Approximation Algorithms for Hamming Clustering Problems

Leszek Gąsieniec[1], Jesper Jansson[2], and Andrzej Lingas[2]

[1] Dept. of Computer Science, University of Liverpool
Peach Street, L69 7ZF, UK
leszek@csc.liv.ac.uk
[2] Dept. of Computer Science, Lund University Box 118, 221 00 Lund, Sweden
{Jesper.Jansson,Andrzej.Lingas}@cs.lth.se

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*.
First, we provide an integer programming formulation of HRC which yields exact solutions in polynomial time whenever k and p are constant. We also observe that HDC admits straightforward polynomial-time solutions when $k = O(\log n)$ or $p = 2$. 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 $\epsilon > 0$ it is NP-hard to split S into at most $pk^{1/7-\epsilon}$ clusters whose Hamming diameter doesn't exceed the p-diameter. Furthermore, we note that by adapting Gonzalez' farthest-point clustering algorithm [6], 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 $0 < \varepsilon < 1$.

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 $i \in \{1,..,n\}$. 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 $d(\alpha_1, \alpha_2)$.

R. Giancarlo and D. Sankoff (Eds.): CPM 2000, LNCS 1848, pp. 108–118, 2000.

The *Hamming radius p-clustering* problem[1] (HRC) is stated as follows: Given a set S of k binary strings $\alpha_i \in \mathbb{Z}_2^n$, where $i = 1, .., k$, and a positive integer p, find p strings $\beta_j \in \mathbb{Z}_2^n$, where $j = 1, .., p$, minimizing the value $\varrho = \max_{1 \le i \le k} \min_{1 \le j \le p} d(\alpha_i, \beta_j)$. Such a set of β_j's is called a *p-center set of S*, and the corresponding value of ϱ is called the *p-radius of S*. Note that an instance of HRC can have several p-center sets.

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 $S_1, .., S_p$ (called *p-clusters of S*) so that the value of $\max_{1 \le q \le p} \max_{\alpha_i, \alpha_j \in S_q} d(\alpha_i, \alpha_j)$ is minimized. This value is called the *p-diameter of S*.

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.

In [4], 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 [3,8,9,10,15]. In the undirected complete graph case, with edge weights satisfying the triangle inequality, all of the three aforementioned problems 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 [8,9,10]. This contrasts with the $p = O(1)$ case when, e.g., the graph p-center and p-supplier problems can be trivially and exactly solved in $n^{O(p)}$ time. HRC doesn't seem easier than these graph problems. Optimal or nearly optimal center solutions to it have to be searched in $\mathbb{Z}_2^n$ whose size might be exponential in the input size. For this reason, HRC is NP-complete already for $p = 1$. 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.

1.1 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

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

center and clustering problems are within graph algorithms or computational geometry respectively [3,8,9,10,15]. 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 $p << k$) 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.

1.2 Organization of the Paper

Section 2 provides polynomial-time solutions for restricted cases of HRC and HDC based on integer programming, exhaustive search, and breadth-first search. In Section 3, 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 of clusters is NP-hard. Section 4 presents three approximations 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 approximative centers.

2 Polynomial-Time Solutions for Restricted Cases

The Hamming radius p-clustering problem is equivalent to a special case of the integer programming problem. A given instance $(\alpha_1, .., \alpha_k, p, \varrho)$ of the decision version of HRC, where $\alpha_i \in \mathbb{Z}_2^n$ for $1 \leq i \leq k$, and $p, \varrho \in \mathbb{N}$, can be expressed as a system of $k \cdot p$ linear inequalities.

[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.

We use two matrices X and Y of 0-1-variables. The rows of X correspond directly to the p strings that constitute a p-center for the supplied instance, and Y is used to make sure that each α_i is within distance ϱ of at least one of the centers.

Let X be a $p \times n$-matrix of variables $x_{jm} \in \mathbb{Z}_2$, where $1 \leq j \leq p$ and $1 \leq m \leq n$. The value of x_{jm} determines the value of the m-th position of the j-th center. Let Y be a $k \times p$-matrix of variables $y_{ij} \in \mathbb{Z}_2$, where $1 \leq i \leq k$ and $1 \leq j \leq p$. $y_{ij} = 1$ only if row j of X is a center string that is closest to α_i, so that for each $i = 1, .., k$, we have $\sum_{j=1}^{p} y_{ij} = 1$. Next, for each $i = 1, .., k$ and $j = 1, .., p$, we have the inequality

$$\sum_{\substack{\alpha_i[m]=0 \\ 1 \leq m \leq n}} x_{jm} + \sum_{\substack{\alpha_i[m]=1 \\ 1 \leq m \leq n}} (1 - x_{jm}) \leq \varrho + (1 - y_{ij}) \cdot D$$

where $D = \max_{1 \leq j \leq k} (\max_{1 \leq i \leq k} d(\alpha_i, \alpha_j))$.

The above system of inequalities can be transformed to the form $Ax \leq b$, where A is a $(kp) \times (np + kp)$ integer matrix, x is a variable vector over $\mathbb{Z}_2^{np+kp}$, and b is a vector in $\mathbb{Z}_2^{kp}$. Note that the scalar product of any prefix of any row of A with a $0-1$-vector of the same length is neither less than $-n$ nor greater than $n + D$. In particular, when $p = 1$, such a product has its absolute value simply bounded by n. Now, we can solve the transformed system of kp inequalities by a well-known dynamic programming procedure [14], proceeding in stages. At the jth stage, we compute the set S_j of all vectors that can be expressed as $\sum_{l=1}^{j} c_l z_l$, where c_l is the lth column of A and $z_l \in \mathbb{Z}_2$. Since the S_j cannot be larger than $(2n + D + 1)^{kp}$ (or $(2n + 1)^k$ if $p = 1$), the whole procedure for a fixed ϱ takes $O((2n + D)^{kp} 2^{kp}(np + kp))$ time (or $O(n^k 2^{2k}(n + k))$ if $p = 1$). Hence, by using binary search to find the smallest possible ϱ, we conclude that HRC for $k = O(1)$ and $p = O(1)$ can be solved in polynomial time.

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

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

Theorem 2. *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 doesn't involve strings outside the input set S. For this reason it seems simpler to solve exactly than HRC does [3]. For example, it has a simpler integer programming formulation involving only a single matrix of indicator variables. Furthermore, it can be solved by exhaustive search in $O(k^2 n + k^2 p^k)$ time, which immediately yields the following result.

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

Theorem 3. *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. 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 4. *For $p = 2$, HDC is solvable in $O(k^2 n)$ time.*

Note that Theorem 4 can be generalized to any metric.

3 NP-Hardness of Approximating HRC and HDC

By approximating HRC or HDC, we mean providing a polynomial-time algorithm yielding a p-center set or a p-clustering approximating 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 doesn't exceed the p-diameter; we show that it is NP-hard to approximate the number of clusters within any reasonable factor.

3.1 NP-Hardness of Approximating the p-Radius and p-Diameter

To prove the hardness results in this subsection, we use the reduction described in [3] 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 correspondingly.)

By straightforward inspection of the aforementioned reduction from vertex cover for planar graphs [3] and using, e.g., the planar graph drawing algorithm from [2] in order to embed the input planar graph in the plane, we can ensure

that the points in the resulting instance of the p-clustering problem in the plane as well as p points in an optimal solution lie on an integer grid of size polynomial in the size of the input planar graph and $(\alpha - 1)^{-1}$. This yields the following technical strengthening of Theorem 2.1 from [3].

Lemma 5. *Let α be a positive constant not less than 1. The radius p-clustering and diameter p-clustering problems for a finite set S of points in the plane with the L_1 metric, where the points in S lie on an integer grid of size polynomial in the cardinality of S and $(\alpha - 1)^{-1}$, and where the approximative solution to the radius version is required to lie on the grid, are NP-hard to approximate within α.*

By using the idea of embedding the L_1-metric on a integer square grid into the Hamming one, we obtain our main result in this section.

Theorem 6. *HRC and HDC are NP-hard to approximate within any constant factor smaller than two.*

Proof. Let S be a set of points on integer square grid of size $q(|S|)$ where $q()$ is a polynomial. Encode each grid point s of coordinates s_x and s_y respectively by the $0-1$ string $e(s)$ of length $2q(|S|)$ composed of s_x consecutive 1's followed by $q(|S|) - s_x$ consecutive 0's, next s_y consecutive 1's, and finally, $q(|S|) - s_y$ consecutive 0's. Note that for any two grid points s' and s'' their L_1 distance is equal to the Hamming distance between their encodings $e(s')$ and $e(s'')$. This observation yields immediately the theorem thesis for HDC by Lemma 5.

Consider an approximative solution $a_1, a_2, .., a_p$ to HRC problem for the strings $e(s)$, $s \in S$. For $i = 1, .., p$, we can transform a_i to a'_i having the form of $1^l 0^{q(|S|)-l} 1^m 0^{q(|S|)-m}$ for some $l, m \leq q$ by moving all the 1's in the first half of a_i to the appropriate prefix of a_i and similarly moving the remaining 1's to the appropriate prefix of the second half of a_i and filling the left positions with the left 0's. Observe that the resulting string sequence $a'_1, a'_2, .., a'_p$ yields at least as good solution as $a_1, a_2, .., a_p$ for the strings $e(s)$, $s \in S$ by the special form of the $e(s)$'s. Also, it can be immediately decoded into a sequence of grid points $g_1, g_2, .., g_p$ such that $a'_i = e(g_i)$ for $i = 1, .., p$. Putting everything together, we obtain the theorem thesis for HRC by Lemma 5. □

3.2 NP-Hardness of Approximating HDC in Terms of the Number of Clusters

Consider the following *clique partition problem*: Given an undirected graph G and a natural number p, partition the set of vertices of G into pairwise disjoint subsets $V_1, .., V_p$ such that for $j = 1, .., p$, 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 $\epsilon > 0$, the problem of finding an approximative solution to the clique partition problem consisting of $pn^{1/7-\epsilon}$ cliques, where n is the number of vertices in the instance graph G, is NP-hard. For our purposes, it will be

convenient to assume that the instance graph is *quasi-regular*, by which we mean that it satisfies the following two properties:

1. It contains two distinguished cliques.
2. All vertices outside the two cliques have the same degree, which is not less than that of any vertex in the cliques.

To achieve this, we can augment G with two cliques on n auxiliary vertices each. Next, we connect each original vertex in G of degree q to we equally distribute the new connections in a cyclic fashion so that each vertex of the two cliques receives at most $\lceil (2n^2 - 2m)/2n \rceil$ connections to the original vertices. Let G^* be the resulting graph on $3n$ vertices. Note that all original vertices have degree $2n$ and all vertices in the n-cliques have degree at most $2n$ in G^*. It is clear that if the vertices of G can be partitioned into l cliques then the vertices of G^* can be partitioned into at most $l + 2$ cliques. Conversely, if the vertices of G^* can be partitioned into l cliques then the vertices of G can also be trivially partitioned into at most l cliques. Putting everything together, we obtain the following technical lemma.

Lemma 7. *For any $\epsilon > 0$, the clique partition problem restricted to quasi-regular graphs cannot be approximated (in terms of the number of cliques) within $n^{1/7-\epsilon}$ unless P=NP.*

By a reduction from the clique partition problem for quasi-regular graphs to HDC, we obtain the following result.

Theorem 8. *For any $\epsilon > 0$, the problem of finding a partition of a set of k binary strings of length $O(k^2)$ into at most $pk^{1/7-\epsilon}$ disjoint clusters such that each cluster has Hamming diameter not exceeding the p-diameter is NP-hard.*

Proof. Consider an instance of the restricted clique partition problem consisting of a quasi-regular graph G on k vertices and m edges, and a natural number p. Enumerate the edges of G. For each vertex v of G, form a string $s(v)$ of length m such that there is a 1 on the ith position in $s(v)$ iff the ith edge of G is incident to v. Let d be the maximum vertex degree of G. It follows that each vertex in G outside the two distinguished cliques has degree d. Note that for any pair of vertices v_1, v_2 in G of degree d, the Hamming distance between $s(v_1)$ and $s(v_2)$ is $2d - 2$ if they are adjacent, otherwise it is $2d$. Also, for any pair of vertices v_1, v_2 in the same distinguished clique of G, the Hamming distance between $s(v_1)$ and $s(v_2)$ is at most $2d - 2$. Therefore, any clique p-partition of G yields a p-clustering of the resulting strings of maximum Hamming diameter less than or equal to $2d - 2$. Conversely, any q-clustering of the resulting strings of maximum Hamming diameter less than or equal to $2d - 2$ trivially yields a clique $(q + 2)$-partition of G. Hence, by Lemma 7 we obtain our result. □

As for the corresponding problem for HRC (i.e., producing a larger set of approximative 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 $1 + \epsilon$, then this problem admits a logarithmic approximation in polynomial time, as it is shown at the end of the next section.

4 Approximation Algorithms for HRC and HDC

In this section, we first observe 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 $\varrho = O(\log(k + n))$.

4.1 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 $l = 2, .., p$: 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. □

The Hamming distance obeys the triangle inequality ([11], p. 424). Therefore, by the proof of Theorem 8.14 in [8], Algorithm A yields an approximative 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 9. *An approximative solution to either HRC or HDC that is always within a factor of two of the optimum can be found in $O(pkn)$ time.*

4.2 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, $1 \leq m \leq n$, on which all strings in R contain the same symbol. Set P to $\{1, 2, .., n\} \setminus Q$. 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 $\max\limits_{1 \leq i \leq k} \min\limits_{c \in P^*} d_H(\alpha_i, c)$. □

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

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

Theorem 11. *Algorithm B constructs a p-center with the 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 $\{\beta_1, .., \beta_p\}$. Partition S into subsets U_1 through U_p such that for $1 \leq j \leq p$ and $\alpha \in U_j$, β_j has minimum Hamming distance to α among $\beta_1, .., \beta_p$. By Lemma 10, the set $\mathcal{C}^p$ constructed in STEP 2 contains $\{\beta_1^*, .., \beta_p^*\}$ such that for $1 \leq j \leq p$ 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 $\mathcal{C}$ doesn't 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 is a constant and $\varrho = O(\log(k + n))$.

Corollary 12. *Algorithm B yields a polynomial-time approximation scheme for the Hamming radius $O(1)$-clustering problem restricted to instances with the p-radius in $O(\log(k + n))$.*

4.3 A Relaxed Type of Approximation for HRC

In this subsection, we consider twofold approximation for HRC allowing for producing more than p approximative 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 10, 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 $\varrho \in \{0, 1, .., n\}$ 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 $|\mathcal{C}| \leq 2^r \varrho k^r$ 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 13. *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 13 is polynomial in n and k as long as $\varrho = O(\log(k + n))$.

5 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 1.1) taking into account the specific distribution of the input?

References

1. M. Bellare, O. Goldreich, and M. Sudan. Free Bits, PCPs, and Non-Approximability – Towards Tight Results. *SIAM Journal on Computing* 27(3), 1998, pp. 804–915.
2. M. Chrobak and T.H. Payne. A linear-time algorithm for drawing a planar graph on a grid. *Information Processing Letters* 54, 1995, pp. 241–246.
3. T. Feder and D. Greene. Optimal Algorithms for Approximate Clustering. *Proceedings of the 20th Annual ACM Symposium on Theory of Computing* (STOC'88), 1988, pp. 434–444.
4. M. Frances and A. Litman. On Covering Problems of Codes. *Theory of Computing Systems* 30, 1997, pp. 113–119.

5. L. Gąsieniec, J. Jansson, and A. Lingas. Efficient Approximation Algorithms for the Hamming Center Problem. *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. *Theoretical Computer Science* 38, 1985, pp. 293–306.
7. D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
8. D.S. Hochbaum (editor). *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, Boston, 1997.
9. D.S. Hochbaum and D.B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operational Research* 10(2), 1985, pp. 180–184.
10. D.S. Hochbaum and D.B. Shmoys. A Unified Approach to Approximation Algorithms for Bottleneck Problems. *Journal of the Association for Computing Machinery* 33(3), 1986, pp. 533-550.
11. B. Kolman, R. Busby, and S. Ross. *Discrete Mathematical Structures* [3rd ed.]. Prentice Hall, New Jersey, 1996.
12. J.K. Lanctot, M. Li, B. Ma, S. Wang, and L. Zhang. Distinguishing String Selection Problems. *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA'99), 1999, pp. 633–642.
13. M. Li, B. Ma, and L. Wang, Finding Similar Regions in Many Strings. *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. *Journal of the ACM* 28(4), 1981, pp. 765–768.
15. S. Vishwanathan. An $O(\log^* n)$ Approximation Algorithm for the Asymmetric p-Center Problem. *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA'96), 1996, pp. 1–5.