Hardness of Median and Center in the Ulam Metric

Nick Fischer
INSAIT, Sofia University "St. Kliment Ohridski", Bulgaria

Elazar Goldenberg
The Academic College of Tel Aviv-Yaffo, Israel

Mursalin Habib
Rutgers University, Piscataway, NJ, USA

Karthik C. S.
Rutgers University, Piscataway, NJ, USA

Abstract

The classical *rank aggregation* problem seeks to combine a set X of n permutations into a single representative "consensus" permutation. In this paper, we investigate two fundamental rank aggregation tasks under the well-studied *Ulam metric*: computing a *median* permutation (which minimizes the sum of Ulam distances to X) and computing a *center* permutation (which minimizes the maximum Ulam distance to X) in two settings.

Continuous Setting: In the continuous setting, the median/center is allowed to be any permutation. It is known that computing a center in the Ulam metric is NP-hard and we add to this by showing that computing a median is NP-hard as well via a simple reduction from the Max-Cut problem. While this result may not be unexpected, it had remained elusive until now and confirms a speculation by Chakraborty, Das, and Krauthgamer [SODA '21].

Discrete Setting: In the discrete setting, the median/center must be a permutation from the input set. We fully resolve the fine-grained complexity of the discrete median and discrete center problems under the Ulam metric, proving that the naive $\widetilde{O}(n^2 L)$-time algorithm (where L is the length of the permutation) is conditionally optimal. This resolves an open problem raised by Abboud, Bateni, Cohen-Addad, Karthik C. S., and Seddighin [APPROX '23]. Our reductions are inspired by the known fine-grained lower bounds for similarity measures, but we face and overcome several new highly technical challenges.

2012 ACM Subject Classification Mathematics of computing → Permutations and combinations; Mathematics of computing → Combinatorial optimization; Theory of computation → Problems, reductions and completeness

Keywords and phrases Ulam distance, median, center, rank aggregation, fine-grained complexity

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.111

Related Version *Full Version*: https://arxiv.org/abs/2504.16437 [25]

Funding Part of this work was done while Mursalin Habib and Karthik C. S. were visiting INSAIT, Sofia University "St. Kliment Ohridski". Nick Fischer, Mursalin Habib, and Karthik C. S. were partially funded by the Ministry of Education and Science of Bulgaria's support for INSAIT as part of the Bulgarian National Roadmap for Research Infrastructure.
Mursalin Habib: Supported by the National Science Foundation under Grants CCF-2313372, CCF-2422558, and CCF-2443697.
Karthik C. S.: Supported by the National Science Foundation under Grants CCF-2313372, CCF-2422558, and CCF-2443697.

Acknowledgements The authors would like to thank the *DIMACS Workshop on Efficient Algorithms for High Dimensional Metrics: New Tools* for a special collaboration opportunity as well as Siam Habib and Antti Roeyskoe for helpful discussions.

1 Introduction

Suppose that n judges each rank the performances of L competitors. Given these rankings, how can the judges agree on a single consensus ranking? This fundamental question lies at the heart of a class of tasks known as *rank aggregation*, which has applications across various fields, including social choice theory [11], bioinformatics [37], information retrieval [29], machine learning [38], and recommendation systems [41], among others. Formally, the judges' rankings can be represented as a set of n permutations $X \subseteq \mathcal{S}_L$. Then, for an appropriate metric $d(\cdot,\cdot)$ on the space of permutations $\mathcal{S}_L$, the two most prominent rank aggregation tasks are to compute a *median* permutation π_M which minimizes $\sum_{\pi \in X} d(\pi_M, \pi)$ [33, 49, 50, 23], or a *center* permutation π_C which minimizes $\max_{\pi \in X} d(\pi_C, \pi)$ [8, 10, 43].

Among the metrics studied in this context, two stand out. The first one is the classic *Kendall's tau distance* which measures the number of disagreeing pairs between two permutations, i.e., the number of pairs (i, j) for which one ranking orders i before j while the other orders j before i. Kendall's tau distance is well-motivated as it satisfies several desirable properties beyond the scope of this paper (e.g., neutrality, consistency, and the extended Condorcet property [33, 47]), and is also well-understood from a computational point-of-view [23, 24]. For example, it is known that computing the median or center of just *four* permutations is already NP-hard [23, 10]. Several approximation algorithms have also been proposed for this metric [5, 46], culminating in a PTAS [34, 42] for approximating the median under Kendall's tau metric.

The other key metric is the *Ulam distance* which measures the minimum number of relocation operations required to turn one permutation π into another permutation π' – i.e., the minimum number of competitors whose ranks have to be adjusted in π so that it agrees with π'. This metric offers a simpler and more practical alternative to Kendall's tau metric for rank aggregation tasks [20, 16, 19, 18]. Perhaps more importantly, the Ulam metric is intimately linked to the more general *edit metric* on arbitrary strings, which enjoys countless applications in computational biology [28, 42], specifically in the context of DNA storage systems [27, 44], and beyond [35, 39]. Despite the significance of Ulam rank aggregation problems and the extensive research dedicated to them [8, 16, 19, 18], some basic questions remain unanswered. This is the starting point of our paper.

1.1 Question 1: Polynomial-Time Algorithms for Ulam Median?

The first basic question is whether polynomial-time algorithms exist for exactly computing the center and median permutations under the Ulam metric. For almost all string metrics, including the aforementioned Kendall's tau metric but also metrics beyond permutations such as the Hamming metric [26, 36, 4] or the edit metric [22, 40], median and center problems are well-known and easily-proven to be NP-hard.[1] Quite surprisingly, while it is known that computing an Ulam center is NP-hard [8], the complexity of computing an Ulam median has remained an open question. This is not due to a lack of interest – despite the absence of an NP-hardness proof, Chakraborty, Das, and Krauthgamer have already initiated the study of approximation algorithms for the Ulam median problem [16, 17], achieving a 1.999-factor approximation in polynomial time. Our first contribution is that we finally provide this missing hardness proof:

► **Theorem 1.** *The median problem is NP-hard in the Ulam metric.*

[1] A notable exception is the Hamming median problem that can trivially be solved in polynomial time by a coordinate-wise plurality vote.

1.2 Question 2: Fine-Grained Complexity of Discrete Ulam Center and Median?

How can we circumvent this new lower bound? There are two typical approaches. The first is to resort to approximation algorithms as was done in [16, 17]. But there is a commonly-studied second option for aggregation and clustering type problems: to restrict the solution space to *only the input set of permutations* X and compute the best median or center from it.[2] The best median or center from the input set X is typically referred to as the *discrete median* or the *discrete center*, respectively, of X. (In the same spirit, we will occasionally refer to the unrestricted median and center, discussed in the previous subsection, as the *continuous median* and *continuous center*, respectively.) Besides being a natural polynomial-time rank aggregation task, computing discrete medians or centers has two other motivations. First, it is easy to see that computing a discrete median/center yields a 2-approximation of the (continuous) median/center problems. In fact, a key observation in [16] is that a discrete median often provides a $(2-\varepsilon)$-approximation for the (continuous) median, particularly in practical DNA storage system instances where distances tend to be large. Second, the discrete median and center problems have gained significant attraction for the easier Hamming metric [2, 4] and harder edit metric [2], often leading to matching upper and lower bounds. Studying the Ulam metric therefore serves as an interesting intermediate problem capturing some – but not all – of the hardness of the edit metric.

Driven by these motivations, we study the *fine-grained complexity* of the discrete median and discrete center problems with respect to the Ulam distance. That is, we aim to pinpoint their *precise* polynomial run times.

Discrete Ulam Center

The trivial algorithm for computing a discrete center is to explicitly compute the Ulam distance $d_U(\pi, \pi')$ for all pairs of permutations $\pi, \pi' \in X$. We can then easily select the permutation $\pi_C \in X$ minimizing $\max_{\pi \in X} d_U(\pi_C, \pi)$. As the Ulam distance between two length-L permutations can be computed in near-linear time[3] $\widetilde{O}(L)$ using the well-known patience sorting algorithm [6], the total time is $\widetilde{O}(n^2 L)$.

We prove that this simple algorithm is *optimal*, up to subpolynomial factors and conditioned on a plausible assumption from fine-grained complexity:

▶ **Theorem 2.** *Let $\varepsilon > 0$ and $\alpha > 0$. There is no algorithm running in time $O((n^2L)^{1-\varepsilon})$ that solves the discrete center problem in the Ulam metric for n permutations of length $L = \Theta(n^\alpha)$, unless the Quantified Strong Exponential Time Hypothesis fails.*

The Quantified Strong Exponential Time Hypothesis (QSETH) is a plausible generalization of the by-now well-established Strong Exponential Time Hypothesis (SETH), postulating that the CNF-SAT problem cannot be solved faster than brute-force search even when only some variables are existentially quantified and others are universally quantified (see Section 3.1 for the formal treatment) [12]. This hypothesis has already proven useful for conditional lower bounds for a wide array of problems [12, 3, 2]. Besides, we remark that it is impossible to obtain SETH-based lower bounds for the discrete center problem unless the *Nondeterministic* Strong Exponential Time Hypothesis [15] is false (see Section 5.1).

[2] Or alternatively, the best median/center among the permutations in another given set $Y \subseteq \mathcal{S}_L$; this is typically referred to as the *bichromatic* discrete center/median problem, or as center/median problem with *facilities* in the theory of clustering. All of our results also apply to these bichromatic variants.

[3] We write $\widetilde{O}(T) = T(\log T)^{O(1)}$ to suppress polylogarithmic factors.

We emphasize that these lower bounds applies to the full range of n versus L, as long as L is at least polynomial in n. In the case when L is very small, $\omega(\log n) < L < n^{o(1)}$, previous work by Abboud, Bateni, Cohen-Addad, Karthik C. S. and Seddighin already established a matching conditional lower bound of $n^{2-o(1)}$ [2].

It is interesting to compare Theorem 2 with the state of the art for discrete center problems in the Hamming metric (say, over a constant-size alphabet) and the edit metric. For concreteness, consider the case $L = \Theta(n)$ (i.e., the input consists of n strings/permutations of length roughly n). Then, on the one hand, the discrete center problem in the Hamming metric can be solved in time $O(n^{\omega})$ [2, 4], where $\omega < 2.3714$ is the exponent of matrix multiplication [7]. On the other hand, the discrete center problem in the edit metric cannot be solved in time $n^{4-\Omega(1)}$ unless QSETH fails [2]. Therefore, remarkably, Theorem 2 indeed places the discrete center problem in the Ulam metric as a problem of *intermediate* complexity $n^{3\pm o(1)}$. This answers an explicit open question posed by Abboud, Bateni, Cohen-Addad, Karthik C. S., and Seddighin [2].

Discrete Ulam Median

The trivial algorithm for the discrete median problem is exactly the same as for the center problem: First compute all pairwise distances $d_U(\pi, \pi')$, then select the permutation π_M minimizing $\sum_{\pi \in X} d_U(\pi_M, \pi)$. It also runs in time $\widetilde{O}(n^2 L)$. So perhaps one could also hope that a matching lower bound follows from our Theorem 2. Unfortunately, this turns out to be true only for a restricted subproblem.[4] Nevertheless, with considerable technical overhead we manage to prove essentially the same *matching* lower bound:

▶ **Theorem 3.** *Let $\varepsilon > 0$ and $\alpha \geq 1$. There is no algorithm running in time $O((n^2 L)^{1-\varepsilon})$ that solves the discrete median in the Ulam metric for n permutations of length $L = \Theta(n^{\alpha})$, unless the SETH fails.*

In comparison to Theorem 2, Theorem 3 has the advantage that it conditions on the *weaker* assumption SETH (thus constituting a *stronger* lower bound). However, the applicable range of parameters is more restricted ($\alpha \geq 1$ forces the permutations to have length $\Omega(n)$).

2 Proof Overview

In this section, we provide the proof overviews of Theorems 1, 2, and 3. We avoid some technicalities to highlight the core ideas – for instance, some subpolynomial factors have been dropped. Complete proofs can be found in Sections 4, 5, and 6, respectively.

2.1 NP-Hardness for Continuous Median in the Ulam Metric

In this section, we provide a high-level sketch of our NP-hardness proof for the continuous median problem in the Ulam metric. Our starting point is the Max-Cut problem[5]. Given a Max-Cut instance $G = (V, E)$ where $V = [n]$, our goal is to construct a set of permutations of length $O(n)$ such that the median of these permutations encodes the cut in G of maximum size.

[4] Namely, the *bichromatic* discrete median problem mentioned in Footnote 2.

[5] Recall that the Max-Cut problem is, given an undirected graph $G = (V, E)$ to compute a vertex partition $V = A \sqcup B$ maximizing the number of edges from A to B.

To achieve this, we first set up a natural correspondence between cuts in G and permutations of length $O(n)$. However, since there are many more permutations than cuts, not all permutations will represent valid cuts. To ensure that only relevant permutations are considered, we construct two special permutations:

$$\pi^L = 1 \circ 2 \circ \cdots \circ (n-1) \circ n \circ X_1 \circ X_2,$$
$$\pi^R = X_1 \circ n \circ (n-1) \circ \cdots \circ 2 \circ 1 \circ X_2,$$

for some fixed long permutations X_1, X_2. The simple key insight is that any median of π^L and π^R has the following form: it starts with some subset $A \subseteq [n]$ of the symbols in *increasing* order, followed by X_1, followed by the symbols in $[n] \setminus A$ in *decreasing* order, finally followed by X_2. Thus, medians of π^L, π^R will naturally encode cuts of the form $(A, [n] \setminus A)$ in G.

To further enforce that the median represents a *maximum* cut, we include additional permutations in our instance. Specifically, for each edge $e \in E$, we include two permutations, π_e^1 and π_e^2, which reward picking solutions that correspond to partitions that cut e. This ensures that the final median permutation encodes a maximum cut of G. The precise construction and formal analysis of these edge-cutting permutations π_e^1, π_e^2 is detailed in Section 4.

2.2 Fine-Grained Lower Bound for Discrete Center in the Ulam Metric

Our proof of the lower bound for the discrete center problem in the Ulam metric relies on two key components. The first is a reduction from the Orthogonal Vectors (OV) problem to the problem of computing the Ulam distance between two permutations. Specifically, we seek a pair of functions that, given two sets of binary vectors as inputs, independently output two permutations whose Ulam distance is small if and only if there exists an orthogonal pair of vectors in the input sets. This falls into a well-established framework in the fine-grained complexity literature [9, 13, 1]: Given two sets of roughly L binary vectors, one constructs "coordinate gadgets", "vector gadgets", and "OR-gadgets" to produce a pair of length-L strings whose edit distance encodes the existence of an orthogonal pair.

Clearly such a reduction cannot exist for the Ulam distance under SETH. The Ulam distance between two permutations can be computed in near-linear time by a simple reduction to the longest increasing subsequence problem [45], and thus, if there were a way to transform sets of $O(L)$ many vectors into permutations of length L such that the existence of an orthogonal pair in the sets could be determined via an Ulam distance computation of these permutations, then that would imply a near-linear time algorithm for OV, falsifying SETH! In light of this observation, we start with $O(\sqrt{L})$ vectors in the OV instance. The constructions of the coordinate and vector gadgets are similar to the edit distance reduction. However, during the construction of the OR-gadgets, there is a quadratic blowup resulting in length L permutations with the desired properties. A detailed construction of these gadgets can be found in Section 5.

The second component in our proof is a reduction from a problem called ∃∀∃∃-Orthogonal Vectors. In this problem, we are given four sets A, B, C, E of binary vectors and we have to decide if there exists $a \in A$, such that for all $b \in B$, there exist $c \in C, e \in E$ such that a, b, c, e are orthogonal[6]. If $|A| = |B| = n$ and $|C| = |E| = \sqrt{L}$, then this problem has a

[6] We say that vectors $a, b, c, e \in \{0,1\}^d$ are *orthogonal* if $\sum_{i \in [d]} a[i]b[i]c[i]e[i] = 0$.

$O((n^2L)^{1-\Omega(1)})$ lower bound under the Quantified Strong Exponential Time Hypothesis. Given such sets A, B, C, E, we proceed as follows. First, for each $a \in A$, we construct the set V_a of $\sqrt{L}$ vectors by taking the pointwise product of a with all $\sqrt{L}$ vectors in C. There will be n such sets V_a, one for each choice of $a \in A$. Similarly, for each $b \in B$, we construct the set W_b of $\sqrt{L}$ vectors by taking the pointwise product of b with all $\sqrt{L}$ vectors in C. We then run the OV to Ulam distance reduction from before on these sets to obtain two sets of n-many permutations of length L. Finally, we show that there exists a permutation in the first set with small Ulam distance to every permutation in the second set if and only if the starting ∃∀∃∃-OV instance is a YES-instance.

To go from these two sets to the final discrete center instance, we append additional symbols to each permutation and introduce a new permutation that is far from every permutation in the second set. This ensures that the center indeed comes from the first set completing the reduction. We defer the details to Section 5.

2.3 Fine-Grained Lower Bound for Discrete Median in the Ulam Metric

Our lower bound proof for the discrete median problem follows a similar initial approach as our proof for the center lower bound, with only one difference: instead of starting with an ∃∀∃∃-OV instance, we begin with a ∃∃∃∃-OV (also known as 4-OV) instance. Given this 4-OV instance, we retrace the same steps to construct two sets, X and Y, each containing n permutations of length L. As in the center proof, we show that there exists a permutation in X whose total Ulam distance to all elements in Y is small if and only if the original 4-OV instance is a YES-instance.

However, going from these two sets to the standard single-set version of the discrete median problem is technically very challenging. In fact, such challenges were addressed in the past in the context of the closest pair problem [21, 32], and more generally identified as the task of *reversing color coding* [14], typically requiring extremal combinatorial objects which are then composed with the input in a black-box manner.

In this paper, we transform the bichromatic instance to a monochromatic one, in multiple steps but in a white-box manner using the structure of the input instance. The first key observation is that all pairwise Ulam distances within X can be computed much faster than the naive $O(n^2L)$ time bound, specifically, in $O(n^2\sqrt{L})$ time. This speedup is possible because the permutations in X are not arbitrary but outputs formed during our OV to Ulam distance reduction. Thus, in $O(n^2\sqrt{L})$ time, we can compute the total Ulam distance of each $x \in X$ to all other elements in X.

Once these n distance sums are computed, we initiate a *balancing* procedure. This procedure iteratively appends additional symbols to each permutation in $X \cup Y$ such that:

- For every permutation in X, the sum of its Ulam distances to all other elements in X becomes equal.
- The relative Ulam distances between permutations across the sets remain unchanged.
- For every permutation in Y, the sum of its Ulam distances to all other elements in Y becomes very large.

We show that this balancing procedure can be performed efficiently without significantly increasing the permutation lengths. Finally, we include all modified permutations into a single set, and output that as our final discrete median instance. The details turn out to be quite involved, and we direct the reader to Section 6 for further details.

3 Preliminaries

Sets, Strings and Permutations

For a positive integer n, let $[n]$ denote the set $\{1, 2, \ldots, n\}$. We denote by $\mathcal{S}_n$ the set of all permutations over $[n]$. Throughout the paper, we treat any permutation $s \in \mathcal{S}_n$ as a string over the alphabet $[n]$ with no repeating symbols, and we write $s := s[1]s[2] \ldots s[n]$. Given k strings $s_1, s_2, \ldots, s_k$ over some alphabet, we denote by $\bigcirc_{i\in[k]} s_i$ the concatenated string $s_1 s_2 \ldots s_k$. We will often require strings that can be obtained by adding some fixed "offset" to each symbol of some other canonical string[7]. For every nonnegative integer k and string of length n, we let $\Delta_k(s) := (s[1] + k)(s[2] + k) \ldots (s[n] + k)$. In other words, $\Delta_k(s)$ is the string obtained by adding k to each symbol of s. We will also often require strings that are obtained by restricting some string to some sub-alphabet. Given any string s over the alphabet Σ and any sub-alphabet $\Sigma' \subseteq \Sigma$, we denote by $s|_{\Sigma'}$ to be string obtained from s by deleting all symbols that are not in Σ'. Given $a, b \in \{0, 1\}^d$, we write $\langle a, b\rangle$ for the inner product of a and b, i.e., $\langle a, b\rangle = \sum_{i\in[d]} a[i]b[i]$. We further write $a \odot b$ for the pointwise product a and b, i.e., $a \odot b \in \{0, 1\}^d$ is the vector satisfying $(a \odot b)[i] = a[i]b[i]$ for all $i \in [d]$.

Ulam Distances and Common Subsequences

The Hamming distance between two equal-length strings x and y, denoted by $d_H(x, y)$, is the number of locations where x and y have different symbols. Let $\pi := \pi[1]\pi[2] \ldots \pi[n] \in \mathcal{S}_n$ be a permutation and $i, j \in [n]$ be distinct positions. A *symbol relocation* operation from position j to position i in π constitutes of deleting the j^{th} symbol of π and reinserting it at position i. More formally, if $\widetilde{\pi} \in \mathcal{S}_n$ is the permutation obtained after applying a symbol relocation from position j to position i in π, then:

$$\widetilde{\pi} := \begin{cases} \pi[1]\pi[2] \ldots \pi[j-1]\pi[j+1] \cdots \pi[i-1]\pi[j]\pi[i]\pi[i+1] \cdots \pi[n], & \text{if } j < i, \\ \pi[1]\pi[2] \cdots \pi[i-1]\pi[j]\pi[i]\pi[i+1] \cdots \pi[j-1]\pi[j+1] \cdots \pi[n], & \text{if } j > i. \end{cases}$$

Given $\pi, \pi' \in \mathcal{S}_n$, the Ulam distance between π and π', denoted by $d_U(\pi, \pi')$, is the minimum number of symbol relocation operations required to transform π into π'. We will further denote by $\mathsf{LCS}(x, y)$ the length of a longest common subsequence of two strings x and y. We will frequently use the following fact throughout the paper, which relates the Ulam distance between two permutations to the length of their longest common subsequence.

▶ **Fact 4** ([6]). *For every $\pi, \pi' \in \mathcal{S}_n$, we have $d_U(\pi, \pi') = n - \mathsf{LCS}(\pi, \pi')$.*

3.1 Hardness Assumptions

Our results are conditional on several hardness assumptions and hypotheses, which we list in this section. The first one is the Strong Exponential Time Hypothesis, which is a standard assumption in the theory of fine-grained complexity.

▶ **Hypothesis 5** (Strong Exponential Time Hypothesis (SETH)). *For all $\varepsilon > 0$, there exists $q \geq 3$ such that no algorithm running in time $O(2^{(1-\varepsilon)n})$ can solve the q-SAT problem on n variables.*

[7] Here we are treating the symbols of a string as integers themselves.

More specifically, for one of our lower bounds, we will need the following corollary of SETH, which we dub Unbalanced 4-OVH.

▶ **Hypothesis 6** (Unbalanced 4-OVH). *For all $\varepsilon > 0$, no algorithm can, given sets $A, B, C, E \subseteq \{0,1\}^d$ with $|A| = n$, $|B|, |C|, |E| = n^{\Theta(1)}$ and $\omega(\log n) < d < n^{o(1)}$, determine if there exists $a \in A$, $b \in B$, $c \in C$, $e \in E$ such that $\sum_{i\in[d]} a[i]b[i]c[i]e[i] = 0$ in time $O((|A||B||C||E|)^{1-\varepsilon})$.*

It is well-known that SETH in conjunction with the sparsification lemma [30] implies Unbalanced 4-OVH [48]. We will also make use of the following strengthening of SETH.

▶ **Hypothesis 7** (∃∀∃SETH). *For all $\varepsilon > 0$ and $0 < \alpha < \beta < 1$, there exists $q \geq 3$, such that given a q-CNF formula ϕ over the variables $x_1, x_2, \ldots, x_n$, no algorithm running in time $O(2^{(1-\varepsilon)n})$ can determine if the following is true:*

$$\exists x_1, \ldots, x_{\lceil \alpha n\rceil} \forall x_{\lceil \alpha n\rceil+1}, \ldots, x_{\lceil \beta n\rceil} \exists x_{\lceil \beta n\rceil+1}, \ldots, x_n \; \phi(x_1, x_2, \ldots, x_n).$$

We note that ∃∀∃SETH is a special case of the Quantified SETH proposed by Bringmann and Chaudhury [12] – a hypothesis about the complexity of deciding quantified q-CNF formulas with a constant number of quantifier blocks where each block contains some constant fraction of the variables. We do not formally define Quantified SETH in all of its generality since we only require three quantfier alternations. In fact, the specific hardness assumption we need is the following which is implied by ∃∀∃SETH.

▶ **Hypothesis 8** (Unbalanced ∃∀∃∃OVH). *For all $\varepsilon > 0$, no algorithm can, given sets $A, B, C, E \subseteq \{0,1\}^d$ with $|A| = n$, $|B|, |C|, |E| = n^{\Theta(1)}$ and $\omega(\log n) < d < n^{o(1)}$, determine if there exists $a \in A$ such that for all $b \in B$, there exist $c \in C, e \in E$ such that $\sum_{i\in[d]} a[i]b[i]c[i]e[i] = 0$ in time $O((|A||B||C||E|)^{1-\varepsilon})$.*

4 NP-Hardness of Continuous Median in the Ulam Metric

In this section, we prove Theorem 1. Before we do so, we first formally define the continuous median problem in the Ulam metric.

Continuous Ulam Median	
Input:	A set $S \subseteq \mathcal{S}_n$ of permutations and an integer d.
Question:	Is there a permutation $\pi^* \in S_n$ such that $\sum_{\pi\in S} d_U(\pi, \pi^*) \leq d$?

The main result of this section is the following. All missing proofs in this section can be found in the full version of this paper [25].

▶ **Theorem 9.** *Continuous Ulam Median is NP-hard.*

Proof. We will reduce from the Max Cut problem, which is NP-hard [31]. Let $G = (V, E)$ be a Max Cut instance with $V = [n]$. From G, we will construct a Continuous Ulam Median instance $S \subseteq \mathcal{S}_N$ consisting of permutations of length $N := 3n + 2$. On a high level, our construction will work as follows. We will first construct many copies of two special permutations that will force every median of S to take on a very specific structure. All permutations of this structure will naturally encode cuts of the vertex set V. Then for each edge e in G, we will construct permutations that reward choosing a median that "cuts" e. Thus, we will end up with a set of permutations whose overall median will encode the maximum cut of G. Details follow.

To describe our construction, it will be convenient to define the following two strings, both of length $(n+1)$:

$$X_1 := (n+1) \circ (n+2) \circ \cdots \circ (2n+1),$$
$$X_2 := (2n+2) \circ (2n+3) \circ \cdots \circ (3n+2).$$

Next, we define the two special permutations $\pi^L, \pi^R \in \mathcal{S}_N$ alluded to earlier:

$$\pi^L := 1 \circ 2 \circ \cdots \circ (n-1) \circ n \circ X_1 \circ X_2,$$
$$\pi^R := X_1 \circ n \circ (n-1) \circ \cdots \circ 2 \circ 1 \circ X_2.$$

We make the observation that every median of the set $\{\pi^L, \pi^R\}$ naturally encodes a cut of the vertex set V.

▶ **Definition 10.** *For a nonnegative integer $r \leq n$, let $A = \{a_1, a_2, \ldots, a_r\}$ and $B = \{b_1, b_2, \ldots, b_{n-r}\}$ be sets such that $A \sqcup B = [n]$, $a_1 < a_2 < \cdots < a_r$ and $b_1 > b_2 > \cdots > b_{n-r}$. Define $\pi^{A,B} \in \mathcal{S}_N$ as:*

$$\pi^{A,B} := a_1 \circ a_2 \circ \cdots \circ a_r \circ X_1 \circ b_1 \circ b_2 \circ \cdots \circ b_{n-r} \circ X_2.$$

Furthermore, define $\mathcal{S}_N^* \subseteq \mathcal{S}_N$ as:

$$\mathcal{S}_N^* := \{\pi \in \mathcal{S}_N : \pi = \pi^{A,B} \text{ for some pair of sets } A, B \text{ with } A \sqcup B = [n]\}.$$

Clearly, permutations in $\mathcal{S}_N^*$ naturally encode cuts of $[n]$. We will first show that every permutation in $\mathcal{S}_N^*$ has the same sum of Ulam distances to π^L and π^R.

▶ **Lemma 11.** *For every $\pi \in \mathcal{S}_N^*$, $d_U(\pi, \pi^L) + d_U(\pi, \pi^R) = n$.*

Next, we show that any permutation that is not in $\mathcal{S}_N^*$ has strictly larger sum of Ulam distances to π^L and π^R.

▶ **Lemma 12.** *Let $\pi \notin \mathcal{S}_N^*$. Then $d_U(\pi, \pi^L) + d_U(\pi, \pi^R) \geq n + 1$.*

The final pieces in our construction are the "edge gadgets", which we now define. For each edge $e = \{i, j\} \in E$, where $i < j$, define the following two strings:

$$\pi_e^1 = j \circ i \circ X_1 \circ X_2 \circ 1 \circ 2 \circ \cdots \circ (i-1) \circ (i+1) \circ \cdots \circ (j-1) \circ (j+1) \circ \cdots \circ n,$$
$$\pi_e^2 = X_1 \circ i \circ j \circ X_2 \circ 1 \circ 2 \circ \cdots \circ (i-1) \circ (i+1) \circ \cdots \circ (j-1) \circ (j+1) \circ \cdots \circ n.$$

The role of the edge gadgets associated with an edge e is to reward choosing partitions of the vertices that cut e. This is formalized in the following lemma.

▶ **Lemma 13.** *Let $\pi \in \mathcal{S}_N^*$ such that $\pi = \pi^{A,B}$ with $A \sqcup B = [n]$ and $e \in E$. Then we have the following:*

$$d_U(\pi_e^1, \pi) + d_U(\pi_e^2, \pi) = \begin{cases} 2n-2, & \text{if } e \text{ is cut by the partition } (A, B), \\ 2n-1, & \text{otherwise.} \end{cases}$$

Now let $S_E = \{\pi_e^1 : e \in E\} \cup \{\pi_e^2 : e \in E\}$ and S_{aux} be the set consisting of $t := |E|(2n-1)$ copies of π^L and π^R. Our final Continuous Ulam Median instance will be the multiset

$$S := S_E \cup S_{\text{aux}}.$$

We now show that G has a cut of size at least k if and only if the median of S has cost at most k', where $k' = |E|(2n-1) - k + tn$. For the completeness case, assume there exist sets A, B with $A \sqcup B = n$ such that the partition (A, B) cuts at least k edges in G. Denote by $E(A, B)$ the set of all edges cut by the partition (A, B). Now consider the permutation $\pi^{A,B} \in \mathcal{S}_n$ and note that:

$$\begin{aligned}
\sum_{\pi \in S} d_U(\pi^{A,B}, \pi) &= \sum_{\pi \in S_E} d_U(\pi^{A,B}, \pi) + \sum_{\pi \in S_{\text{aux}}} d_U(\pi^{A,B}, \pi) \\
&= \left(\sum_{e \in E} (d_U(\pi^{A,B}, \pi_e^1) + d_U(\pi^{A,B}, \pi_e^2)) \right) + t(d_U(\pi^{A,B}, \pi^L) + d_U(\pi^{A,B}, \pi^R)) \\
&= \left(\sum_{e \in E(A,B)} (d_U(\pi^{A,B}, \pi_e^1) + d_U(\pi^{A,B}, \pi_e^2)) \right) \\
&\qquad + \left(\sum_{e \notin E(A,B)} (d_U(\pi^{A,B}, \pi_e^1) + d_U(\pi^{A,B}, \pi_e^2)) \right) + tn \\
&= (|E(A, B)|(2n-2) + (|E| - |E(A, B)|)(2n-1)) + tn \\
&= |E|(2n-1) - |E(A, B)| + tn \leq |E|(2n-1) - k + tn.
\end{aligned}$$

For the soundness case, assume that there exists $\pi^* \in \mathcal{S}_N$ whose median cost to S is at most k'. We can further assume that $\pi^* \in \mathcal{S}_N^*$ since otherwise, every $\widetilde{\pi} \in \mathcal{S}_N^*$ will have a median cost that is at most that of π^*. Indeed, assume that $\pi^* \notin \mathcal{S}_N^*$ and fix any $\widetilde{\pi} \in \mathcal{S}_N^*$. We have:

$$\begin{aligned}
\sum_{\pi \in S} d_U(\pi^*, \pi) &= \sum_{\pi \in S_{\text{aux}}} d_U(\pi^*, \pi) + \sum_{\pi \in S_E} d_U(\pi^*, \pi) \\
&\geq t(n+1) + 0 = tn + t \\
&= tn + |E|(2n-1) \geq tn + \sum_{\pi \in S_E} d_U(\widetilde{\pi}, \pi) \\
&= \sum_{\pi \in S_{\text{aux}}} d_U(\widetilde{\pi}, \pi) + \sum_{\pi \in S_E} d_U(\widetilde{\pi}, \pi) = \sum_{\pi \in S} d_U(\widetilde{\pi}, \pi).
\end{aligned}$$

Thus, the assumption that $\pi^* \in \mathcal{S}_N^*$ is without loss of generality. Then, we have $\pi^* = \pi^{A,B}$ for sets A, B with $A \sqcup B = [n]$. We claim that the partition (A, B) cuts at least k edges in G. Indeed, since $\sum_{\pi \in S_{\text{aux}}} d_U(\pi^*, \pi) = tn$, we have $\sum_{\pi \in S_E} d_U(\pi^*, \pi) \leq k' - tn = |E|(2n-1) - k = k(2n-2) + (|E| - k)(2n-1)$. Then, by Lemma 13 the partition (A, B) cuts at least k edges. ◀

▶ Remark 14. Although our NP-hardness reduction produces multisets instead of sets, it is not to hard to turn them into sets by appending a unique permutation to each permutation without affecting the structure of the solution. See [25, Appendix B] for details.

5 Fine-Grained Complexity of Discrete Center in the Ulam Metric

In this section, we prove Theorem 2. We first formally define the discrete center problem in the Ulam metric.

The main result of this section is the following. All missing proofs in this section can be found in the full version of the paper [25].

Discrete Ulam Center	
Input:	A set $S \subseteq \mathcal{S}_L$ of permutations such that $\lvert S\rvert = n$ and an integer τ.
Question:	Is there a permutation $\pi^* \in S$ such that $\max_{\pi\in S} d_U(\pi, \pi^*) \leq \tau$?

▶ **Theorem 15.** *Let $\varepsilon > 0$ and $\alpha > 0$. There is no algorithm running in time $O((n^2L)^{1-\varepsilon})$ that solves the* Discrete Ulam Center *problem for n permutations of length $L = \Theta(n^\alpha)$, unless ∃∀∃SETH fails.*

The key step in the proof of Theorem 15 is a reduction from Orthogonal Vectors to Ulam Distance, i.e., to construct a pair of functions that, given a set of n binary vectors of length d each as input, outputs, independently of each other, a pair of permutations whose Ulam distance is small if and only if there exists an orthogonal pair of vectors in the input sets.

▶ **Theorem 16.** *There exists a pair of functions f and g such that for all sets $A, B \subseteq \{0,1\}^d$ with $\lvert A\rvert = \lvert B\rvert = n$, the following holds.*
- *$f(A), g(B) \in \mathcal{S}_{(5d-1)n^2}$, i.e., both f and g output permutations of length $(5d-1)n^2$.*
- *If there exists $(a, b) \in A \times B$ such that $\langle a, b\rangle = 0$, then the Ulam distance between $f(A)$ and $g(B)$ is at most $3n^2d - 1$. Otherwise, the Ulam distance between $f(A)$ and $g(B)$ is exactly $3n^2d$.*
- *Both f and g are computable in time $O(n^2d)$.*

Equipped with Theorem 16, Theorem 15 follows in a straightforward manner. For details, the reader is referred to [25].

5.1 The Need for Quantifiers

Our lower bound for Discrete Ulam Center is based on a plausible generalization of the Strong Exponential Time Hypothesis, namely ∃∀∃SETH. One could ask if we could get a similar lower bound under the weaker but more standard SETH instead. We remark that this is impossible unless the Nondeterministic Strong Exponential Time Hypothesis (NSETH) [15] is false. This is because Discrete Ulam Center can be solved in (co-)nondeterministic time $\widetilde{O}(nL)$ with the following simple algorithm. We first guess the center π_C and compute $d^* := \max_{\pi\in X} d_U(\pi_C, \pi)$. Then for each permutation π, we guess the furthest permutation $\pi' \in X$ and verify that $d_U(\pi, \pi') \geq d^*$, thereby certifying that our guess π_C is optimal. In light of this algorithm, a $O((n^2L)^{1-\Omega(1)})$ fine-grained lower bound based on SETH would contradict NSETH.

6 Fine-Grained Complexity of Discrete Median in the Ulam Metric

In this section, we prove a tight fine-grained lower bound for the Discrete Ulam Median problem conditioned on SETH. We first give a formal statement of the problem.

Discrete Ulam Median	
Input:	A set $S \subseteq \mathcal{S}_L$ of permutations such that $\lvert S\rvert = n$ and an integer τ.
Question:	Is there a permutation $\pi^* \in S$ such that $\sum_{\pi\in S} d_U(\pi, \pi^*) \leq \tau$?

This section is organized as follows. In Section 6.1, we start with a simple lower bound for the *bichromatic* version of the problem that essentially follows from the proof for the Discrete Ulam Center problem from before. Going from the bichromatic version to the standard monochromatic version defined above is presented in Section 6.2. This step is technically very involved: we show that any set of n permutations can be *balanced* in such a way that the sum of Ulam distances from each permutation to all the others is (approximately) *equal* (see Theorem 19).

6.1 Hardness for Bichromatic Instances

In this subsection, we give a fine-grained lower bound for the Bichromatic Discrete Ulam Median problem. In this problem, we given *two* sets X, Y of permutations and an integer τ, and the goal is to determine if there exists $x \in X$ such that $\sum_{y \in Y} d_U(x, y) \leq \tau$.

▶ **Theorem 17.** *Let $\varepsilon > 0$ and $\alpha > 0$. There is no algorithm running in time $O((n^2 L)^{1-\varepsilon})$ that solves the Bichromatic Discrete Ulam Median problem for n permutations of length $L = \Theta(n^\alpha)$, unless the SETH fails.*

Proof. The proof is almost identical to the first half of the proof of Theorem 15. The only difference is the starting problem, which is 4-OV (i.e., ∃∃∃∃OV) instead of ∃∀∃∃OV. Given a 4-OV instance $A, B, C, E \subseteq \{0,1\}^d$ where $|A| = |B| = n$, $|C| = |E| = m = n^{\Theta(1)}$ and $\omega(\log n) < d < n^{o(1)}$, we retrace the proof of Theorem 15 and construct the sets X and Y. Finally, we set $\tau := 3m^2 nd - 1$. Once again, by Theorem 16, it is not hard to see that there exists $x \in X$ such that $\sum_{y \in Y} d_U(x, y) \leq \tau$ if and only if the starting 4-OV instance is a YES-instance. The conclusion then follows from Unbalanced 4-OVH. ◀

6.2 Hardness for Monochromatic Instances

Finally, in this subsection, we provide a fine-grained lower bound for the Monochromatic Discrete Ulam Median problem. To do so, we need the following technical results (which is one of our main contributions), whose proofs can be found in the full version of the paper [25].

▶ **Lemma 18** (Embedding the Hamming Metric on Small Alphabets)**.** *Let $a_1, \dots, a_n \in \{0,1,2\}^L$. In time $O(nL)$ we can construct permutations $\pi_1, \dots, \pi_n \in \mathcal{S}_{3L}$ such that $d_H(a_i, a_j) = d_U(\pi_i, \pi_j)$ for all $i, j \in [n]$.*

▶ **Theorem 19** (Full Balancing)**.** *Let n and L be integers such that n is divisible by 4 and let $k_1, \dots, k_n \in [O(nL)]$. Given $k_1, \dots, k_n$, in time $\widetilde{O}(n^2 + nL)$ we can construct permutations $\pi_1, \dots, \pi_n, \tau \in \mathcal{S}_{\widetilde{O}(n+L)}$ and an integer d with the following two properties:*

- *Writing $d_i = \sum_{j \neq i} d_U(\pi_i, \pi_j)$, it holds that $|(k_i + d_i) - d| \leq 1$ for all $i \in [n]$.*
- *It holds that $d_U(\pi_1, \tau) = \cdots = d_U(\pi_n, \tau)$.*

Using the above theorem, we now prove the following:

▶ **Theorem 20.** *Let $\varepsilon > 0$ and $\alpha \geq 1$. There is no algorithm running in time $O((n^2 L)^{1-\varepsilon})$ that solves the (Monochromatic) Discrete Ulam Median problem for n permutations of length $L = \Theta(n^\alpha)$, unless the SETH fails.*

Proof. We follow the same reduction as in Theorem 17, which, given an initial 4-OV instance produces an instance (X, Y) of Bichromatic Discrete Ulam Median on n permutations of length L. During the reduction, we can set everything up so that $L = \Omega(n)$. Without loss of generality, we may also assume that n is divisible by 4 by adding some dummy vectors in the initial 4-OV instance if necessary. Let $x_1, \dots, x_n$ denote the permutations in X and let $y_1, \dots, y_n$ denote the permutations in Y. As a first preprocessing step, concatenate each permutation x_i (or y_i) two times to itself (using fresh symbols when necessary) so that each permutation becomes of length $3L$ and the median distance $\min_i \sum_j d_U(x_i, y_j)$ becomes a multiple of 3.

We first make the observation that the Ulam distance between any two permutations in X can be computed very quickly – in time that is proportional to the square root of the length of the permutations.

▶ **Observation 21.** *Let $x, x' \in X$. Then $d_U(x, x')$ can be computed in time $O(L^{1/2+o(1)})$.*

Proof. Since $x \in X$, there exist $T = \{t_1, t_2, \dots, t_m\} \subseteq \{0,1\}^d$ such that $x = f(T) \circ f(T) \circ f(T)$, where f is the function from Theorem 16 and each concatenation is done using a fresh set of symbols. Similarly, there exist $T' = \{t'_1, t'_2, \dots, t'_m\} \subseteq \{0,1\}^d$ such that $x' = f(T') \circ f(T') \circ f(T')$, where again each concatenation is with a fresh set of symbols.

Therefore, we have $d_U(x, x') = 3m \sum_{i \in [m]} d_H(t_i, t'_i)$. Thus, $d_U(x, x')$ can be found by computing the Hamming distance between two bit strings of length md. Further recall that $L = O(m^2 d)$. Therefore, $md = O(L^{1/2+o(1)})$ and the conclusion follows. ◀

By Observation 21, we can compute in time $O(n^2 L^{1/2+o(1)})$ all pairwise distances $d_U(x_i, x_j)$. Let $k_i := \sum_{j \neq i} d_U(x_i, x_j)$; clearly we have that $k_i \leq 3nL$. Thus, we may apply Theorem 19 to obtain permutations $\pi_1, \dots, \pi_n, \tau \in \mathcal{S}_{L'}$, where $L' = \widetilde{O}(n + L) = \widetilde{O}(L)$, with

$$|(k_i + \sum_{j \neq i} d_U(\pi_i, \pi_j)) - D| \leq 1$$

for some integer D and for all $i \in [n]$, and with $M := d_U(\pi_1, \tau) = \dots = d_U(\pi_n, \tau)$.

Additionally, let $K := 10(3L + L')$. Compute some length-$O(K)$ permutations $\mu, \eta_1, \dots, \eta_n$ such that $d_U(\mu, \eta_i) = K$ and such that $\sum_j d_U(\eta_i, \eta_j) = nK$. For instance, viewing $\mu, \eta_1, \dots, \eta_n$ as 0-1-strings under the Hamming distance to be embedded by Lemma 18, take μ to be the all-0 string of length $2K$, let half of the strings η_i be the string $0^K 1^K$ and let the other half of the strings η_i be the string $1^K 0^K$.

We are now ready to construct the Monochromatic Discrete Ulam Median instance Z. We include into Z the following $2n$ permutations:[8]

- $x'_i := x_i \, \pi_i \, \mu$ (for $i \in [n]$), and
- $y'_i := y_i \, \tau \, \eta_i$ (for $i \in [n]$).

We claim that this construction is correct in the following sense: (1) All discrete medians of Z are strings x'_i. (2) Whenever x'_i is a discrete median in Z, then x_i is a discrete median in (X, Y). (3) There is some discrete median x_i in (X, Y) such that x'_i is a discrete median in Z. The proofs of all three claims easily follow from the following calculations. On the one hand, the median distance for each x'_i is

[8] As always, we use fresh symbols when necessary to ensure that the resulting strings are permutations.

$$\begin{aligned}\sum_{z\in Z} d_U(x'_i, z) &= \sum_j d_U(x'_i, x'_j) + \sum_j d_U(x'_i, y'_j)\\ &= \sum_j (d_U(x_i, x_j) + d_U(\pi_i, \pi_j) + d_U(\mu, \mu))\\ &\qquad\qquad + \sum_j (d_U(x_i, y_j) + d_U(\pi_i, \tau) + d_U(\mu, \eta_i))\\ &= k_i + \sum_j d_U(\pi_i, \pi_j) + \sum_j d_U(x_i, y_j) + nM + nK\\ &= \sum_j d_U(x_i, y_j) + nM + nK + D \pm 1.\end{aligned}$$

On the other hand, the median distance for each y'_i is

$$\sum_{z\in Z} d_U(y'_i, z) = \sum_j d_U(y'_i, x'_j) + \sum_j d_U(y'_i, y'_j) \geq \sum_j d_U(\eta_i, \mu) + \sum_j d_U(\eta_i, \eta_j) = 2nK.$$

Comparing these two terms, and recalling that $\sum_j d_U(x_i, y_j) + nM + D \leq 3nL + nL' + nL' < nK$, it is clear that the median distance of any y'_i is always significantly larger that the median distance of any x'_i proving (1). Recalling further that all median distances $\sum_j d_U(x_i, y_j)$ in the original instance are multiples of 3, the ± 1 term in the first computation becomes irrelevant, completing the proofs of claims (2) and (3).

Finally we comment on the running time. The original reduction, along with the computation of the values k_i takes time $\widetilde{O}(nL + n^2 L^{1/2+o(1)})$. Running Theorem 19 takes time $\widetilde{O}(n^2 + nL)$, and the final instance Z can be implemented in negligible overhead. ◀

References

1. Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 59–78. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.14.
2. Amir Abboud, MohammadHossein Bateni, Vincent Cohen-Addad, Karthik C. S., and Saeed Seddighin. On complexity of 1-center in various metrics. In Nicole Megow and Adam D. Smith, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023, September 11-13, 2023, Atlanta, Georgia, USA*, volume 275 of *LIPIcs*, pages 1:1–1:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.APPROX/RANDOM.2023.1.
3. Amir Abboud, Karl Bringmann, Danny Hermelin, and Dvir Shabtay. Scheduling lower bounds via AND subset sum. *J. Comput. Syst. Sci.*, 127:29–40, 2022. doi:10.1016/J.JCSS.2022.01.005.
4. Amir Abboud, Nick Fischer, Elazar Goldenberg, Karthik C. S., and Ron Safier. Can you solve closest string faster than exhaustive search? In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 3:1–3:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ESA.2023.3.
5. Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5):23:1–23:27, 2008. doi:10.1145/1411509.1411513.
6. David Aldous and Persi Diaconis. Longest increasing subsequences: from patience sorting to the Baik-Deift-Johansson theorem. *Bulletin of the American Mathematical Society*, 36(4):413–432, 1999.

7 Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 2005–2039. SIAM, 2025. doi:10.1137/1.9781611978322.63.

8 Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, and Andreas Hofmeier. On the hardness of maximum rank aggregation problems. *J. Discrete Algorithms*, 31:2–13, 2015. doi:10.1016/J.JDA.2014.10.002.

9 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). *SIAM J. Comput.*, 47(3):1087–1097, 2018. doi:10.1137/15M1053128.

10 Therese Biedl, Franz-Josef Brandenburg, and Xiaotie Deng. On the complexity of crossings in permutations. *Discret. Math.*, 309(7):1813–1823, 2009. doi:10.1016/J.DISC.2007.12.088.

11 Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016. doi:10.1017/CBO9781107446984.

12 Karl Bringmann and Bhaskar Ray Chaudhury. Polyline simplification has cubic complexity. *J. Comput. Geom.*, 11(2):94–130, 2020. doi:10.20382/JOCG.V11I2A5.

13 Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 79–97. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.15.

14 Boris Bukh, Karthik C. S., and Bhargav Narayanan. Applications of random algebraic constructions to hardness of approximation. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 237–244. IEEE, 2021. doi:10.1109/FOCS52979.2021.00032.

15 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 261–270. ACM, 2016. doi:10.1145/2840728.2840746.

16 Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. Approximating the median under the ulam metric. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 761–775. SIAM, 2021. doi:10.1137/1.9781611976465.48.

17 Diptarka Chakraborty, Debarati Das, and Robert Krauthgamer. Clustering permutations: New techniques with streaming applications. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 31:1–31:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.ITCS.2023.31.

18 Diptarka Chakraborty, Syamantak Das, Arindam Khan, and Aditya Subramanian. Fair rank aggregation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/974309ef51ebd89034adc64a57e304f2-Abstract-Conference.html.

19 Diptarka Chakraborty, Kshitij Gajjar, and Agastya Vibhuti Jha. Approximating the center ranking under ulam. In Mikolaj Bojanczyk and Chandra Chekuri, editors, *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021, December 15-17, 2021, Virtual Conference*, volume 213 of *LIPIcs*, pages 12:1–12:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.FSTTCS.2021.12.

20 Graham Cormode, S. Muthukrishnan, and Süleyman Cenk Sahinalp. Permutation editing and matching via embeddings. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *Automata, Languages and Programming, 28th International Colloquium, ICALP 2001, Crete, Greece, July 8-12, 2001, Proceedings*, volume 2076 of *Lecture Notes in Computer Science*, pages 481–492. Springer, 2001. doi:10.1007/3-540-48224-5_40.

21 Roee David, Karthik C. S., and Bundit Laekhanukit. On the complexity of closest pair via polar-pair of point-sets. *SIAM J. Discret. Math.*, 33(1):509–527, 2019. doi:10.1137/17M1128733.

22 Colin de la Higuera and Francisco Casacuberta. Topology of strings: Median string is np-complete. *Theor. Comput. Sci.*, 230(1-2):39–48, 2000. doi:10.1016/S0304-3975(97)00240-5.

23 Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko, editors, *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 613–622. ACM, 2001. doi:10.1145/371920.372165.

24 Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In Alon Y. Halevy, Zachary G. Ives, and AnHai Doan, editors, *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003*, pages 301–312. ACM, 2003. doi:10.1145/872757.872795.

25 Nick Fischer, Elazar Goldenberg, Mursalin Habib, and Karthik C. S. Hardness of median and center in the ulam metric. *CoRR*, abs/2504.16437, 2025. doi:10.48550/arXiv.2504.16437.

26 Moti Frances and Ami Litman. On covering problems of codes. *Theory Comput. Syst.*, 30(2):113–119, 1997. doi:10.1007/S002240000044.

27 Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M. LeProust, Botond Sipos, and Ewan Birney. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nat.*, 494(7435):77–80, 2013. doi:10.1038/NATURE11875.

28 Dan Gusfield. *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press, 1997. doi:10.1017/CBO9780511574931.

29 Donna Harman. Ranking algorithms. In William B. Frakes and Ricardo A. Baeza-Yates, editors, *Information Retrieval: Data Structures & Algorithms*, pages 363–392. Prentice-Hall, 1992.

30 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/JCSS.2001.1774.

31 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

32 Karthik C. S. and Pasin Manurangsi. On closest pair in euclidean metric: Monochromatic is as hard as bichromatic. *Comb.*, 40(4):539–573, 2020. doi:10.1007/S00493-019-4113-1.

33 John G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

34 Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 95–103. ACM, 2007. doi:10.1145/1250790.1250806.

35 Teuvo Kohonen. Median strings. *Pattern Recognit. Lett.*, 3(5):309–313, 1985. doi:10.1016/0167-8655(85)90061-3.

36 J. Kevin Lanctôt, Ming Li, Bin Ma, Shaojiu Wang, and Louxin Zhang. Distinguishing string selection problems. *Inf. Comput.*, 185(1):41–55, 2003. doi:10.1016/S0890-5401(03)00057-9.

37 Xue Li, Xinlei Wang, and Guanghua Xiao. A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications. *Briefings in bioinformatics*, 20(1):178–189, 2019. doi:10.1093/BIB/BBX101.

38 Yuting Liu, Tie-Yan Liu, Tao Qin, Zhiming Ma, and Hang Li. Supervised rank aggregation. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 481–490. ACM, 2007. doi:10.1145/1242572.1242638.

39 Carlos D. Martínez-Hinarejos, Alfons Juan, and Francisco Casacuberta. Use of median string for classification. In *15th International Conference on Pattern Recognition, ICPR'00, Barcelona, Spain, September 3-8, 2000*, pages 2903–2906. IEEE Computer Society, 2000. doi:10.1109/ICPR.2000.906220.

40 François Nicolas and Eric Rivals. Complexities of the centre and median string problems. In Ricardo A. Baeza-Yates, Edgar Chávez, and Maxime Crochemore, editors, *Combinatorial Pattern Matching, 14th Annual Symposium, CPM 2003, Morelia, Michocán, Mexico, June 25-27, 2003, Proceedings*, volume 2676 of *Lecture Notes in Computer Science*, pages 315–327. Springer, 2003. doi:10.1007/3-540-44888-8_23.

41 Samuel E. L. Oliveira, Victor Diniz, Anísio Lacerda, Luiz H. C. Merschmann, and Gisele L. Pappa. Is rank aggregation effective in recommender systems? an experimental analysis. *ACM Trans. Intell. Syst. Technol.*, 11(2):16:1–16:26, 2020. doi:10.1145/3365375.

42 Pavel A. Pevzner. *Computational molecular biology - an algorithmic approach.* MIT Press, 2000.

43 V. Y. Popov. Multiple genome rearrangement by swaps and by element duplications. *Theor. Comput. Sci.*, 385(1-3):115–126, 2007. doi:10.1016/J.TCS.2007.05.029.

44 Cyrus Rashtchian, Konstantin Makarychev, Miklós Z. Rácz, Siena Ang, Djordje Jevdjic, Sergey Yekhanin, Luis Ceze, and Karin Strauss. Clustering billions of reads for DNA data storage. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3360–3371, 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/ab7314887865c4265e896c6e209d1cd6-Abstract.html.

45 Craige Schensted. Longest increasing and decreasing subsequences. *Canadian Journal of mathematics*, 13:179–191, 1961.

46 Anke van Zuylen and David P. Williamson. Deterministic algorithms for rank aggregation and other ranking and clustering problems. In Christos Kaklamanis and Martin Skutella, editors, *Approximation and Online Algorithms, 5th International Workshop, WAOA 2007, Eilat, Israel, October 11-12, 2007. Revised Papers*, volume 4927 of *Lecture Notes in Computer Science*, pages 260–273. Springer, 2007. doi:10.1007/978-3-540-77918-6_21.

47 Tiance Wang, John Sturm, Paul W. Cuff, and Sanjeev R. Kulkarni. Condorcet voting methods avoid the paradoxes of voting theory. In *50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012, Allerton Park & Retreat Center, Monticello, IL, USA, October 1-5, 2012*, pages 201–203. IEEE, 2012. doi:10.1109/ALLERTON.2012.6483218.

48 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. doi:10.1016/J.TCS.2005.09.023.

49 H. Peyton Young. Condorcet's theory of voting. *American Political science review*, 82(4):1231–1244, 1988.

50 H. Peyton Young and Arthur Levenglick. A consistent extension of condorcet's election principle. *SIAM Journal on applied Mathematics*, 35(2):285–300, 1978.