Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Discrete Algorithms 2 (2004) 289–301
JOURNAL OF DISCRETE ALGORITHMS
www.elsevier.com/locate/jda

Approximation algorithms for Hamming clustering problems

Leszek Gąsieniec [a,*], Jesper Jansson [b], Andrzej Lingas [b]

[a] *Department of Computer Science, University of Liverpool, Peach Street, Liverpool, L69 7ZF, UK*
[b] *Department of Computer Science, Lund University, Box 118, 221 00 Lund, Sweden*

Abstract

We study Hamming versions of two classical clustering problems. The *Hamming radius p-clustering problem* (HRC) for a set S of k binary strings, each of length n, is to find p binary strings of length n that minimize the maximum Hamming distance between a string in S and the closest of the p strings; this minimum value is termed the *p-radius of S* and is denoted by ϱ. The related *Hamming diameter p-clustering problem* (HDC) is to split S into p groups so that the maximum of the Hamming group diameters is minimized; this latter value is called the *p-diameter of S*.

We provide an integer programming formulation of HRC which yields exact solutions in polynomial time whenever k is constant. We also observe that HDC admits straightforward polynomial-time solutions when $k = O(\log n)$ and $p = O(1)$, or when $p = 2$. Next, by reduction from the corresponding geometric p-clustering problems in the plane under the L_1 metric, we show that neither HRC nor HDC can be approximated within any constant factor smaller than two unless P = NP. We also prove that for any $\varepsilon > 0$ it is NP-hard to split S into at most $pk^{1/7-\varepsilon}$ clusters whose Hamming diameter does not exceed the p-diameter, and that solving HDC exactly is an NP-complete problem already for $p = 3$. Furthermore, we note that by adapting Gonzalez' farthest-point clustering algorithm [T. Gonzalez, Theoret. Comput. Sci. 38 (1985) 293–306], HRC and HDC can be approximated within a factor of two in time $O(pkn)$. Next, we describe a $2^{O(p\varrho/\varepsilon)}k^{O(p/\varepsilon)}n^2$-time $(1+\varepsilon)$-approximation algorithm for HRC. In particular, it runs in polynomial time when $p = O(1)$ and $\varrho = O(\log(k+n))$. Finally, we show how to find in $O((\frac{n}{\varepsilon} + kn\log n + k^2\log n)(2^{\varrho}k)^{2/\varepsilon})$ time a set L of $O(p\log k)$ strings of length n such that for each string in S there is at least one string in L within distance $(1+\varepsilon)\varrho$, for any constant $0 < \varepsilon < 1$.

Keywords: Hamming distance; p-clustering problem; NP-hardness; Approximation algorithms; Integer programming

* Corresponding author.
E-mail addresses: leszek@csc.liv.ac.uk (L. Gąsieniec), jesper.jansson@cs.lth.se (J. Jansson), andrzej.lingas@cs.lth.se (A. Lingas).

1570-8667/$ – see front matter
doi:10.1016/S1570-8667(03)00079-0

1. Introduction

Let $\mathbb{Z}_2^n$ be the set of all strings of length n over the alphabet $\{0, 1\}$. For any $\alpha \in \mathbb{Z}_2^n$, we use the notation $\alpha[i]$ to refer to the symbol placed at the ith position of α, where $i \in \{1, \ldots, n\}$. The *Hamming distance* between $\alpha_1, \alpha_2 \in \mathbb{Z}_2^n$ is defined as the number of positions in which the strings differ, and is denoted by $d_H(\alpha_1, \alpha_2)$.

The *Hamming radius p-clustering problem*[1] (HRC) is: Given a set S of k binary strings $\alpha_i \in \mathbb{Z}_2^n$, where $i = 1, \ldots, k$, and a positive integer p, find p strings $\beta_j \in \mathbb{Z}_2^n$, where $j = 1, \ldots, p$, minimizing the value

$$\max_{\alpha_i \in S} \min_{1 \leqslant j \leqslant p} d_H(\alpha_i, \beta_j). \tag{1}$$

Such a set of optimal β_j's is called a *p-center set of S*. (Note that an instance of HRC can have several p-center sets.) The corresponding value of (1) is called the *p-radius of S*, and is written as ϱ.

The *Hamming diameter p-clustering problem* (HDC) is defined on the same set of instances as HRC, and is stated as follows: Partition S into p disjoint subsets $S_1, \ldots, S_p$ (called *p-clusters of S*) so that the value of

$$\max_{1 \leqslant q \leqslant p} \max_{\alpha_i, \alpha_j \in S_q} d_H(\alpha_i, \alpha_j) \tag{2}$$

is minimized. The minimum value of (2) is called the *p-diameter of S*, and is referred to as d.

One can immediately generalize HRC and HDC by considering a larger finite size alphabet instead of $\{0, 1\}$, making the problem more amenable to biological applications. However, as long as the distance between two different characters is measured as one, such a generalization involves only trivial generalizations of our approximation methods. Therefore, we only consider the original binary versions of HRC and HDC throughout this paper.

1.1. Previous results

In [3], Frances and Litman showed that the decision version of the Hamming radius 1-clustering problem (1-HRC) is NP-complete. Motivated by the intractability of 1-HRC and its applications in computational biology, coding theory, and data compression, two groups of authors recently provided several close approximation algorithms [5,12]. This was followed by a polynomial-time approximation scheme (PTAS) for 1-HRC [13]. As for the more general HRC and HDC, one can merely find work on the related graph or geometric p-center, p-supplier, and p-clustering problems in the literature [2,8–10,15]. In the case of an undirected complete graph with edge weights satisfying the triangle inequality, all of the three problems mentioned above are known to admit 2-approximation or 3-approximation polynomial-time algorithms, but none of them are approximable within $2 - \varepsilon$ for any $\varepsilon > 0$ in polynomial time unless P = NP [8–10]. This contrasts with the case

[1] The corresponding graph problem is often termed the *p-center* problem in the literature [8].

$p = O(1)$ in which, e.g., the graph p-center and p-supplier problems can be trivially and exactly solved in $n^{O(p)}$ time. HRC does not seem easier than these graph problems. Since HRC is NP-complete even for $p = 1$, optimal or nearly optimal center solutions to it may have to be searched for in $\mathbb{Z}_2^n$, whose size can be exponential in the input size. Our results indicate that in the general case, HRC as well as HDC are equally hard to approximate in polynomial time as the p-center or p-clustering graph problems are.

1.2. Motivation

Clustering is used to solve classification problems in which the elements of a specified set have to be divided into classes so that all members of a class are similar to each other in some sense. HRC and HDC are equally fundamental problems within strings algorithms as the corresponding graph and geometric center and clustering problems are within graph algorithms or computational geometry respectively [2,8–10,15]. They have potential applications in computational biology and pattern matching.

For example, when classifying biomolecular sequences, consensus representatives are useful. The around 100000 different proteins in humans can be divided into 1000 (or less) protein families, which makes it easier for researchers to understand their structures and biological functions [7]. A lot of information about a newly discovered protein may be deduced by establishing which family it belongs to. During identification, it is more efficient to try to align the new protein to representatives for various families than to individual family members. Conversely, given a set S of k related sequences, one way to find other similar sequences is by computing p representatives (where $p \ll k$) for S and then using the representatives to probe a genome database. The representatives should resemble all sequences in S, and must be chosen carefully. For instance, when $p = 1$, the sequence s that minimizes the sum of all pairwise distances between s and elements in S is biased towards sequences that occur frequently, but using a 1-center as representative will avoid this problem.[2] For $p > 1$, the representatives can be the members in the p-center set or simply p sequences, each from a different p-cluster.

In pattern matching applications, the number of classes p can be large; a system for Chinese character recognition, for example, would need to be able to discriminate between thousands of characters.

1.3. Organization of the paper

Section 2 demonstrates that while the p-diameter of a set of binary strings is not necessarily equal to its p-radius, it is always within a factor of two. Next, Section 3 provides polynomial-time solutions for restricted cases of HRC and HDC based on integer programming, exhaustive search, and breadth-first search. In Section 4, we prove the NP-hardness of approximating HRC and HDC within any constant factor smaller than two. In the same section, we also prove that another type of approximation for HDC in terms of the number

[2] Depending on the application, the difference between strings is sometimes measured in terms of edit distance, which also takes insertions and deletions into account, rather than Hamming distance, which just considers substitutions.

of clusters is NP-hard, and that solving HDC exactly is an NP-complete problem already for $p = 3$. Section 5 presents three approximation algorithms for HRC and HDR: a two-approximation algorithm for HRC and HDC based on Gonzalez' furthest-point clustering method [6], an approximation scheme, i.e., a $(1 + \varepsilon)$-approximation algorithm for HRC, and a $(1 + \varepsilon)$-approximation algorithm for HRC using a moderately larger number of approximate centers.

2. Preliminaries

HRC and HRC are defined for the same set of instances, but the p-radius ϱ and the p-diameter d of a set of binary strings are different in general, as the following example illustrates.

Example 2.1. Consider the instance $S = \{00010000,\ 00100000,\ 01000000,\ 10000000,\ 11110000,\ 11111111\}$ with $p = 2$.

An optimal solution to HRC is $\{\beta_1 = 00000000,\ \beta_2 = 11110101\}$ with $\varrho = 2$.

On the other hand, an optimal solution to HDC is $\{S_1 = \{00010000,\ 00100000,\ 01000000,\ 10000000,\ 11110000\}, S_2 = \{11111111\}\}$ with d equal to 3.

Let (S, p) be an instance of HRC/HDC. A p-center set $\{\beta_1, \ldots, \beta_p\}$ of S with p-radius ϱ induces an approximate p-cluster set $\{\widetilde{S}_1, \ldots, \widetilde{S}_p\}$ of S with diameter $\tilde{d}$ (for $i = 1, \ldots, k$, if β_q is a center string that is closest to α_i and has the lowest possible index then let $\alpha_i \in \widetilde{S}_q$). Analogously, a p-cluster set $\{S_1, \ldots, S_p\}$ of S with p-diameter d induces an approximate p-center set $\{\tilde{\beta}_1, \ldots, \tilde{\beta}_p\}$ of S with radius $\tilde{\varrho}$ (for $q = 1, \ldots, p$, let $\{\tilde{\beta}_q\}$ be a 1-center set for the set of strings belonging to S_q).

Example 2.2. Let S be the instance in Example 2.1.

The approximate 2-cluster set induced by $\{\beta_1, \beta_2\}$ is $\{\widetilde{S}_1 = \{00010000,\ 00100000,\ 01000000,\ 10000000\}, \widetilde{S}_2 = \{11110000,\ 11111111\}\}$, so the corresponding value of $\tilde{d}$ is 4.

An approximate 2-center set induced by $\{S_1, S_2\}$ is $\{\tilde{\beta}_1 = 01010000, \tilde{\beta}_2 = 11111111\}$, which implies $\tilde{\varrho} = 3$.

The next lemma shows that an approximate solution to HDC induced by an optimal solution to HRC is within a factor of two of optimum, and vice versa. Moreover, it shows that the p-diameter of a set of binary strings is always less than or equal to twice its p-radius.

Lemma 2.3. *Given an instance of HRC/HDC, define $\varrho, \tilde{\varrho}, d$, and $\tilde{d}$ as above. Then*:

(a) $\tilde{\varrho} \leqslant 2\varrho$;
(b) $\tilde{d} \leqslant 2d$;
(c) $\varrho \leqslant d \leqslant 2\varrho$.

Proof. By definition, we have (1) $\varrho \leqslant \tilde{\varrho}$ and (2) $d \leqslant \tilde{d}$. Also, (3) $\tilde{\varrho} \leqslant d$ because setting $\tilde{\beta}_q$ to an arbitrary string in S_q for each $q \in \{1, \ldots, p\}$ gives an approximate p-center set with radius less than or equal to d. Next, since the Hamming distance obeys the triangle inequality [11, p. 424], the distance between two strings α_i, α_j that end up in the same $\widetilde{S}_q$ must be less than or equal to $d_H(\alpha_i, \beta_q) + d_H(\beta_q, \alpha_j) \leqslant 2\varrho$, so it holds that (4) $\tilde{d} \leqslant 2\varrho$.

Now, (a) follows from (3), (2), and (4); (b) follows from (4), (1), and (3). Finally, (c) follows from (1), (3), (2), and (4). □

3. Polynomial-time solutions for restricted cases

The Hamming radius 1-clustering problem (1-HRC) is equivalent to a special case of the integer programming problem. Any given instance $(\alpha_1, \ldots, \alpha_k)$ of 1-HRC, where $\alpha_i \in \mathbb{Z}_2^n$ for $1 \leqslant i \leqslant k$, can be expressed as a system of k linear inequalities as follows.

For $i = 1, \ldots, k$, let the ith inequality be

$$\sum_{\substack{\alpha_i[m]=0 \\ 1 \leqslant m \leqslant n}} x_m + \sum_{\substack{\alpha_i[m]=0 \\ 1 \leqslant m \leqslant n}} (1 - x_m) \leqslant \varrho$$

and let $X = (x_1, \ldots, x_n) \in \mathbb{Z}_2^n$ be a vector of 0–1-variables representing a 1-center of $\{\alpha_1, \ldots, \alpha_k\}$. ϱ is an integer variable corresponding to the 1-radius. The left-hand side of inequality i equals the Hamming distance between α_i and X. (For each position m, if $\alpha_i[m] = 0$ then the sum is incremented by one if and only if $x_m = 1$, and conversely, if $\alpha_i[m] = 1$ then the sum is incremented by one if and only if $x_m = 0$.) The constraint "$\leqslant \varrho$" ensures that $d_H(\alpha_i, X)$ is smaller than or equal to the radius.

The above system of inequalities can be transformed into the form $Ax \leqslant b(\varrho)$, where A is a $(k \times n)$-matrix with every entry belonging to the set $\{-1, 1\}$, x is a $(n \times 1)$-vector of variables belonging to $\mathbb{Z}_2$, and $b(\varrho)$ is a $(k \times 1)$-vector that depends on ϱ. The scalar product of any prefix of any row in A with a 0–1-vector of the same length is neither less than $-n$ nor greater than n. Therefore, we can solve the transformed system of k inequalities by a dynamic programming procedure, proceeding in stages [14]. In stage l, we compute the set W_l of all $(k \times 1)$-vectors that can be expressed as $\sum_{m=1}^{l} c_m z_m$, where c_m is the mth column of A and $z_m \in \mathbb{Z}_2$. Since the cardinality of W_l cannot be larger than $(2n + 1)^k$ and there are n stages, this procedure takes a total of $O((2n)^k \cdot n)$ time. Next, for each v in W_n, solve $v \leqslant b(\varrho)$ in $O(k)$ time to identify a v^* which yields the smallest possible value of ϱ. A 1-center β for the given instance is then obtained by setting $\beta[m] = z_m^*$ for $1 \leqslant m \leqslant n$, where

$$v^* = \sum_{m=1}^{n} c_m z_m^*.$$

The whole algorithm uses

$$O\big((2n)^k \cdot n + (2n)^k \cdot k + n\big) = n^{O(k)}$$

time.

Lemma 3.1. 1-*HRC for instances with k strings of length n is solvable in $n^{O(k)}$ time.*

If k is constant then any instance of the Hamming radius p-clustering problem can be transformed into a polynomial number of instances of 1-HRC. Let $(\alpha_1, \ldots, \alpha_k, p)$ be a given instance of HRC, where $\alpha_i \in \mathbb{Z}_2^n$ for $1 \leqslant i \leqslant k$ and $p \in \mathbb{N}$. For each of the $O(p^k)$ ways to partition the k strings into p subsets $\{S_1, \ldots, S_p\}$, construct p instances of 1-HRC such that instance j consists of subset S_j, use the method in Lemma 3.1 to solve each instance, and let the value of this partition equal the maximum of the p resulting 1-radii. As the final solution, return the set of 1-centers in a partition that yields the smallest value.

To prove the correctness of this method, consider an optimal p-center set $\{\beta_1, \ldots, \beta_p\}$. It induces a partition $\{\widetilde{S}_1, \ldots, \widetilde{S}_p\}$ of $\{\alpha_1, \ldots, \alpha_k\}$, where for $1 \leqslant i \leqslant k$, $\alpha_i \in \widetilde{S}_q$ if β_q is the center string with lowest index closest to α_i. Let ϱ be the p-radius. By the definition of a p-center set, $d_H(\alpha_i, \beta_q) \leqslant \varrho$ for all $\alpha_i \in \widetilde{S}_q$. Thus, the distance between an optimal 1-center of $\widetilde{S}_q$ and a string in $\widetilde{S}_q$ cannot be greater than ϱ. All partitions of the input strings, including $\{\widetilde{S}_1, \ldots, \widetilde{S}_p\}$, are tested, so an optimal solution will be found.

The method takes a total of $O(p^k) \cdot O(p) \cdot n^{O(k)} = n^{O(k)}$ time. We conclude that HRC with $k = O(1)$ and arbitrary p can be solved exactly in polynomial time.

Theorem 3.2. *HRC for instances with k strings of length n is solvable in $n^{O(k)}$ time.*

On the other hand, if $n = O(\log k)$, exhaustive search gives a $k^{O(p)}$-time solution.

Theorem 3.3. *HRC restricted to instances with k strings of length $O(\log k)$ is solvable in $k^{O(p)}$ time.*

One of the main differences between HDC and HRC is that the former does not involve strings outside the input set S. For this reason, it seems simpler to solve exactly than HRC does.[3] Furthermore, it can be solved by exhaustive search in $O(k^2 n + k^2 p^k)$ time, which immediately yields the following result.

Theorem 3.4. *HDC restricted to instances with $O(\log n)$ strings of length n is solvable in $n^{O(\log p)}$ time.*

More interestingly, the Hamming diameter 2-clustering problem admits the following, rather straightforward polynomial-time solution. Let d be a candidate value for the maximum Hamming cluster diameter in an optimal 2-clustering of the k input strings of length n. Form a graph G with vertices in one-to-one correspondence with the input strings, and connect a pair of vertices by an edge whenever the Hamming distance between the corresponding strings is less than or equal to d. Now, the problem of Hamming diameter 2-clustering for the input strings becomes equivalent to that of partitioning the vertices of G into two cliques. The latter problem in turn reduces to 2-coloring the complement graph.

[3] Paradoxically, as for approximation in terms of the number of clusters, it might be more difficult, as is observed in the next sections.

By breadth-first search, we can find a 2-coloring of the complement graph, if one exists, in $O(k^2)$ time. To find the smallest possible d, we use the procedure just described to test different values of d, generated by a binary search. Calculating all pairwise Hamming distances requires $O(k^2 n)$ time, but this can be done before starting the search for d. Hence, we obtain the following result.

Theorem 3.5. *For $p = 2$, HDC is solvable in $O(k^2 n)$ time.*

Note that Theorem 3.5 can be generalized to any metric.

4. NP-hardness of approximating HRC and HDC

By approximating HRC or HDC, we mean providing a polynomial-time algorithm yielding a solution which approximates the p-radius or the p-diameter, respectively. Our results from the first subsection prove the NP-hardness of this type of approximation of HRC and HDC. In the second subsection, we consider another kind of approximation of HDC relaxing the requirement on the number of produced clusters under the condition that their diameter does not exceed the p-diameter; we show that it is NP-hard to approximate the number of clusters within any reasonable factor.

4.1. NP-hardness of approximating the p-radius and p-diameter

To prove the hardness results in this subsection, we use the reduction in [2] from vertex cover for planar graphs of degree at most three to the corresponding p-clustering problem in the plane under the L_1 metric. (The *radius p-clustering problem in the plane under the L_1 metric* is the following: For a finite set S of points in the plane, find a set P of p points in the plane that minimizes $\max_{s \in S} \min_{u \in P} d_1(s, u)$, where d_1 is the L_1 distance. The *diameter p-clustering problem in the plane under the L_1 metric* is defined analogously.) By inspection of the aforementioned reduction, we show that the points in the resulting instance of the p-clustering problem in the plane as well as the points in an approximate p-center can be required to lie on an integer grid of size polynomial in the size of the input planar graph. This gives the following technical strengthening of Theorem 2.1 in [2].

Lemma 4.1. *Let α be a positive constant less than 2. The radius p-clustering and diameter p-clustering problems in the plane under the L_1 metric for a finite set S of points, where the points in S lie on an integer square grid of size polynomial in the cardinality of S and where the approximate solution to the radius version is required to lie on the grid, are NP-hard to approximate within α.*

Proof. The reduction in [2] embeds an instance of vertex cover for planar graphs of degree at most three in the plane by replacing all edges with odd length paths composed of unit length edges. The midpoints of these unit length edges form an instance $\mathcal{I}$ of radius or diameter p-clustering in the plane which admits a solution with p-radius 0.5 or p-diameter 1, respectively, if and only if the embedded graph has a vertex cover with p nodes. The key

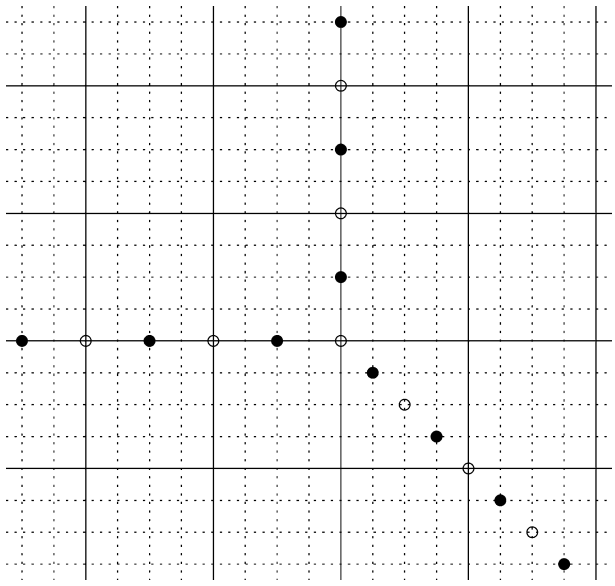

Fig. 1. The L_1 distance between two edge midpoints (shown as filled circles) is 1 if the edges are adjacent, and $\geqslant 2$ otherwise.

observation is that the minimum distance between the midpoints of two non-adjacent edges is at least 2 in case of the L_1 metric (see Fig. 1). It follows that *finding an approximate solution to $\mathcal{I}$ within any factor smaller than* 2 *is as hard as finding an exact solution*, yielding the NP-hardness of approximating radius and diameter p-clustering in the plane under the L_1 metric within any factor smaller than 2. For further details concerning the reduction, see [2] or [8].

Consider the smallest square box B with sides parallel to the x- and y-axes which contains the embedded graph constructed in the reduction. Since the graph can be assumed to be connected, the length of a side of the box is $O(l)$, where l is the number of points in the instance of the radius or diameter clustering problem in the plane. Note that l has to be polynomial in the size n of the original vertex cover instance [2]. We conclude that the box has size polynomial in n.

Form a uniform point grid within B such that the distance between nearest neighbors in the grid is ε, where $0 < \varepsilon \leqslant 0.01$. Move each of the midpoints in $\mathcal{I}$ to its nearest grid point. Such a movement changes the relative distance between two midpoints by at most 2ε. Adding the requirement that an approximate p-center must also lie on the grid can further increase the radius by at most ε. It follows that $\mathcal{I}$ admits a clustering with p-radius 0.5 or p-diameter 1, respectively, if and only if the resulting instance $\mathcal{I}'$ of clustering on the grid admits a solution with p-radius $(0.5 + \varepsilon) + \varepsilon = 0.5 + 2\varepsilon$ or p-diameter $1 + 2\varepsilon$. By the key observation, it also follows that $\mathcal{I}$ has p-radius at least 1 or p-diameter at least 2 if and only if $\mathcal{I}'$ has p-radius $\geqslant 1 - 2\varepsilon$ or p-diameter $\geqslant 2 - 2\varepsilon$. Now, if the p-radius of $\mathcal{I}'$ could be approximated within $2 - 12\varepsilon$ then the p-radius of $\mathcal{I}$ could be computed exactly since $(0.5 + 2\varepsilon) \cdot (2 - 12\varepsilon) < 1 - 2\varepsilon$. Similarly, if the p-diameter of $\mathcal{I}'$ could be

approximated within $2-6\varepsilon$ then the p-diameter of $\mathcal{I}$ could be computed exactly since $(1+2\varepsilon)\cdot(2-6\varepsilon)<2-2\varepsilon$.

Since ε can be selected arbitrarily close to 0 and $\mathcal{I}'$ can be constructed in time polynomial in n for any fixed ε, it is sufficient to transform the grid to an integer grid by rescaling by $1/\varepsilon$ and shifting appropriately in order to obtain the theorem in both cases. □

By embedding the L_1 metric on an integer square grid into the Hamming metric, we obtain our main result in this section.

Theorem 4.2. *HRC and HDC are NP-hard to approximate within any constant factor smaller than two.*

Proof. Let S be a set of points on an integer square grid of size $q(|S|)\times q(|S|)$, where $q(|S|)$ is polynomial in $|S|$. For each $s\in S$, denote the x- and y-coordinates of s by s_x and s_y, respectively. Encode each $s\in S$ by the binary string $e(s)$ of length $2q(|S|)$ composed of s_x consecutive 1's followed by $q(|S|)-s_x$ consecutive 0's, then s_y consecutive 1's, and finally, $q(|S|)-s_y$ consecutive 0's. Note that for any two points s' and s'' in S, their L_1 distance is equal to the Hamming distance between their encodings $e(s')$ and $e(s'')$. This observation immediately yields the theorem thesis for HDC by Lemma 4.1.

Consider an approximate solution $a_1,\ldots,a_p$ to HRC for the strings $e(s)$, $s\in S$. For $i=1,\ldots,p$, transform a_i to a_i' having the form $1^l0^{q(|S|)-l}1^m0^{q(|S|)-m}$ for some $l,m\leqslant q(|S|)$ by moving all the 1's contained in the left half of a_i to the beginning of the left half, and all the 1's in the right half of a_i to the beginning of the right half. The resulting string sequence $a_1',\ldots,a_p'$ is a solution which is at least as good as $a_1,\ldots,a_p$ for the strings $e(s)$, $s\in S$. Also, it can be directly decoded into a sequence of grid points $g_1,\ldots,g_p$ such that $a_i'=e(g_i)$ for $i=1,\ldots,p$. Putting everything together, we obtain the theorem thesis for HRC by Lemma 4.1. □

4.2. NP-hardness of approximating HDC in terms of the number of clusters

The *clique partition problem* is: Given an undirected graph G and a natural number p, partition the set of vertices of G into pairwise disjoint subsets $V_1,\ldots,V_p$ such that for $j=1,\ldots,p$, the subgraph of G induced by V_j is a clique. Clearly, this problem is equivalent to coloring the complement graph with p colors. It follows from known inapproximability results for graph coloring [1] that for any $\varepsilon>0$, the problem of finding an approximate solution to the clique partition problem consisting of at most $pn^{1/7-\varepsilon}$ cliques, where n is the number of vertices in the instance graph G, is NP-hard.

By a reduction from the clique partition problem to HDC, we obtain:

Theorem 4.3. *For any $\varepsilon>0$, the problem of finding a partition of a set of k binary strings of length $O(k^2)$ into at most $pk^{1/7-\varepsilon}$ disjoint clusters such that each cluster has Hamming diameter not exceeding the p-diameter is NP-hard.*

Proof. Let G be an undirected graph with n vertices. Construct an undirected graph G' with $2n$ vertices by augmenting G with n new vertices and then, for every vertex v appear-

ing in G, adding edges between v and new vertices until v gets degree n in G'. Enumerate the edges of G' from 1 to m, where $m = O(n^2)$. For every vertex v in G, form a string $s(v)$ of length m such that there is a 1 on the ith position in $s(v)$ if and only if the ith edge of G' is incident to v. Note that for any pair of vertices v_1, v_2 in G, the Hamming distance between $s(v_1)$ and $s(v_2)$ is $2n-2$ if they are adjacent, otherwise it is $2n$. Therefore, any clique partition of G into p cliques yields a p-clustering of the resulting strings of maximum Hamming diameter less than or equal to $2n-2$, and conversely, any q-clustering of the resulting strings of maximum Hamming diameter less than or equal to $2n-2$ trivially yields a partition of G into q cliques. Hence, by the inapproximability result cited above, we obtain our result. □

Since the clique partition problem is NP-complete for all fixed $p \geqslant 3$ (see [4]), the reduction in the proof of Theorem 4.3 together with the fact that HDC belongs to NP imply the following:

Corollary 4.4. *HDC is NP-complete for all fixed* $p \geqslant 3$.

As for the corresponding approximation problem for HRC (i.e., producing a larger set of approximate centers such that each input string is within the p-radius from at least one of the centers), we doubt whether it is equally hard to approximate. At least, if we weaken the requirement of being within the p-radius by a multiplicative factor of $1+\varepsilon$, then this problem admits a logarithmic approximation in polynomial time, as is shown at the end of the next section.

5. Approximation algorithms for HRC and HDC

In this section, we first see how an approximation factor of two for HRC and HDC can be achieved. Next, we provide an approximation scheme for HRC running in polynomial time when $p = O(1)$ and $\varrho = O(\log(k+n))$. Finally, we give a relaxed type of arbitrarily close approximation of ϱ due to a moderate increase in the number of clusters which runs in polynomial time whenever $\varrho = O(\log(k+n))$.

5.1. A 2-approximation algorithm for HRC and HDC

To obtain an approximation factor of two, we adapt Gonzalez' farthest-point clustering algorithm [6] to HRC and HDC respectively as follows:

Algorithm A.

STEP 1. Set P^* to $\{\alpha_i\}$, where α_i is an arbitrary string in S.

STEP 2. For $l = 2, \ldots, p$: augment P^* by a string in S that maximizes the minimum distance to P^*, i.e., that is as far away as possible from the strings already in P^*.

STEP 3 (HRC). Return P^*.

STEP 3 (HDC). Assign each string in S to a closest member in P^* and return the resulting clusters.

As mentioned in the proof of Lemma 2.3, the Hamming distance obeys the triangle inequality. Therefore, by Theorem 8.14 in [8], Algorithm A yields an approximate solution to either HRC or HDC that is always within a factor of two of the optimum. We can implement this algorithm by updating the Hamming distance of each string outside P^* to the nearest string in P^* after each augmentation of P^*. To update and then compute a string in S furthermost from P^* takes $O(kn)$ time in each iteration. Hence, we obtain the following theorem.

Theorem 5.1. *An approximate solution to either HRC or HDC that is always within a factor of two of the optimum can be found in* $O(pkn)$ *time.*

5.2. An approximation scheme for HRC

In this subsection we present a $2^{O(p\varrho/\varepsilon)}k^{O(p/\varepsilon)}n^2$-time $(1+\varepsilon)$-approximation algorithm for HRC. Our scheme is partly based on the idea used in the PTAS for 1-HRC in [13].

Algorithm B.

STEP 1. Set $\mathcal{C}$ to an empty subset of $\mathbb{Z}_2^n$. For each subset R of S having exactly r strings, compute the set Q consisting of all positions m, $1 \leqslant m \leqslant n$, on which all strings in R contain the same symbol. Set P to $\{1, 2, \ldots, n\} \setminus Q$. For every possible $f : P \to \{0,\ 1\}$, let q_f be the string in $\mathbb{Z}_2^n$ which agrees with the strings in R on the positions in Q and contains $f(j)$ in each position $j \in P$. Augment $\mathcal{C}$ by q_f.

STEP 2. Let $\mathcal{C}^p$ be the family of all subsets of the set $\mathcal{C}$ of size p. Test all sets in $\mathcal{C}^p$ and return the $P^* \in \mathcal{C}^p$ that minimizes $\max_{1\leqslant i\leqslant k}\ \min_{c\in P^*}\ d_H(\alpha_i, c)$.

The next lemma can be proved analogously as Lemma 11 in [13] (the key lemma for the PTAS for the Hamming radius 1-clustering problem) is proved in case of a logarithmic or smaller sized radius.

Lemma 5.2. *For any subset* U *of* S*, there is a* c *in* $\mathcal{C}$ *such that*

$$\max_{\alpha\in U} d_H(\alpha, c) \leqslant \left(1 + \frac{1}{2r-1}\right) \min_{\beta\in\mathbb{Z}_2^n} \max_{\alpha\in U} d_H(\alpha, \beta).$$

Theorem 5.3. *Algorithm* B *constructs a solution to HRC with approximation factor* $1 + \frac{1}{2r-1}$ *in* $O(2^{pr\varrho+1}k^{pr+1}n^2)$ *time.*

Proof. To prove the correctness and the approximation factor of Algorithm B, consider an optimal p-center for S, say $\{\beta_1, \ldots, \beta_p\}$. Partition S into subsets U_1 through U_p such that for $1 \leqslant j \leqslant p$ and $\alpha \in U_j$, β_j has minimum Hamming distance to α among $\beta_1, \ldots, \beta_p$. By Lemma 5.2, the set $\mathcal{C}^p$ constructed in STEP 2 contains $\{\beta_1^*, \ldots, \beta_p^*\}$ such that for $1 \leqslant j \leqslant p$ and any $\alpha \in U_j$, the Hamming distance between α and β_j^* is at most $1 + \frac{1}{2r-1}$ times the radius of U_j. Thus, Algorithm B yields a solution within $1 + \frac{1}{2r-1}$ of the optimum.

To derive the upper bound on the running time of Algorithm B, first observe that each of the sets P has size at most $r\varrho$ and that a string q_f can be constructed in $O(nr)$ time. Hence,

the size of the set $\mathcal{C}$ does not exceed $2^{r\varrho}k^r$, and $\mathcal{C}$ can be constructed in $O(r2^{r\varrho}k^r n)$ time. Consequently, $\mathcal{C}^p$ is of size at most $k^{rp}2^{pr\varrho}$ and its construction from $\mathcal{C}$ takes $O(2^{pr\varrho}k^{pr}n)$ time. All that remains is to note that the test of each p-tuple in $\mathcal{C}^p$ can be performed in $O(kn)$ time. □

Note that the running time of Algorithm B is polynomial in n and k as long as p and r are constant and $\varrho = O(\log(k+n))$.

Corollary 5.4. *Algorithm* B *yields a polynomial-time approximation scheme for the Hamming radius* $O(1)$*-clustering problem restricted to instances with the* p*-radius in* $O(\log(k+n))$.

5.3. A relaxed type of approximation for HRC

In this subsection, we consider twofold approximation for HRC allowing for producing more than p approximate centers and slightly exceeding the p-radius.

For each c in $\mathcal{C}$ (see Algorithm B), let $S(c)$ be the set of all strings in S within distance $(1+\frac{1}{2r-1})\varrho$ of c. By Lemma 5.2, there is a set consisting of p such sets, covering all of S. If ϱ is known, we run the classical greedy heuristic for minimum set cover (see [8]) on the instance $(S,\ \{S(c) \mid c \in \mathcal{C}\})$ to find a set of $O(p\log k)$ sets covering S. Otherwise, we perform a binary search for the smallest possible value of $\varrho \in \{0,1,\ldots,n\}$ in the definition of the sets $S(c)$ by running the aforementioned heuristic $O(\log n)$ times and each time testing whether or not the resulting cover of S has size $O(p\log k)$. Recall that $|\mathcal{C}| \leqslant 2^{r\varrho}k^r$ and that $\mathcal{C}$ can be constructed in $O(r2^{r\varrho}k^r n)$ time. The instance of set cover corresponding to a given value of ϱ can be constructed in $O(|\mathcal{C}|kn)$ time; the greedy heuristic can be implemented to run in $O(|\mathcal{C}|k^2)$ time. By choosing r so that $\frac{1+\varepsilon}{2\varepsilon} < r < \frac{2}{\varepsilon}$, we obtain the following result.

Theorem 5.5. *For any constant* $0 < \varepsilon < 1$*, we can construct a set* L *of* $O(p\log k)$ *strings of length* n *in* $O((\frac{n}{\varepsilon} + kn\log n + k^2\log n)(2^{\varrho}k)^{2/\varepsilon})$ *time such that for each of the* k *strings in* S *there is at least one string in* L *within distance* $(1+\varepsilon)$ *of the* p*-radius.*

The time bound in Theorem 5.5 is polynomial in n and k as long as $\varrho = O(\log(k+n))$.

6. Conclusions

We have shown not only that two is the best approximation factor for HRC and HDC achievable in polynomial time unless P = NP, but also that it is possible to provide exact solutions or much better approximation solutions to HRC or HDC in several special or relaxed cases. It seems that there are plenty of interesting open problems in the latter direction. For example, is it possible to design very close and efficient approximation algorithms for protein data (see Section 1.2) taking into account the specific distribution of the input?

References

[1] M. Bellare, O. Goldreich, M. Sudan, Free bits, PCPs, and non-approximability—towards tight results, SIAM J. Comput. 27 (3) (1998) 804–915.
[2] T. Feder, D. Greene, Optimal algorithms for approximate clustering, in: Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC'88), 1988, pp. 434–444.
[3] M. Frances, A. Litman, On covering problems of codes, Theory of Computing Systems 30 (1997) 113–119.
[4] M. Garey, D. Johnson, Computers and Intractability—A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[5] L. Gąsieniec, J. Jansson, A. Lingas, Efficient approximation algorithms for the Hamming center problem, in: Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99), 1999, pp. S905–S906.
[6] T. Gonzalez, Clustering to minimize the maximum intercluster distance, Theoret. Comput. Sci. 38 (1985) 293–306.
[7] D. Gusfield, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge Univ. Press, Cambridge, 1997.
[8] D.S. Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS Publishing Company, Boston, 1997.
[9] D.S. Hochbaum, D.B. Shmoys, A best possible heuristic for the k-center problem, Math. Oper. Res. 10 (2) (1985) 180–184.
[10] D.S. Hochbaum, D.B. Shmoys, A unified approach to approximation algorithms for Bottleneck problems, J. Assoc. Comput. Mach. 33 (3) (1986) 533–550.
[11] B. Kolman, R. Busby, S. Ross, Discrete Mathematical Structures, third ed., Prentice Hall, Englewood Cliffs, NJ, 1996.
[12] J.K. Lanctot, M. Li, B. Ma, S. Wang, L. Zhang, Distinguishing string selection problems, in: Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'99), 1999, pp. 633–642.
[13] M. Li, B. Ma, L. Wang, Finding similar regions in many strings, in: Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC'99), 1999, pp. 473–482.
[14] C. Papadimitriou, On the complexity of integer programming, J. ACM 28 (4) (1981) 765–768.
[15] S. Vishwanathan, An $O(\log^* n)$ approximation algorithm for the asymmetric p-center problem, in: Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'96), 1996, pp. 1–5.