lemma 2 *Let $Q_i = O_i \setminus T_i$. There exists a bipartite matching between Q_i and T_i that covers Q_i and has weight at most $(1+\varepsilon)|Q_i|r$.*

Proof. Let M be the set of all maximal bipartite matchings between Q_i and T_i. Any maximal matching in M covers Q_i, because $|Q_i| \leqslant |T_i|$. There are $P(k, |Q_i|) = \frac{k!}{(k-|Q_i|)!}$ matchings in M. Each edge $(q,t) \in Q_i \times T_i$ appears in exactly $P(k-1, |Q_i|-1)$ of such matchings. Therefore, the sum of the weights of all matchings in M is:

$$\begin{aligned} &P(k-1,|Q_i|-1) \sum_{q \in Q_i} \sum_{t \in T_i} d(q,t) \\ &\quad \leqslant P(k-1,|Q_i|-1) \sum_{q \in Q_i} (1+\varepsilon) k r \\ &\quad = P(k-1,|Q_i|-1)(1+\varepsilon)|Q_i| k r \\ &\quad = P(k,|Q_i|)(1+\varepsilon)|Q_i| r, \end{aligned}$$

where the first inequality holds by Theorem 1. Therefore, the expected weight of the matchings in M is at most $(1+\varepsilon)|Q_i|r$, and hence, there must exist a matching in M whose weight does not exceed this expectation. □

Theorem 3 *The local search algorithm computes a factor-$(6+\varepsilon)$ disjoint composable coreset for the remote-clique problem.*

Proof. Let M_i be a maximal bipartite matching between Q_i and T_i, obtained by Lemma 2. Let M be the union of M_i's. Since all T_i's are disjoint, M forms a matching between $Q = O \setminus T$ and T that covers all vertices of Q and has weight at most $(1+\varepsilon)|Q|r$.

Let $f : O \to T$ be a function that maps each vertex $o \in O \cap T$ to o itself, and each vertex $o \in O \setminus T$ to the vertex matched to o by M. The weight of this mapping is equal to the weight of M, and hence, is at most $(1+\varepsilon)|Q|r$. Moreover, for each vertex in range(f), there are at most two vertices of O mapped to it. Now, we can use triangle inequality to get:

$$\begin{aligned} \text{div}(O) &= \sum_{o_1,o_2 \in O} d(o_1,o_2) \\ &\leqslant \sum_{o_1,o_2 \in O} [d(o_1,f(o_1)) + d(f(o_1),f(o_2)) + d(f(o_2),o_2)] \\ &= (|O|-1) \sum_{o \in O} d(o,f(o)) + \sum_{o_1,o_2 \in O} d(f(o_1),f(o_2)) \\ &\leqslant (|O|-1)(1+\varepsilon)(|Q|r) + 4\,\text{div}(\text{range}(f)) \\ &\leqslant 2(1+\varepsilon)\binom{k}{2} r + 4\,\text{div}_k(T) \leqslant (6+2\varepsilon)\,\text{div}_k(T), \end{aligned}$$

where in the last two inequalities we used $|Q| \leqslant |O| = k$, and $\text{div}_k(T) \geqslant \binom{k}{2} r$. □

Remark. When input sets are not necessarily disjoint, we prove that the local search algorithm computes a factor $7+4\sqrt{2}+\varepsilon$ composable coreset for the remote-clique problem. Details will be provided in the full version.

3.2 Remote Bipartition and Remote Star

In order to provide improved composable coresets for the remote-bipartition and remote-star problems, we first show that the weight of the clique of a point set approximates the weight of the minimum bipartition and the minimum star of that point set to within factors $3-\frac{4}{k}$ and $2-\frac{2}{k}$, respectively. These improve the previous bounds of 5 and 2, respectively, proved in [19]. Both our new bounds are indeed tight.

Lemma 4 *For any point set of size $k \geqslant 2$, the weight of its clique is a $(3-\frac{4}{k})$-approximation of the weight of its minimum bipartition. This bound is tight.*

Proof. Recall that a bipartition of a point set P of size k is obtained by dividing P into two subsets L and R of equal size $k/2$, and the weight of such bipartition is the total weight of edges between L and R. It is clear from the definition that $w(\text{bipartition}(L,R)) \leqslant w(\text{clique}(P))$. By triangle inequality, for any two vertices $u, v \in R$ and any $w \in L$, we have $d(u,v) \leqslant d(u,w) + d(w,v)$. Summing this inequality over all $u, v \in R$ and $w \in L$ yields:

$$\frac{k}{2} \times w(\text{clique}(R)) \leqslant \left(\frac{k}{2}-1\right) w(\text{bipartition}(L,R)).$$

The same inequality holds for $w(\text{clique}(L))$. Therefore, $w(\text{clique}(P)) = w(\text{clique}(L)) + w(\text{clique}(R)) + w(\text{bipartition}(L,R)) \leqslant \left(3-\frac{4}{k}\right) w(\text{bipartition}(L,R))$. To see tightness, consider an example in which all edges inside L and R have weight 2, and the edges between L and R have weight 1. The approximation factor in this case is $\left((\frac{k}{2})^2 + 4\binom{k/2}{2}\right)/(\frac{k}{2})^2 = 3-\frac{4}{k}$. □

The proof for the remote-star is similar, and is omitted here. Combined with the factor-$(6+\varepsilon)$ composable coreset for the remote-clique problem presented in Theorem 3, and by setting $\varepsilon = O(1/k)$, we get the following result.

Theorem 5 *The local search algorithm computes a factor-12 composable coreset for the remote-star problem, and a factor-18 composable coreset for the remote-bipartition problem, when input sets are disjoint.*

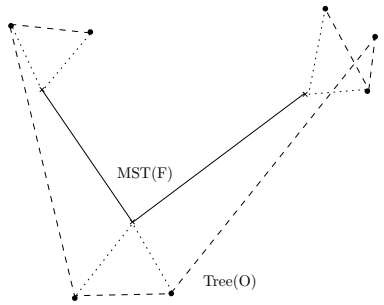

Figure 1: Tree(O) built from MST(F). Dotted lines show the mapping from O to F.

Remark. When input sets are not disjoint, our improved composable coreset for the remote-clique problem which has an approximation factor of $7+4\sqrt{2}+\varepsilon \approx 12.32+\varepsilon$ yields a factor-26 composable coreset for the remote-star problem, and a factor-38 composable coreset for the remote-bipartition problem.

3.3 Remote Tree and Remote Cycle

In this section, we provide a factor-4 composable coreset for the remote-tree problem, and a factor-3 composable coreset for the remote-cycle problem. For both problems, we first run GMM on each S_i to obtain $T_i = \text{GMM}(S_i, k)$. We then obtain the union of the coresets $T = \cup_{i=1}^{\ell} T_i$, and set $r = \max_i \min_{p,q \in T_i} d(p,q)$.

Theorem 6 *The GMM algorithm computes a factor-4 composable coreset for the remote-tree problem.*

Proof. Let $\text{div}(S) = w(\text{MST}(S))$ denote the remote-tree diversity, and let O be a k-subset of S maximizing $\text{div}(O)$. We show that $\text{div}(O) \leqslant 4\,\text{div}_k(T)$.

Consider a mapping $f : O \to T$ that maps each point $o \in O$ to its closest point in T. Let $F = \{f(o) : o \in O\} \subseteq T$ be the range of f, and fix a minimum spanning tree MST(F) of F.

We partition O into subsets $Q_1, \ldots, Q_m$ such that $p, q \in Q_i$ if and only if $f(p) = f(q)$. We now build a spanning tree Tree(O) on O by first building an arbitrary tree on each subset Q_i, and then connecting two components Q_i and Q_j if there are $o_i \in Q_i$ and $o_j \in Q_j$ such that $f(o_i)$ and $f(o_j)$ are connected in MST(F). (See Figure 1.)

By the anticover property of GMM, the length of edges between each o_i and $f(o_i)$ is at most r. So, by triangle inequality, the total cost of edges corresponding to the trees Q_i is at most $(k - |F|) \times 2r$. For each edge $e_f \in \text{MST}(F)$, there is an edge $e_o \in \text{Tree}(O)$ such that $e_o \leqslant e_f + 2r$. There are $|F| - 1$ such edges in total. Therefore,

$$\begin{aligned} w(\text{Tree}(O)) &\leqslant w(\text{MST}(F)) + 2r(k - |F|) + 2r(|F| - 1) \\ &= w(\text{MST}(F)) + 2r(k-1). \end{aligned}$$

Now, let $R \subseteq T$ be an arbitrary superset of F of size k, and let ST(R) be a minimum Steiner tree of R that connects the vertices of F. It is well-known that $w(\text{MST}(F)) \leqslant 2 \cdot w(\text{ST}(R))$ (see, e.g., [23]). Moreover, it is obvious that $w(\text{ST}(R)) \leqslant w(\text{MST}(R))$, because ST($R$) is a minimum-weight tree that only connects a subset of R, as opposed to MST(R) that connects all points in R. Therefore, we have $w(\text{MST}(F)) \leqslant 2 \cdot w(\text{MST}(R))$, and hence,

$$\begin{aligned} w(\text{MST}(O)) &\leqslant w(\text{Tree}(O)) \\ &\leqslant w(\text{MST}(F)) + 2r(k-1) \\ &\leqslant 2 \cdot w(\text{MST}(R)) + 2\,\text{div}_k(T) \leqslant 4\,\text{div}_k(T), \end{aligned}$$

where, the inequality $(k-1)r \leqslant \text{div}_k(T)$ follows from the fact that by the definition of r, there is a set T_i with k points whose pairwise distance is at least r. □

Theorem 7 *The GMM algorithm computes a factor-3 composable coreset for the remote-cycle problem.*

Proof. Let $\text{div}(S) = w(\text{TSP}(S))$ denote the remote-cycle diversity, and let O be a k-subset of S maximizing $\text{div}(O)$. We show that $\text{div}(O) \leqslant 3\,\text{div}_k(T)$.

Consider a function $f : O \to T$ that maps each vertex $o \in O$ to its closest point in T. By the anticover property of GMM, we have $d(o, f(o)) \leqslant r$. Let $R = \{f(o) : o \in O\} \subseteq T$ be the range of f, and let TSP(R) be an optimal tour on R.

We build a graph G on the vertex set $O \cup R$, by first adding to G the edges of TSP(R), and then, adding for each $o \in O$, two copies of the edge $(o, f(o))$ to G. Obviously, G is connected and all its vertices are even. Therefore, G contains an Eulerian tour E. Let C be a cycle obtained from E by short-cutting the vertices not in O. Then,

$$\begin{aligned} w(\text{TSP}(O)) &\leqslant w(C) \leqslant w(E) \\ &\leqslant w(\text{TSP}(R)) + 2kr \\ &\leqslant w(\text{TSP}(R)) + 2\,\text{div}_k(T) \leqslant 3\,\text{div}_k(T), \end{aligned}$$

where, the inequality $w(\text{TSP}(R)) \leqslant \text{div}_k(T)$ holds because TSP($\cdot$) is a monotone increasing function—i.e., for any $A \subseteq B$, we have $w(\text{TSP}(A)) \leqslant w(\text{TSP}(B))$. Moreover, the inequality $kr \leqslant \text{div}_k(T)$ holds because by the definition of r, there is a set T_i with k points whose pairwise distance is at least r. □

Using similar arguments, we can obtain a factor-4 composable coreset for the remote t-trees problem, and a factor-5 composable coreset for the remote t-cycles problem. Details are omitted in this version.

4 Lower Bound

In this section, we prove a general lower bound of 3 on the approximation factor of composable coresets for various notions of diversity in a metric space. This implies

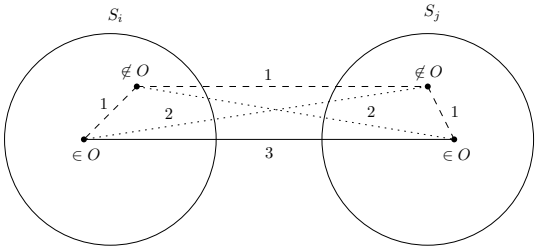

Figure 2: A lower bound example

the optimality of the composable coreset presented in Section 3.3 for the remote-cycle problem. This also settles an open problem posed by Indyk *et al.* [19] on the existence of better composable coresets for the remote-edge problem.

Theorem 8 *Let (X, d) be a metric space, and Π be a graph structure defined over induced subsets of X, such that all graphs with Π structure on a k-point set have the same number of edges. Then, the remote-Π problem admits no α-composable coreset, for any $\alpha < 3$.*

Proof. Consider k sets $S_i \subseteq X$, where each set has at least $k+1$ points. Suppose that the optimal solution O has exactly one point from each set S_i. Let the edges inside each S_i, as well as the edges between non-optimal points from different S_i's have weight 1, the edges connecting points in O have weight 3, and the remaining edges have weight 2. (See Figure 2.) It is easy to verify that this weight function is metric.

Let c be any function that computes a composable coreset $T_i = c(S_i)$ for the remote-Π problem. Due to edge weight symmetry inside each S_i, we can assume that T_i is a k-subset of $S_i \setminus O$. Therefore, the resulting set $T = \cup_i T_i$ will be a subset of $S \setminus O$, and hence, includes only edges of weight 1. Since all edges between the vertices of O have weight 3, the k-diversity of O will be 3 times the k-diversity of T with respect to Π. □

References

[1] S. Abbar, S. Amer-Yahia, P. Indyk, S. Mahabadi, and K. R. Varadarajan. Diverse near neighbor problem. In *Proc. 29th Annu. ACM Sympos. Comput. Geom.*, pages 207–214, 2013.

[2] Z. Abbassi, V. S. Mirrokni, and M. Thakur. Diversity maximization under matroid constraints. In *Proc. 19th ACM SIGKDD Internat. Conf. Knowledge Discovery and Data Mining*, pages 32–40, 2013.

[3] P. K. Agarwal, G. Cormode, Z. Huang, J. M. Phillips, Z. Wei, and K. Yi. Mergeable summaries. *ACM Transactions on Database Systems*, 38(4):26, 2013.

[4] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004.

[5] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proc. 2nd ACM Internat. Conf. Web Search and Data Mining*, pages 5–14, 2009.

[6] A. Angel and N. Koudas. Efficient diversity-aware search. In *Proc. 2011 ACM SIGMOD Internat. Conf. Management of Data*, pages 781–792, 2011.

[7] C. Baur and S. P. Fekete. Approximation of geometric dispersion problems. *Algorithmica*, 30(3):451–470, 2001.

[8] B. Birnbaum and K. J. Goldman. An improved analysis for a greedy remote-clique algorithm using factor-revealing lps. *Algorithmica*, 55(1):42–59, 2009.

[9] T. M. Chan. Faster core-set constructions and data-stream algorithms in fixed dimensions. *Comput. Geom. Theory Appl.*, 35(1):20–35, 2006.

[10] B. Chandra and M. M. Halldórsson. Approximation algorithms for dispersion problems. *J. Algorithms*, 38(2):438–465, 2001.

[11] A. Czygrinow. Maximum dispersion problem in dense graphs. *Oper. Res. Lett.*, 27(5):223–227, 2000.

[12] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[13] S. P. Fekete and H. Meijer. Maximum dispersion and geometric maximum weight cliques. *Algorithmica*, 38(3):501–511, 2004.

[14] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *Proc. 18th Internat. Conf. World Wide Web*, pages 381–390, 2009.

[15] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[16] M. M. Halldórsson, K. Iwano, N. Katoh, and T. Tokuyama. Finding subsets maximizing minimum structures. *SIAM Journal on Discrete Mathematics*, 12(3):342–359, 1999.

[17] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 291–300, 2004.

[18] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21(3):133–137, 1997.

[19] P. Indyk, S. Mahabadi, M. Mahdian, and V. S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *Proc. 33rd ACM Sympos. Principles of Database Systems*, pages 100–108, 2014.

[20] M. J. Kuby. Programming models for facility dispersion: The ρ-dispersion and maxisum dispersion problems. *Geographical Analysis*, 19(4):315–329, 1987.

[21] I. D. Moon and S. S. Chaudhry. An analysis of network location problems with distance constraints. *Management Science*, 30(3):290–307, 1984.

[22] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994.

[23] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.

[24] H. Zarrabi-Zadeh. Core-preserving algorithms. In *Proc. 20th Canad. Conf. Computat. Geom.*, pages 159–162, 2008.

[25] H. Zarrabi-Zadeh. An almost space-optimal streaming algorithm for coresets in fixed dimensions. *Algorithmica*, 60(1):46–59, 2011.