# Efficient Approximation Algorithms for the Hamming Center Problem

Leszek Gąsieniec [*] Jesper Jansson [†] Andrzej Lingas [†]

**Abstract**

The *Hamming center problem* for a set $S$ of $k$ binary strings, each of length $n$, is to find a binary string $\beta$ of length $n$ that minimizes the maximum Hamming distance between $\beta$ and any string in $S$. Its decision version is known to be NP-complete [2]. We provide several approximation algorithms for the Hamming center problem. Our main result is a randomized $(\frac{4}{3}+\varepsilon)$-approximation algorithm running in polynomial time if the Hamming radius of $S$ is at least superlogarithmic in $k$. Furthermore, we show how to find in polynomial time a set $B$ of $O(\log k)$ strings of length $n$ such that for each string in $S$ there is at least one string in $B$ within Hamming distance not exceeding the radius of $S$.

## 1 Introduction

Let $\mathbb{Z}_2^n$ be the set of all strings of length $n$ over the alphabet $\{0,1\}$. For a string $\alpha \in \mathbb{Z}_2^n$ we use the notation $\alpha[i]$ to refer to the $i$th symbol of $\alpha$ (the symbol placed on the $i$th position of $\alpha$), for $i = 1,..,n$, and we let $\alpha[i..j]$ represent the substring of $\alpha$ starting at position $i$ and ending at position $j$, where $1 \leq i < j \leq n$. The *Hamming distance* between any $\alpha_1, \alpha_2 \in \mathbb{Z}_2^n$ is defined as the number of positions in which the strings differ and is denoted by $d(\alpha_1, \alpha_2)$.

The *Hamming center problem* (HCP) is stated as follows: Given a set $S$ of $k$ binary strings $\alpha_i \in \mathbb{Z}_2^n$, where $i = 1,2,..,k$, find a string $\beta \in \mathbb{Z}_2^n$ such that the value $r = \max_{1\leq i\leq k} d(\beta, \alpha_i)$ is minimized. The string $\beta$ is called a *center* of the set $S$ (most instances of HCP have several centers), and the value $r$ is called its *radius*.

A variant of HCP has been studied by Berman *et al.* in [1], where they look for all maximal blocks within a set of aligned binary strings, having centers with radius of size at most 1. Their study is motivated by the detection of potentially important regions within closely related DNA sequences. They provide a linear time algorithm to their problem. In fact, there exists a straightforward polynomial-time algorithm for HCP as long as the radius is $O(1)$ (see Section 4). Berman *et al.* left the question open whether there exists an efficient algorithm for the case where the size of the radius is arbitrarily large.

In [2], Frances and Litman showed that the decision version of HCP, by them called the Minimum Radius Problem, as well as the equivalent dual Maximum Covering Radius Problem (MCR) are NP-complete.

Motivated by the intractability of HCP, we present several approximation algorithms. Some similar results and many other related results can be found in [4].

## 2 Preliminaries

The following simple heuristic yields a 2-approximation for HCP instantly by the triangle inequality: Set the approximative solution $\beta'$ to $\alpha_l$, where $l$ is chosen arbitrarily from the integers $1,2,..,k$.
Any given instance $(\alpha_1,\alpha_2,..,\alpha_k,r_p)$ of the decision version of HCP, where $\alpha_i \in \mathbb{Z}_2^n$ for $1 \leq i \leq k$ and $r_p \in \mathbb{N}$, can be expressed as the system of $k$ linear inequalities. Let $x = (x_1,x_2,..,x_n) \in \{0,1\}^n$ represent a vector of $0-1$ variables and let the $i$th inequality be

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

A $0-1$-solution $x$ that is compatible with the system of inequalities corresponds directly to a solution $\beta$ for the supplied instance of HCP, as can be seen by setting $\beta[m] = x_m$, for $1 \leq m \leq n$.

By the known methods of solving the integer programming problem in polynomial time in case the number of variables or the number of inequalities is constant, we obtain:

LEMMA 2.1. *HCP restricted to instances with $O(1)$ strings as well as HCP restricted to instances with strings of length $O(1)$ are solvable in polynomial time.*

## 3 Randomized Rounding

By relaxing the integer constraints on $x$ in the 0-1-integer programming formulation of HCP, we get a linear programming problem solvable in polynomial time [3]. Then, we use the following straightforward randomized rounding scheme to obtain an approximative solution $\beta'$ to the original problem: Let $\hat{x}_m$ be the value assigned to $x_m$ in the solution to the linear program. For each $m$, where $1 \leq m \leq n$, set $\beta'[m]$ to 1 with probability $\hat{x}_m$ and to 0 with probability $1-\hat{x}_m$.

The analysis of this method proceeds analogously to the one used in lattice approximation problem (see [3]). We consider each inequality separately, and flip the probability from $p_i$ to $1-p_i$ in case $\alpha_i[m] = 1$. By using two variants of Chernoff's bounds (see [3]) and defining the *generalized distance* between any $z_1, z_2 \in [0,1]^n$ as $\sum_{m=1}^{n} |z_1[m] - z_2[m]|$, we obtain:

LEMMA 3.1. *The maximum Hamming distance between the approximative solution $\beta'$ found by randomized rounding and any string in the input instance of HCP*

[*]Dept. of Computer Science, University of Liverpool, Peach Street, L69 7ZF, UK, leszek@csc.liv.ac.uk

[†]Dept. of Computer Science, Lund University, Box 118, 221 00 Lund, Sweden, { Jesper.Jansson, Andrzej.Lingas } @dna.lth.se

*with radius $r$ is at most $r + O(\sqrt{n \log n})$ with probability $\geq 1 - \frac{k}{n^2}$, and at most $r(1+\varepsilon)$ with probability $\geq 1 - k \cdot 2\exp(-0.38\varepsilon^2 w)$, where $w$ is the minimum generalized distance between a string in the instance and an optimal solution to the relaxed version of the instance.*

COROLLARY 3.1. *If the radius of the instance is larger than $q\sqrt{n \log n}$ then the maximum distance between $\beta'$ and any string in the instance is at most $r + O(q^{-1})$ with probability $\geq 1 - \frac{k}{n^2}$. If the minimum generalized distance $w$ between a string in the instance and an optimal relaxed solution is at least $\frac{\ln(4k)}{0.38\varepsilon^2}$ then the maximum distance between $\beta'$ and any string in the instance is at most $(1+\varepsilon)r$ with probability $\geq \frac{1}{2}$.*

## 4 $(\frac{4}{3}+\varepsilon)$-Approximation Algorithm

Here, we present a randomized approximation algorithm for HCP. For $\varepsilon > 0$, let $\beta'$ be the approximative solution delivered by our algorithm and let $r' = \max_{1 \leq i \leq k} d(\beta', \alpha_i)$. We prove that $r' \leq (\frac{4}{3}+\varepsilon)r$ with probability $\geq \frac{1}{2}$. Let $f$ be *the diameter* of the input set $S$, i.e., $f = \max_{1 \leq i \leq k} (\max_{1 \leq j \leq k} d(\alpha_i, \alpha_j))$. We distinguish two cases, depending on the value of $f$: (1) $f < \frac{2\ln(4k)}{0.38\varepsilon^3}$, and (2) $f \geq \frac{2\ln(4k)}{0.38\varepsilon^3}$

The algorithm begins by computing $f$ and $\frac{2\ln(4k)}{0.38\varepsilon^3}$ for the given instance and the specified value of $\varepsilon$, whereupon it branches to either Case 1 or Case 2.

Case 1: $f = O(1)$. The algorithm finds an exact solution by trying all of the $\sum_{j=0}^{f} \binom{n}{j} = O(n^f)$ strings in $\mathbb{Z}_2^n$ within Hamming distance $f$ of $\alpha_l$, where $l \in \{1, 2, .., k\}$. The total time required is $O(kn \cdot n^f)$.

Case 2: (2a) $f \leq \frac{4}{3}r$, and (2b) $f > \frac{4}{3}r$. Since we don't know beforehand which subcase holds for a given instance, the algorithm runs both of the procedures below and chooses the better one of the two solutions.

Subcase 2a: $f \leq \frac{4}{3}r$. Set $\beta' = \alpha_l$, where $l \in \{1, 2, .., k\}$. Since $r' \leq f \leq \frac{4}{3}r$, so we are done.

Subcase 2b: $f > \frac{4}{3}r$. Let $d(\alpha_1, \alpha_k) = f$. Normalize all strings by flipping $\alpha_i[m]$ to $1 - \alpha_i[m]$ whenever $\alpha_1[m] = 1$, for all $m = 1, .., n$. Then, let some permutation $\sigma$ act on the columns so that the $f$ positions of $\alpha_k$ that contain 1s are located at $\alpha_k[1], .., \alpha_k[f]$. None of the transformations changes the Hamming distances.

Apply the idea of the simple heuristic from Section 2 on the first $f$ positions of $\alpha_i$, $i = 1, .., k$. Set $\gamma[1..f]$ to that $\alpha_i[1..f]$ which minimizes the maximum distance to the remaining ones, and set the last $n - f$ positions of $\gamma$ to 000...0. Next, use the randomized rounding heuristic described in Section 3 on the first $f$ positions of $\alpha_i$, $i = 1, .., k$, to obtain $\mu[1..f]$, and let $\mu[(f+1)..n] = 000...0$. Set $\beta'$ to that of the strings $\gamma, \mu$ which has the smallest maximum distance to a string $\alpha_i$, $i = 1, .., k$.

LEMMA 4.1. *If $f \geq \frac{2\ln(4k)}{0.38\varepsilon^3}$ then the maximum distance between $\beta'[1..f]$ and a string $\alpha_i[1..f]$, $i = 1, .., k$, is within $1+\varepsilon$ of the radius of $\alpha_i[1..f]$, $i = 1, .., k$, with probability at least $\frac{1}{2}$.*

*Proof.* If the minimum generalized distance between $\alpha_i[1..f]$, $i = 1, .., k$, and an optimal relaxed solution for $\alpha_i[1..f]$, $i = 1, .., k$, is at most $\frac{\varepsilon}{2}f$ then we are done since the radius of $\alpha_i[1..f]$, $i = 1, .., k$, is at least $\frac{f}{2}$. Otherwise, the lemma follows from the second part of Corollary 3.1 for $w = \frac{\varepsilon}{2}f$.□

LEMMA 4.2. *$\beta[(f+1)..n]$ contains less than $\frac{1}{3}r$ ones.*

*Proof.* Suppose $\beta[(f+1)..n]$ contains $\geq \frac{1}{3}r$ ones. Since $d(\beta, \alpha_1) \leq r$, $\beta[1..f]$ must contain $\leq \frac{2}{3}r$ ones. Similarly, $d(\beta, \alpha_k) \leq r$ implies that $\beta[1..f]$ contains $\leq \frac{2}{3}r$ zeros. But this yields $f \leq \frac{2}{3}r + \frac{2}{3}r = \frac{4}{3}r$, contradicting $f > \frac{4}{3}r$.□

Let $r'$ be the maximum distance between $\beta'$ and a string $\alpha_i$, $i = 1, .., k$. Since $f \geq \frac{2\ln(4k)}{0.38\varepsilon^3}$, it follows from Lemma 4.1 and Lemma 4.2 that $r' \leq r(1+\varepsilon) + r/3$ with probability $\geq \frac{1}{2}$. To conclude Subcase 2b, transform $\beta'$ back to the original set of strings, apply $\sigma^{-1}$ to $\beta'$ and flip $\beta'[m]$ to $1 - \beta'[m]$ whenever $\alpha_1[m] = 1$.

THEOREM 4.1. *The approximative center $\beta'$ found by the algorithm for an instance of HCP is within distance $(\frac{4}{3}+\varepsilon)r$ of any string $\alpha_i$, $i = 1, .., k$, with probability $\geq \frac{1}{2}$. If the diameter $f$ of the input instance satisfies $f \geq \frac{2\ln(4k)}{0.38\varepsilon^3}$ then $\beta'$ will be constructed in polynomial time. If $f < \frac{2\ln(4k)}{0.38\varepsilon^3}$, then $\beta'$ is actually an optimal solution, and will be found in $O(kn^{\frac{2\ln(4k)}{0.38\varepsilon^3}+1})$ time.*

Analogously as the similar approximation algorithm from [4], our randomized algorithm can be easily derandomized using the method of conditional probabilities.

## 5 Other Types of Approximation

To find a small set of approximative centers instead of a single one, we transform the given instance of HCP to its integer programming formulation. By using the relaxation and randomized rounding, we can find in polynomial time an approximative solution $\beta'_1$ that satisfies at least a quarter of the inequalities with probability greater than $\frac{1}{3}$. After that, we remove all inequalities satisfied by $\beta'_1$, and iterate our method for the remaining inequalities to obtain a solution $\beta'_2$, remove all inequalities satisfied by $\beta'_2$, and so on. After $K = O(\log k)$ iterations, each inequality is satisfied by some $\beta'_j$, $1 \leq j \leq K$, with probability $\geq n^{-O(1)}$.

THEOREM 5.1. *For an instance of HCP with binary strings $\alpha_i$, $i = 1, .., k$, each of length $n$, we can find, in expected time polynomial in $n+k$, a set of binary strings $\beta'_j$, $j = 1, .., K$ with $K = O(\log k)$, such that for each $\alpha_i$, $i = 1, .., k$, there is a $j \in \{1, .., K\}$ for which the Hamming distance between $\alpha_i$ and $\beta'_j$ doesn't exceed the radius of the instance.*

## References

[1] P. Berman, D. Gumucio, R. Hardison, W. Miller, and N. Stojanovic, A Linear-Time Algorithm for the 1-Mismatch Problem, *Proc. of* WADS'97, 1997.

[2] M. Frances and A. Litman, On Covering Problems of Codes, *Theory of Comp. Syst.* **30**, pp. 113–119, 1997.

[3] D.S. Hochbaum, *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Comp., Boston, 1996.

[4] J.K. Lanctot, M. Li, B. Ma, S. Wang and L. Zhang, *Distinguishing String Selection Problems*, to appear in this proceedings.