Approximation Algorithms for Clustering to Minimize the Sum of Diameters[1]

Srinivas R. Doddi[1], Madhav V. Marathe[2], S. S. Ravi[3], David Scot Taylor[4], and Peter Widmayer[5]

[1] Los Alamos National Laboratory, P. O. Box 1663, MS B265
Los Alamos, NM 87545, USA
srinu@lanl.gov

[2] Los Alamos National Laboratory, P. O. Box 1663, MS M997
Los Alamos, NM 87545, USA
marathe@lanl.gov

[3] Department of Computer Science
University at Albany - State University of New York
Albany, NY 12222, USA
ravi@cs.albany.edu

[4] Department of Computer Science
University of California
Los Angeles, CA 90095-1596, USA
dstaylor@cs.ucla.edu

[5] Institute for Theoretical Computer Science
ETH, 8092 Zürich, Switzerland
widmayer@inf.ethz.ch

Abstract. We consider the problem of partitioning the nodes of a complete edge weighted graph into k clusters so as to minimize the sum of the diameters of the clusters. Since the problem is NP-complete, our focus is on the development of good approximation algorithms. When edge weights satisfy the triangle inequality, we present the first approximation algorithm for the problem. The approximation algorithm yields a solution that has no more than $10k$ clusters such that the total diameter of these clusters is within a factor $O(\log(n/k))$ of the optimal value for k clusters, where n is the number of nodes in the complete graph. For any fixed k, we present an approximation algorithm that produces k clusters whose total diameter is at most twice the optimal value. When the distances are not required to satisfy the triangle inequality, we show that, unless $\mathsf{P} = \mathsf{NP}$, for any $\rho \geq 1$, there is no polynomial time approximation algorithm that can provide a performance guarantee of ρ even when the number of clusters is fixed at 3. Other results obtained include a polynomial time algorithm for the problem when the underlying graph is a tree with edge weights.

[1] Research Supported by Department of Energy Contract W-7405-ENG-36 and by NSF Grant CCR-97-34936.

1 Introduction

1.1 Motivation

The main goal of clustering is to partition a set of objects into homogeneous and well separated subsets (clusters). Clustering techniques have been used in a wide variety of application areas including information retrieval, image processing, pattern recognition and database systems [Ra97,ZRL96,JD88,DH73]. Over the last three decades, several clustering methods have been developed for specific applications [HJ97,JD88]. Many of these methods define a distance (or a similarity measure) between each pair of objects, and partition the collection into clusters so as to optimize a suitable objective based on the distances. Some of the objectives that have been studied in the literature include minimizing the maximum diameter or radius, total pairwise distances in clusters, etc. The survey paper by Hansen and Jaumard [HJ97] provides an extensive list of clustering objectives and applications for these objectives.

Clustering problems where the objective is to minimize the maximum cluster diameter have been well studied from an algorithmic point of view (see Section 1.4 for a summary). The focus of this paper is on clustering problems where the objective is to partition a given collection of objects into a specified number of clusters so as to minimize the sum of the diameters of individual clusters. The motivation for this objective is derived from the fact that in several applications, clustering algorithms that minimize the maximum diameter produce a "dissection effect" [HJ97,MS89]. This effect causes objects that should normally belong to the same cluster to be assigned to different clusters, as otherwise the diameter of a cluster becomes too large. In such applications, the sum of diameters objective is more useful as it reduces the dissection effect [HJ97,MS89].

1.2 Problem Formulation and Previous Work

To study the clustering problem in a general setting, we represent the objects to be clustered as nodes of a complete edge-weighted undirected graph $G(V, E)$ with $|V| = n$. The distance (or similarity measure) between any pair of objects can then be represented as the weight of the corresponding edge in E. For an edge $\{u, v\}$ in E, we use $\omega(u, v)$ to denote the weight of the edge. It is assumed that the edge weights are nonnegative. For any subset V' of V, the **diameter** of V' (denoted by $\mathrm{DIA}(V')$) is the weight of a largest edge in the complete subgraph of G induced on V'. Note that when $|V'| = 1$, $\mathrm{DIA}(V') = 0$. A formal statement of the clustering problem considered in this paper is as follows.

Clustering to Minimize Sum of Diameters (CMSD)

Instance: A complete graph $G(V, E)$, a nonnegative weight (or distance) $\omega(u, v)$ for each edge $\{u, v\}$ in E and an integer $k \le |V|$.

Requirement: Partition V into k subsets V_1, V_2, ..., V_k such that $\sum_{i=1}^{k} \mathrm{DIA}(V_i)$ is minimized.

In general, edge weights in instances of CMSD need not satisfy the triangle inequality. We use CMSD$_\Delta$ to denote instances of CMSD where edge weights satisfy the triangle inequality. Most of our results are for the CMSD$_\Delta$ problem. We assume without loss of generality that the optimal solution value to any given instance of CMSD$_\Delta$ is strictly greater than zero. We may do so since it is easy to determine whether a given instance of CMSD$_\Delta$ can be partitioned into a specified number of clusters each of which has a diameter of zero.

We now summarize the known results from the algorithmic literature for the CMSD problem. Brucker [Br78] showed that CMSD (without triangle inequality) is NP-complete for any fixed $k \geq 3$. Hansen and Jaumard [HJ87] studied the CMSD problem with $k = 2$ and presented an algorithm with a running time of $O(n^3 \log n)$. They also showed that for $k = 2$, the minimization problem for any given function of the two diameters can be solved in $O(n^5)$ time. When the input is specified as an undirected edge weighted graph with n nodes and m edges, Monma and Suri [MS89] showed that the CMSD problem for $k = 2$ can be solved in time $O(nm \log n)$. This is an improvement over the algorithm of [HJ87] for sparse graphs. Brucker [Br78] observed that the 1-dimensional version of CMSD$_\Delta$ can be solved efficiently for any value of k. For the Euclidean version of CMSD$_\Delta$ with $k = 2$, Monma and Suri [MS89] presented an algorithm which uses $O(n)$ space and runs in $O(n^2)$ time. Capoyleas et al. [CRW91] also studied a generalized version of the CMSD$_\Delta$ problem for points in $\Re^2$. They showed that for any fixed k, the problem can be solved in polynomial time for any monotonic increasing function of cluster radius or diameter. Examples of such monotonic increasing functions include sum of diameters (or radii), maximum diameter (or radius), etc.

1.3 Summary of Main Results

We study the complexity and approximability of the CMSD problem. The main results of this paper can be summarized as follows:

1. We show that unless P = NP, CMSD cannot be efficiently approximated to within any factor even when the number of clusters is fixed at 3. (In contrast, note that CMSD is known to be efficiently solvable when the number of clusters is equal to 2 [HJ87,MS89].)
2. For CMSD$_\Delta$, we show that if the constraint on the number of clusters must be met, then it is NP-hard to approximate the total diameter to within a factor $2 - \epsilon$, for any $\epsilon > 0$.
3. In contrast to the non-approximability results above, we present a polynomial time bicriteria approximation algorithm [MR+98] for CMSD$_\Delta$. This approximation algorithm outputs a solution with at most $10k$ clusters whose total diameter is within a factor of $O(\log(n/k))$ of the minimum possible total diameter with k clusters.
4. We also show that when the number of clusters k is fixed, there is an approximation algorithm for CMSD$_\Delta$ which produces at most k clusters whose total diameter is within a factor of 2 of the minimum possible total diameter.

A brief summary of our other results is given in Section 5.

1.4 Other Related Work

A number of researchers have addressed the clustering problem where the goal is to minimize the maximum diameter or radius of a cluster. In the location theory literature, the problem of minimizing the maximum radius is also known as the k-center problem. For the metric version of the problem of minimizing the maximum diameter, Gonzalez [Go85] presented a simple greedy heuristic that runs in $O(nk)$ time and provides a performance guarantee of 2. He also showed that, unless $\mathsf{P} = \mathsf{NP}$, the performance guarantee cannot be improved. Using a general technique for approximating bottleneck problems, Hochbaum and Shmoys [HS86] also presented a heuristic with a performance guarantee of 2 for the metric version of the k-center problem.

In [FPT81,MS84], it is shown that the problems of minimizing the maximum radius or diameter remain NP-hard even for points in $\Re^2$. For this geometric version, Feder and Greene [FG88] improved the running time of Gonzalez's heuristic to $O(n \log n)$. They also showed that it is NP-hard to achieve a performance guarantee of 1.82 and 1.97 respectively for the diameter and radius problems in $\Re^2$. Recently, Agarwal and Procopiuc [AP98] have presented an exact algorithm with a running time of $n^{O(k^{1-1/d})}$ for the k-center problem for points in $\Re^d$. For any $\epsilon > 0$, they have also presented an $(1+\epsilon)$ approximation algorithm with a running time of $O(n \log k) + (k/\epsilon)^{O(k^{(1-1/d)})}$ for the problem.

Plesník [Pl82] has addressed the problem of partitioning the *edges* of a given graph $G(V, E)$ into k subsets so that each subset forms a connected graph on the vertex set V, and a given function of the diameters of the resulting subgraphs is minimized. The objectives considered in [Pl82] include minimizing the maximum diameter and minimizing the total diameter. It is shown that, unless $\mathsf{P} = \mathsf{NP}$, even for $k = 2$, these objectives cannot be efficiently approximated to within factors less than 3/2 and 5/4 respectively.

Several other types of clustering problems have also been studied in the literature. For example, Charikar et al. [CC+97] study an incremental version of the clustering problem for minimizing the maximum radius. Pferschy et al. [PRW94] study geometric versions of clustering problems using objectives such as minimizing the total perimeter. Agarwal and Procopiuc [AP00] study projective clustering problems where the goal is to cover a set of points in $\Re^d$ using hyper-strips, and the objective is to minimize the maximum width of the strips. References where other types of clustering problems are studied include [Ma99,ABC+98,GH98,DKS97,Da94,BKK94].

2 Preliminaries

2.1 A Simple Upper Bound on the Optimal Solution Value

Given any instance of CMSD, we can easily construct a feasible solution consisting of k clusters with total diameter at most the maximum edge weight: form one cluster consisting of $n-k+1$ arbitrarily chosen vertices and make each of the remaining $k-1$ vertices a singleton cluster. This observation is stated formally below.

Remark 1. *For any instance of I of* CMSD, *the optimal solution value is at most the maximum edge weight in I.* $\square$

2.2 A Merging Lemma

The formulation of the CMSD problem requires that the clusters be pairwise disjoint. Our approximation algorithms may produce clusters which may not satisfy the disjointness condition. The following lemma points out that for instances of CMSD_Δ, we can merge pairs of intersecting sets without increasing the total diameter.

Lemma 1. *Let I be an instance of CMSD_Δ given by the edge weighted complete graph $G(V, E)$ and integer k. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ be a collection of subsets of V such that their union is V and the sum of the diameters of all the subsets in $\mathcal{C}$ is ψ. Further, suppose C_i and C_j $(i \neq j)$ are two sets in $\mathcal{C}$ such that $C_i \cap C_j \neq \emptyset$. Then the total diameter of the collection $\mathcal{C}'$ obtained by deleting C_i and C_j from $\mathcal{C}$ and adding the set $C_i \cup C_j$ is at most ψ.*

Proof. The lemma would follow by showing that $\text{DIA}(C_i \cup C_j) \leq \text{DIA}(C_i) + \text{DIA}(C_j)$. To do this, let x be a node in $C_i \cap C_j$ and let u and v be two nodes in $C_i \cup C_j$ such that $\omega(u, v) = \text{DIA}(C_i \cup C_j)$. If u and v are both in C_i (or both in C_j), then $\omega(u, v) \leq \text{DIA}(C_i)$ ($\omega(u, v) \leq \text{DIA}(C_j)$), and the proof is trivial. So, assume that $u \in C_i$ and $v \in C_j$. By the triangle inequality, $\omega(u, v) \leq \omega(u, x) + \omega(v, x)$. Since u and x are both in C_i, $\omega(u, x) \leq \text{DIA}(C_i)$. Similarly, $\omega(v, x) \leq \text{DIA}(C_j)$. Therefore, $\text{DIA}(C_i \cup C_j) = \omega(u, v) \leq \text{DIA}(C_i) + \text{DIA}(C_j)$, and this completes the proof. $\square$

In view of the above lemma, when considering instances of CMSD_Δ, we may repeatedly merge pairs of clusters with nonempty intersection until the clusters are pairwise disjoint. The merging process does not increase the total diameter of the clusters.

2.3 Transformation to Weighted Set Cover

Our results rely on a transformation from instances of CMSD_Δ to instances of the weighted set cover problem. Given an instance of CMSD_Δ along with a nonnegative value f, the transformation in Figure 1 produces an instance of the weighted set cover problem. It is clear that the transformation can be carried out in polynomial time. The following lemma points out an important property of the resulting set cover instance.

Lemma 2. *Let I denote an instance of CMSD_Δ problem and let f be a nonnegative number. Let I' denote the instance of the weighted set cover problem produced by the transformation in Figure 1 when I and f are given as inputs. Let $\text{OPT}(I)$ and $\text{OPT}(I')$ denote the optimum solution values to I and I' respectively. Then, $\text{OPT}(I') \leq 2\,\text{OPT}(I) + f$.*

TransformToSetCover($G(V,E), k, f$)
f is a nonnegative parameter.

Output: An instance of weighted set cover problem with base set Q, and collection $\mathcal{W}$ of nonempty subsets of Q, each with a weight. The weight of a set $W \in \mathcal{W}$ is denoted by $c(W)$.

1. $Q = V$ /* Note: $|Q| = n$. */
2. $\mathcal{W} = \emptyset$
3. **for** each $v \in V$ **do**
   (a) Sort $\{\omega(v,u) \,:\, u \in V\}$ into (strictly) increasing order.
   (b) Let $\alpha_1 = 0 < \alpha_2 < \ldots < \alpha_{r_v}$ denote the sorted order.
   (c) **for** $i = 1$ **to** r_v **do**
      i. Let $W_v^i = \{u \,:\, \omega(u,v) \le \alpha_i\}$
      ii. Let $c(W_v^i) = \mathrm{DIA}(W_v^i) + f/k$
      iii. Add (W_v^i) to $\mathcal{W}$
4. **return**$(Q, \mathcal{W})$

Fig. 1. Transformation from Cmsd_Δ to Weighted Set Cover

Proof. Let C_1, C_2, …, C_k denote the clusters in an optimal solution to I. Thus, $\mathrm{OPT}(I) = \sum_{i=1}^{k} \mathrm{DIA}(C_i)$. We will show that there is a subcollection of k sets in I' such that the sets in the subcollection together cover the base set Q and the total weight of the sets in the subcollection is at most $2\,\mathrm{OPT}(I) + f$. The lemma would then follow immediately.

Consider each cluster C_i $(1 \le i \le k)$ in the optimal solution to I. If C_i contains two or more nodes, let v_i be a node in C_i such that v_i is one of the endpoints of an edge whose weight is equal to $\mathrm{DIA}(C_i)$. If C_i contains only one node (i.e., $\mathrm{DIA}(C_i) = 0$), let v_i be that node. Now, by the transformation of Figure 1, I' has a set, say W_i, that includes all the nodes which are at a distance of at most $\mathrm{DIA}(C_i)$ from v_i. By the triangle inequality, $\mathrm{DIA}(W_i) \le 2\,\mathrm{DIA}(C_i)$. So, $c(W_i) = \mathrm{DIA}(W_i) + f/k \le 2\,\mathrm{DIA}(C_i) + f/k$. Clearly, the subcollection $\{W_1, W_2, \ldots, W_k\}$ covers the base set Q. The weight of this cover is $\sum_{i=1}^{k} c(W_i)$, which is at most $\sum_{i=1}^{k}(2\,\mathrm{DIA}(C_i) + f/k) = 2\,\mathrm{OPT}(I) + f$. This completes the proof of the lemma. ☐

2.4 The Budgeted Maximum Coverage Problem

For obtaining our approximation result for Cmsd_Δ (where the number of clusters k is a part of the problem instance), we use a known approximation result for the Budgeted Maximum Coverage Problem (Bmcp). Below, we provide a definition of the problem and state the necessary approximation result.

An instance of Bmcp consists of a base set $Q = \{q_1, q_2, \ldots, q_n\}$, a collection $\mathcal{W}$ of nonempty subsets of Q, a nonnegative weight $c(W)$ for each set $W \in \mathcal{W}$ and a nonnegative budget B. The goal is to choose a subcollection of sets from $\mathcal{W}$

so that the total cost of the chosen sets is at most B and the number of elements covered by the chosen sets is maximum. This problem is NP-hard since it is a restatement of the minimum cost set cover problem. The following approximation result for BMCP is proved in [KMN99].

Theorem 1. BMCP *can be efficiently approximated to within a factor* $(1 - 1/e)$. $\square$

It is shown in [KMN99] that the approximation algorithm referred to in Theorem 1 can also be used for the more general version of BMCP where there is a weight associated with each element of the base set, and the goal is to maximize the weight of the elements covered by the chosen sets. For our results, the unit weight version of BMCP where the weight of each element of the base set is 1, suffices.

3 Approximating CMSD$_\Delta$

3.1 Algorithm Overview

We give a brief top-down description of our approximation algorithm APPROX-CMSD$_\Delta$, and introduce the terminology used in the analysis. At all times, APPROX-CMSD$_\Delta$ maintains a set $\mathcal{D}$ of clusters which cover all vertices in V, at cost Ψ. We call these *global clusters*, since they cover all vertices in V. The algorithm begins with $\mathcal{D}$ consisting of $|V|$ singleton clusters, and progresses through a series of rounds. During each round, it constructs a vertex set N by selecting an arbitrary vertex from each of its current clusters. It then finds a clustering $\mathcal{C}$ on N. We call the clusters in $\mathcal{C}$ *local clusters*, since they do not need to cover all of V, but only N. As will be shown, the number of clusters $|\mathcal{C}|$ is at most $3k[1+\ln(|N|/k)]$. We use ψ to denote their total cost. Next, APPROX-CMSD$_\Delta$ uses MERGE to suitably combine the clusters in $\mathcal{D}$ into a set of just $|\mathcal{C}|$ clusters, which cover all of V at cost at most $\Psi + \psi$. This entire process is repeated until the number of clusters in $\mathcal{D}$ is at most $10k$.

APPROX-CMSD$_\Delta$ uses FINDCOVER to return the required $\mathcal{C}$ clusters during each round. FINDCOVER, in turn, iterates through at most $O(\ln(|N|/k))$ calls to PARAMETRICBMCP each of which returns a set of at most $3k$ clusters which cover all but a $1/e$ fraction of the remaining uncovered vertices from N. These clusters have cost no more than $3(1+\epsilon)\mathrm{OPT}$.

Using TRANSFORMTOSETCOVER from Figure 1, PARAMETRICBMCP converts the problem to a set cover instance, and repeatedly calls the Budgeted Maximum Coverage Approximation Algorithm BMCP, with growing budgets, until the budget is large enough to make BMCP cover the required fraction of vertices. A complete description of the approximation algorithm is given in Figure 2.

APPROX-CMSD$_\Delta$$(G(V,E),k)$
Output: A set of no more than $10k$ clusters with sum of diameters $O(\ln(|V|/k))$ OPT.

1. $\mathcal{D} = \{\{v\} \ : \ v \in V\}$
2. **while** $(|\mathcal{D}| > 10k)$ **do** /* We call each iteration a Round */
   (a) N = Set of vertices obtained by choosing one arbitrary vertex from each $D \in \mathcal{D}$.
   (b) $\mathcal{C}$ =FINDCOVER$(G(N,E'),k)$ /* $G(N,E')$: Complete subgraph on N */
   (c) $\mathcal{D}$ =MERGE$(\mathcal{D},\mathcal{C})$
3. **return**$(\mathcal{D})$

FINDCOVER$(G(N,E),k)$
Output: A set of no more than $3k[1+\ln(|N|/k)]$ clusters which cover N with cost no more than $3[1+\ln(|N|/k)](1+\epsilon)$OPT.

1. $\mathcal{C} = \emptyset$
2. **while** $(N \neq \emptyset)$ **do**
   (a) $\mathcal{C}' =$ PARAMETRICBMCP$(G(N,E),k)$
   (b) $\mathcal{C} = \mathcal{C} \cup \mathcal{C}'$
   (c) $N = N - \{i \ : \ i \in C$ for some $C \in \mathcal{C}'\}$
   (d) E = Edges in the complete subgraph induced on the new, smaller N
3. **return**$(\mathcal{C})$

PARAMETRICBMCP$(G(N,E),k)$
Output: A set of no more than $3k$ clusters which cover $(1-1/e)|N|$ or more vertices from N with cost no more than $3(1+\epsilon)$OPT for any fixed $\epsilon > 0$.

1. f = the smallest non-zero edge weight in $G(N,E)$
2. $\mathcal{C}' = \{\{v\} \ : \ v \in N\}$
3. **while** $(|\mathcal{C}'| > 3k$ **or** $|\{v \ : \ v \in C$ for some $C \in \mathcal{C}'\}| < (1-1/e)|N|)$ **do**
   (a) $\mathcal{S}$ =TRANSFORMTOSETCOVER$(G(N,E),k,f)$
   (b) $\mathcal{C}' =$ BMCP$(\mathcal{S},3f)$
   (c) $f = (1+\epsilon)f$
4. **return**$(\mathcal{C}')$

MERGE$(\mathcal{D},\mathcal{C})$
Remark: $\mathcal{D},\mathcal{C}$ collections of vertex sets such that $\forall D \in \mathcal{D}$, $\exists C \in \mathcal{C}$ such that $(D \cap C \neq \emptyset)$.
Output: A set of $|\mathcal{C}|$ vertex sets which cover all $\{v \ : \ v \in X$ for some $X \in \mathcal{D} \cup \mathcal{C}\}$ at cost no more than the sum of the costs of $\mathcal{C}$ and $\mathcal{D}$.

1. **for** each $C \in \mathcal{C}$ **do**
   for each $D \in \mathcal{D}$ **do**
      if $(C \cap D \neq \emptyset)$ **then**
         $C = C \cup D$; $\mathcal{D} = \mathcal{D} - D$
2. **return**$(\mathcal{C})$

Fig. 2. Outline of APPROX-CMSD$_\Delta$

3.2 Correctness of Algorithm

To show that our algorithm runs in polynomial time and achieves the stated performance guarantees, we analyze it from the lower level functions up to the top level call, beginning with ParametricBmcp, and finishing with Approx-Cmsd$_\Delta$.

Lemma 3. *Given graph G with optimal k-cluster cost* OPT*,* ParametricBmcp *returns no more than $3k$ clusters which contain at least $(1-1/e)|N|$ of the vertices from $|N|$. Further, the sum of diameters of the returned clusters is no more than $3(1+\epsilon)\mathrm{OPT}$.*

Proof. By Lemma 2, the call to TransformToSetCover returns a set cover instance with optimal solution no more than $2\,\mathrm{OPT}+f$. When $f > \mathrm{OPT}$, by Theorem 1, the call to Bmcp with budget $3f > 3\,\mathrm{OPT} > 2\,\mathrm{OPT}+f$, will return sets which cover the stated number of vertices. Also, when $f > 0$, this solution cannot have more than $3k$ clusters: each of the clusters has minimum cost f/k, so any more than $3k$ clusters will have cost more than $3f$. Therefore, with any $f > \mathrm{OPT}$, the call to Bmcp with budget $3f$ will return at most $3k$ clusters which cover enough vertices.

Since we start f at the smallest possible (non-zero) value (in fact, we first implicitly test if $f = 0$ suffices), and increase it by factors of $(1+\epsilon)$, we are guaranteed to try a value f such that $f < (1+\epsilon)\mathrm{OPT}$. This will occur within $O(\log_{1+\epsilon}\mathrm{OPT})$ iterations. Since OPT is at most the maximum edge weight (Remark 1), the number of iterations is polynomial. □

Lemma 4. *Given graph $G(N,E)$ with optimal k-cluster cost* OPT*,* FindCover *returns no more than $3k[1+\ln(|N|/k)]$ clusters which cover N with cost no more than $3[1+\ln(|N|/k)](1+\epsilon)\mathrm{OPT}$.*

Proof. By Lemma 3, each call to ParametricBmcp will return at most $3k$ clusters of cost $3(1+\epsilon)\mathrm{OPT}$, and will leave at most $|N|/e$ of the $|N|$ vertices uncovered. In the ensuing iterations of ParametricBmcp, we use a subset of N which certainly has an optimal k-clustering with cost no greater than OPT. After i iterations, we are guaranteed to have no more than $3k$ remaining vertices, where $|N|/e^i \leq 3k$. To upper bound i, notice that if i is not the last iteration, $|N|/e^{i-1} > 3k$, and $i \leq 1+\ln(|N|/3k) \leq \ln(|N|/k)$. The final iteration generates at most $3k$ additional singleton clusters with cost zero. Each of the $1+\ln(|N|/k)$ iterations returns no more than $3k$ clusters, of cost at most $3(1+\epsilon)\mathrm{OPT}$. The lemma follows. □

Lemma 5. Merge *returns $|\mathcal{C}|$ vertex sets which cover all $\{v : v \in X$ for some cluster $X \in \mathcal{D} \cup \mathcal{C}\}$, with cost no more than the sum of the costs of $\mathcal{C}$ and $\mathcal{D}$.*

Proof. Consider all $\mathcal{C} \cup \mathcal{D}$ clusters whose cost is the sum of the costs of $\mathcal{C}$ and $\mathcal{D}$. Since each $D \in \mathcal{D}$ intersects some $C \in \mathcal{C}$, we may replace D and C with $D \cup C$, at no additional cost, by Lemma 1. This process can be continued until each cluster in $\mathcal{D}$ has been merged into some cluster in $\mathcal{C}$. □

Finally, we need to show that the top level function Approx-Cmsd$_\Delta$ does in fact halt within a polynomial number of iterations. To do this, we show that the number of clusters in $\mathcal{D}$ is eventually less than $10k$, and that this happens after no more than $O(\log_2 \log_2 (n/k))$ rounds.

Our algorithm begins with n vertices, and by Lemma 4, after the end of the first round, we are left with $3k[1 + \ln (n/k)]$ clusters, each of which contributes one vertex towards the second round. Generalizing this for all rounds, let $\mathcal{D}_i$ be the set of global clusters at the end of round i, and $n_i = |\mathcal{D}_i|$. Then, $n_0 = n$, and n_{i-1} is both the number of clusters at the end of round $i-1$ and the number of vertices we need to cluster in the i^{th} round. We get the recurrence

$$n_{i+1} \leq 3k[1 + \ln(n_i/k)].$$

Let $t_i = n_i/k$, we have $t_{i+1} \leq 3 + 3 \cdot \ln t_i \leq 6 \cdot \ln t_i$ for $t_i \geq e$. By having enough rounds to make t_i constant, we will have a total of $O(k)$ clusters. After $O(\log^* t_0)$ rounds, t_i becomes a constant, but here we will instead give a simple proof that $O(\log_2 \log_2 t_0) = O(\log_2 \log_2 (n/k))$ rounds are sufficient.

Lemma 6. *After at most $5 + \log_2 \log_2 (n/k)$ rounds, $|\mathcal{D}|$ contains at most $10k$ clusters.*

Proof. Consider the "iterating" function used to get $\log^* x$ from $\log_2 x$. For any function f such that $f(x) < x$ for sufficiently large x, the iterating function is the number of times you must apply that function to get a constant. More specifically, define the function $(f)^*(x)_C$ to be the number of times that $f()$ must be iteratively applied to get a result less than C. (Thus, $(\log_2)^*(x)_1$ gives the familiar function $\log^* x$.) Next, we use the fact that for $x > 2109$, $6 \cdot \ln x < \sqrt{x}$. Thus, $(6 \cdot \ln)^*(x)_{2109} \leq (\sqrt{\ })^*(x)_{2109} \leq (\sqrt{\ })^*(x)_1$. However, $(\sqrt{\ })^*(x)_1 = \lceil \log_2 \log_2 x \rceil$, so we need to iterate less than $\log_2 \log_2 t_0$ times before reaching $t_i \leq 2109$. One more iteration for n gives us $n_{1+\log_2 \log_2 t_1} \leq 3k + 3k \cdot \ln 2109 \leq 26k$. Applying the recursion four more times gives $n_{5+\log_2 \log_2 (n/k)} \leq 10k$. $\square$

Thus, Approx-Cmsd$_\Delta$ will terminate in $O(\log \log (n/k))$ rounds. Each round has a call to FindCover, which makes at most $O(\log (n/k))$ calls to ParametricBmcp. Using $T(x)$ to denote the running time of Bmcp, the time taken by all the calls to ParametricBmcp is $O((n^2 \log n + T(n^2)) \log_{1+\epsilon} \mathrm{OPT})$. Thus, the running time of the approximation algorithm is

$$O(\log \log (n/k)[\log (n/k)(n^2 \log n + T(n^2)) \log_{1+\epsilon} \mathrm{OPT}]).$$

Since $T(x)$ is polynomial by [KMN99], so is our algorithm.

Now all that is left is to show that the total cost is no more than the stated bound. Let $\mathcal{C}_i$ denote the set of local clusters from round i. Since $\mathcal{C}_i$ covers the set of N vertices, one from each $D \in \mathcal{D}_i$, we know that each $D \in \mathcal{D}_i$ intersects a cluster C in $\mathcal{C}_i$. Let Ψ_i and ψ_i be the sum of diameters of the global and local clusters during the i^{th} round respectively. The following lemma can be proven by induction on i.

Lemma 7. *After i rounds of* APPROX-CMSD$_\Delta$, $\Psi_i \le \sum_{j=1}^{i} \psi_i$. □

To get the total cost of all global clusters at the end of the algorithm, we just need to compute $\Psi_{5+\log_2 \log_2 (n/k)}$, since it was shown in Lemma 6 that the number of rounds is at most $5 + \log_2 \log_2 (n/k)$.

Lemma 8. $\Psi_{5+\log_2 \log_2 (n/k)} = \mathrm{OPT} \cdot O(\ln (n/k))$.

Proof. Note that by Lemma 7, $\Psi_{5+\log_2 \log_2 (n/k)} = \sum_{i=1}^{5+\log_2 \log_2 (n/k)} \psi_i$. By separating the summation into the first term and all others, and noticing that n_i is decreasing (i.e., all terms with $n_{i\ge 1}$ are upper bounded by n_1), we get that the first term in the summation is $3[1 + \ln (n/k)](1 + \epsilon)\mathrm{OPT}$, and the rest of the terms are $\mathrm{OPT} \cdot O((\log_2 \log_2 (n/k))^2)$. For large enough n/k, the first term dominates all of the rest, so for some constant $\epsilon' > \epsilon$, the cost is no more than $3[1 + \ln (n/k)](1 + \epsilon')\mathrm{OPT} = \mathrm{OPT} \cdot O(\log_2(n/k))$, with small constant terms. □

Summarizing the above discussion, we have:

Theorem 2. *There is a polynomial time approximation algorithm for the* CMSD$_\Delta$ *problem that returns at most* $10k$ *clusters whose total diameter is at most* $O(\ln(n/k))$ *times the optimal solution value with* k *clusters.* □

3.3 An Approximation Algorithm for CMSD$_\Delta$ with Fixed k

When k is fixed, it is possible to obtain a simple 2-approximation algorithm for the CMSD$_\Delta$ problem using the transformation shown in Figure 1. We present this result below.

Theorem 3. *When the number of clusters k is* fixed, *there is a 2-approximation algorithm for* CMSD$_\Delta$.

Proof. The steps of the approximation algorithm are as follows.

1. Using the transformation of Figure 1, construct an instance of the minimum cost set cover problem from the given instance of CMSD$_\Delta$ with the parameter f set to zero.
2. Find a minimum cost set cover consisting of at most k sets. Since k is fixed, this step can be done in polynomial time by exhaustive search.
3. If the collection of sets obtained in Step 2 are not pairwise disjoint, then repeatedly merge pairs of sets with nonempty intersection until the collection is pairwise disjoint.
4. Output the collection of sets found in Step 3 as the solution to the CMSD$_\Delta$ instance.

Clearly, the approximation algorithm runs in polynomial time. Applying Lemma 2 with $f = 0$, the cost of an optimal set cover is at most twice the optimal solution value of the CMSD$_\Delta$ instance. Step 2 finds an optimal solution to the set cover problem, and by Lemma 1, the merging operations in Step 3 do not increase the total diameter of the clusters. Thus, the total diameter of the clusters produced is at most twice the optimal value. □

4 Non-Approximability Results

4.1 Non-Approximability Without Triangle Inequality

We show that, unless $\mathsf{P} = \mathsf{NP}$, CMSD cannot be efficiently approximated to within any factor even when the number of clusters is fixed at 3. This result can be established through a simple modification to the reduction from Graph 3-colorability (3-COLOR) to CMSD given in [Br78]. We omit this proof due to space constraints.

Proposition 1. *Unless* $\mathsf{P} = \mathsf{NP}$, *for any* $\rho \geq 1$, *no polynomial time algorithm for the* CMSD *problem can provide a performance guarantee of* ρ. □

This non-approximability result should be contrasted with the known result that the CMSD problem is solvable in polynomial time for 2 clusters [HJ87,MS89].

4.2 A Non-Approximability Result for CMSD$_\Delta$

Here, we prove our non-approximability result for CMSD$_\Delta$. We establish this result through a reduction from the well known CLIQUE problem [GJ79].

Proposition 2. *Unless* $\mathsf{P} = \mathsf{NP}$, *for any* $\epsilon > 0$, *no polynomial time algorithm for the* CMSD$_\Delta$ *problem can provide a solution which satisfies the bound on the number of clusters and whose total diameter is within a factor* $2-\epsilon$ *of the optimal value.*

Proof. We use a reduction from the CLIQUE problem. Let the undirected graph $G(V,E)$ and integer $J \leq |V|$ denote an arbitrary instance of the CLIQUE problem. We construct an instance of the CMSD$_\Delta$ problem consisting of a complete edge weighted graph G' on the vertex set V as follows. For any pair of vertices u and v, the weight of $\{u,v\}$ is set to 1 if $\{u,v\}$ is an edge in E and to 2 otherwise. Obviously, the resulting edge weights satisfy the triangle inequality. The number of clusters k is set to $|V| - J + 1$. Now, it is straightforward to see that if G has a clique with J or more vertices, then G' can be partitioned into at most k clusters with a total diameter of 1: the vertices of the clique form one cluster of diameter 1 and each of the remaining $|V| - J$ vertices forms a separate cluster with a diameter of zero. Further, if G does not have a clique with J or more vertices, then any solution with at most k clusters must have a total diameter of at least 2. The proposition follows. □

5 Other Results

In this section, we briefly mention our other results on the CMSD problem. Details concerning these results will appear in a complete version of the paper.

We have considered the CMSD problem when the underlying graph is a tree with edge weights (rather than a complete graph). In this version, the distance between any pair of nodes is the length of the path between the nodes in the tree. For this problem, we have developed a polynomial time algorithm using dynamic programming. This algorithm uses $O(kn^2)$ space and runs in $O(k^2n^3)$ time. It can also be extended to work for graphs of bounded treewidth.

We have also considered the clustering problem where the goal is to minimize the sum of the radii of the clusters (rather than the sum of the diameters). To discuss these results, we first recall the definition of cluster radius. Let C be a cluster. For any node v in C, let d_v denote the maximum distance between v and any other node in C. The radius of C is given by $\min\{d_v \ : \ v \in C\}$. A node v for which d_v is equal to the radius of C is a **center** of C. When edge weights satisfy the triangle inequality, the diameter of a cluster is at most twice the radius. Therefore, our approximation result for CMSD_Δ (Section 3) carries over (with a different constant within the big-O) to the clustering problem where the goal is to minimize the sum of the radii. We have also been able to show an interesting contrast between the diameter and radius problems for the non-metric case. For fixed k, while it is NP-hard to obtain even an approximation for the non-metric version of the diameter problem (Section 4.1), the corresponding problem for radius can be solved in polynomial time.

References

[ABC+98] B. Awerbuch, B. Berger, L. Cowen and D. Peleg, "Near-Linear Time Construction of Sparse Neighborhood Covers", *SIAM J. Computing*, Vol. 28, No. 1, 1998, pp. 263–277.

[AP98] P. K. Agarwal and C. M. Procopiuc, "Exact and Approximate Algorithms for Clustering", *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms* (SODA'98), San Francisco, CA, Jan. 1998, pp. 658–667.

[AP00] P. K. Agarwal and C. M. Procopiuc, "Approximation Algorithms for Projective Clustering", *Proc. 11th ACM-SIAM Symposium on Discrete Algorithms* (SODA'2000), San Francisco, CA, Jan. 2000, pp. 538–547.

[BKK94] V. Batagelj, S. Korenjak-Cerne and S. Klavzar, "Dynamic Programming and Convex Clustering", *Algorithmica*, Vol. 11, No. 2, Feb. 1994, pp. 93–103.

[Br78] P. Brucker, "On the Complexity of Clustering Problems", in *Optimization and Operations Research*, Lecture Notes in Economics and Mathematical Systems, Vol. 157, Edited by M. Beckmann and H. Kunzi, Springer-Verlag, Heidelberg, 1978, pp. 45–54.

[CC+97] M. Charikar, C. Chekuri, T. Feder and R. Motwani, "Incremental Clustering and Dynamic Information Retrieval", *Proc. 29th Annual ACM Symposium on Theory of Computing* (STOC'97), El Paso, TX, May 1997, pp. 626–634.

[CRW91] V. Capoyleas, G. Rote and G. Woeginger, "Geometric Clusterings", *J. Algorithms*, Vol. 12, No. 2, Jun. 1991, pp. 341–356.

[Da94] A. Datta, "Efficient Parallel Algorithms for Geometric k-Clustering Problems", *Proc. 11th Annual Symposium on Theoretical Aspects of Computer Science* (STACS'94), Caen, France, Feb. 1994, Springer-Verlag Lecture Notes in Computer Science, Vol. 775, pp. 475–486.

[DH73] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, New York, NY, 1973.

[DKS97] J. S. Deogan, D. Kratsch and G. Steiner, “An Approximation Algorithm for Clustering Graphs with a Dominating Diametral Path”, *Information Processing Letters*, Vol. 61, No. 3, Feb. 1997, pp. 121–127.

[FG88] T. Feder and D. H. Greene, “Optimal Algorithms for Approximate Clustering”, *Proc. 20th Annual ACM Symposium on Theory of Computing* (STOC’88), Chicago, IL, May 1988, pp. 434–444.

[FPT81] R. Fowler, M. Paterson and S. Tanimoto, “Optimal Packing and Covering in the Plane”, *Information Processing Letters*, Vol. 12, 1981, pp. 133–137.

[GH98] N. Guttmann-Beck and R. Hassin, “Approximation Algorithms for Min-sum p-Clustering”, *Discrete Applied Mathematics*, Vol. 89, 1998, pp. 125–142.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Co., San Francisco, CA, 1979.

[Go85] T. F. Gonzalez, “Clustering to Minimize the Maximum Intercluster Distance”, *Theoretical Computer Science*, Vol. 38, No. 2-3, Jun. 1985, pp. 293–306.

[HJ87] P. Hansen and B. Jaumard, “Minimum Sum of Diameters Clustering,” *Journal of Classification*, Vol. 4, 1987, pp. 215–226.

[HJ97] P. Hansen and B. Jaumard, “Cluster Analysis and Mathematical Programming, *Mathematical Programming*, Vol. 79, Aug. 1997, pp. 191–215.

[Ho97] D. S. Hochbaum (Editor), *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, Boston, MA, 1997.

[HS86] D. S. Hochbaum and D. B. Shmoys, “A Unified Approach to Approximation Algorithms for Bottleneck Problems”, *J. ACM*, Vol. 33, No. 3, July 1986, pp. 533–550.

[JD88] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988.

[KMN99] S. Khuller, A. Moss and J. Naor, “The Budgeted Maximum Coverage Problem”, *Information Processing Letters*, Vol. 70, 1999, pp. 39–45.

[Ma99] J. Matousek, “On Approximate Geometric k-Clustering”, Manuscript, Department of Applied Mathematics, Charles University, Prague, Czech Republic, 1999.

[MR+98] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz and H. B. Hunt III, “Bicriteria Network Design Problems”, *J. Algorithms*, Vol. 28, No. 1, July 1998, pp. 142–171.

[MS84] N. Meggiddo and K. J. Supowit, “On the complexity of some common geometric location problems,” *SIAM J. Computing*, Vol. 13, 1984, pp. 182–196.

[MS89] C. L. Monma and S. Suri, “Partitioning Points and Graphs to Minimize the Maximum or the Sum of Diameters”, *Proc. 6th Int. Conf. Theory and Applications of Graphs*, Kalamazoo, Michigan, May 1989.

[Pl82] J. Plesník, “Complexity of Decomposing Graphs into Factors with Given Diameters or Radii”, *Math. Slovaca*, Vol. 32, No. 4, 1982, pp. 379–388.

[PRW94] U. Pferschy, R. Rudolf and G. J. Woeginger, “Some Geometric Clustering Problems”, *Nordic J. Computing*, Vol. 1, No. 2, Summer 1994, pp. 246–263.

[Ra97] P. Raghavan, “Information Retrieval Algorithms: A Survey”, *Proc. 8th ACM-SIAM Symposium on Discrete Algorithms* (SODA’97), Jan. 1997, pp. 11–18.

[ZRL96] T. Zhang, R. Ramakrishnan and M. Livny, “Birch: An Efficient Data Clustering Method for Very Large Databases”, *Proc. ACM-SIGMOD International Conference on Management of Data* (SIGMOD’96), Aug. 1996, pp. 103–114.