Sorting as Gradient Flow on the Permutohedron

Jonathan Landers
jonathan.robert.landers@gmail.com
Preprint. Submitted to arXiv, April 2025.

ABSTRACT

We investigate how sorting algorithms efficiently overcome the exponential size of the permutation space. Our main contribution is a new continuous-time formulation of sorting as a gradient flow on the permutohedron, yielding an independent proof of the classical $\Omega(n \log n)$ lower bound for comparison-based sorting. This formulation reveals how exponential contraction of disorder occurs under simple geometric dynamics. In support of this analysis, we present algebraic, combinatorial, and geometric perspectives, including decision-tree arguments and linear constraints on the permutohedron. The idea that efficient sorting arises from structure-guided logarithmic reduction offers a unifying lens for how comparisons tame exponential spaces. These observations connect to broader questions in theoretical computer science, such as whether the existence of structure can explain why certain computational problems permit efficient solutions.

1. INTRODUCTION

Consider the classical problem of sorting a list

$$L = [l_1, l_2, \dots, l_n]$$

drawn from a totally ordered set D. A naive approach is to enumerate all $n!$ permutations of L and then select the permutation that is in non-decreasing order:

$$\text{BruteForceSort}(L) = \min_{\pi \in S_n} \{ \pi(L) \mid \pi(L) \text{ is sorted} \}. \tag{1}$$

This method has an obvious time complexity of $O(n \cdot n!)$, as each of the $n!$ permutations requires $O(n)$ time to check for order. Although straightforward, such exponential complexity becomes infeasible even for moderately sized lists.

In contrast, classical sorting algorithms (such as MergeSort, QuickSort, and HeapSort) operate within $O(n \log n)$ time by exploiting inherent order structure to prune the search among all $n!$ possibilities. This striking efficiency has long inspired curiosity about the deep connection between structure and computational tractability. Foundational work by Knuth [1], Shannon [2], and others, along with recent geometric and combinatorial perspectives [3, 4, 5], reflect this ongoing effort to understand how structure enables efficient computation.

The main innovation presented in this paper is a new continuous-time formulation of sorting as a gradient flow on the permutohedron. This framework yields an independent geometric proof of the classical $\Omega(n \log n)$ lower bound for comparison-based sorting. By interpreting comparisons as projections along a smooth descent, we recast sorting as a process of exponential contraction in a continuous geometric space. This viewpoint complements existing combinatorial and algebraic arguments by offering a dynamical, intuitive explanation for why efficient sorting is possible.

To organize the discussion, we emphasize that efficient algorithms shrink factorial search spaces through structure-guided logarithmic narrowing. While not the primary focus, the concept provides a helpful lens for understanding how sorting problems can admit efficient solutions through principled decomposition.

2. STRUCTURED REDUCTION OF THE PERMUTATION SPACE

Let $\Omega = S_n$ denote the set of all permutations of n elements. An efficient sorting algorithm can be viewed as a function

$$\mathcal{A} : D^n \to D^n_{\text{sorted}},$$

which achieves sorting without exhaustively exploring the entirety of Ω. Instead, $\mathcal{A}$ operates by partitioning Ω into a collection of smaller, structured subfamilies. Formally, suppose there exists a collection $\mathcal{F} \subset 2^{\Omega}$ with the following properties:

- **Covering:**
$$\bigcup_{F \in \mathcal{F}} F \supset \{\pi_{\text{sorted}}\},$$
so that the correctly sorted permutation is contained in the union.
- **Structured Factorization:** Each $F \in \mathcal{F}$ satisfies
$$|F| \ll n!, \quad \text{and in particular,} \quad \log_2 |F| = O(n \log n).$$
- **Logarithmic Reduction:** The decision procedure employed by $\mathcal{A}$ uses only $O(\log |F|)$ comparisons per subfamily. The total over recursive levels aggregates to $O(n \log n)$ comparisons.

This framework of structure-guided logarithmic reduction helps explain how efficient sorting algorithms narrow the solution space. By organizing permutations into recursively searchable subfamilies, algorithms can reduce factorial complexity to the optimal $O(n \log n)$ regime. Later sections will revisit these ideas through a geometric and continuous lens, showing how such structure arises naturally in both discrete and continuous formulations. This highlights a broader theme: the existence of exploitable structure may be the key to why a subset of problems admit efficient solutions, despite their exponential naïve formulations.

A related geometric viewpoint is developed by Blelloch and Dobson, who connect comparison-based sorting to offline binary search trees (BSTs) using a planar geometric representation of permutations and search sequences. Their notion of a log-interleave bound quantifies permutation complexity in this setting, offering a visual and information-theoretic measure of sorting cost [3]. Whereas their framework is tied to the geometry of binary-search-tree permutations, our perspective extends the same contraction principle to any constraint-driven partition of the permutohedron, highlighting how geometric structure underlies a wide class of efficient algorithms.

3. ALGEBRAIC TREATMENT: DECISION-TREE LOWER BOUND

We first establish the lower bound for any comparison-based sorting algorithm by modeling the process as a binary decision tree.

Lemma 3.1 (Decision-Tree Lower Bound)**.** *Let T be a binary decision tree that correctly sorts n elements using comparisons. Then the height h of T satisfies*

$$h \geq \log_2(n!). \tag{2}$$

Proof. A deterministic comparison-based algorithm must yield a different sequence of comparisons for each of the $n!$ possible orderings of the input. Consequently, the decision tree must have at least $n!$ leaves. Because any binary tree of height h has at most 2^h leaves, we obtain:

$$2^h \geq n!.$$

Taking the base-2 logarithm of both sides yields:

$$h \geq \log_2(n!).$$

Using Stirling's approximation,

$$\log_2(n!) = n \log_2 n - O(n), \tag{3}$$

so that $h = \Omega(n \log n)$. This lower bound is tight for comparison-based sorting algorithms. □

Interestingly, Harris, Kretschmann, and Mori [4] have shown that the aggregate number of leaf-to-root comparisons in all $n!$ decision-trees of size n coincides with the number of parking preference lists admitting exactly $n - 1$ lucky cars, providing an alternative combinatorial proof of equation (3).

Figure 1 illustrates the principle with a complete binary decision tree for sorting three elements. The structure shows how $\lceil \log_2(3!) \rceil = 3$ comparisons suffice to isolate the correct permutation, aligning with the derived lower bound. The interpretation is that in order to reduce the exponential number of permutation possibilities to one unique sorted order, the algorithm must perform at least $\Omega(n \log n)$ comparisons. A sorting algorithm meeting this bound must, implicitly, group many permutations into structured subsets, each identified via a small number of comparisons. Knuth's seminal work established this fundamental lower bound through a comprehensive analysis of sorting algorithms and their information-theoretic constraints [1]. This reasoning reflects why exponential lower bounds exist for general search problems and why finding polynomial-time solutions requires leveraging structural properties of the solution space.

4. GEOMETRIC PERSPECTIVE: THE PERMUTOHEDRON AND SEQUENCE OF CONSTRAINTS

We now turn to a geometric interpretation by considering the permutohedron and demonstrate how a sequence of linear constraints (each corresponding to a comparison) isolates the sorted order.

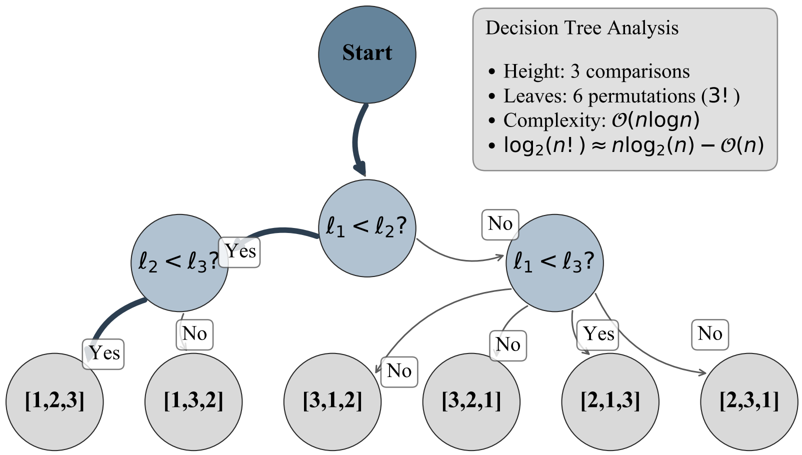

Figure 1: Binary decision tree representing all comparison-based paths for sorting three elements. Each internal node encodes a binary comparison, and each leaf corresponds to one of the six permutations in S_3. The highlighted path traces the sorted permutation $[1, 2, 3]$, requiring only three comparisons— the information-theoretic minimum $\lceil \log_2(3!) \rceil = 3$. This structure exemplifies recursive logarithmic reduction, where each comparison halves the remaining search space and reduces factorial complexity to the optimal $O(nlogn)$ regime.

Definition 4.1 (Permutohedron)**.** Let

$$v = (1, 2, \ldots, n) \in \mathbb{R}^n.$$

The permutohedron $\mathcal{P}_n$ is defined as

$$\mathcal{P}_n = \mathrm{conv}\{\pi(v) \mid \pi \in S_n\}. \tag{4}$$

Proposition 4.2 (Dimensionality and Affine Containment)**.** *The permutohedron $\mathcal{P}_n$ is an $(n-1)$-dimensional polytope contained in the affine hyperplane*

$$H = \left\{ x \in \mathbb{R}^n \,\middle|\, \sum_{i=1}^{n} x_i = \frac{n(n+1)}{2} \right\}. \tag{5}$$

Proof. For any permutation $\pi \in S_n$,

$$\sum_{i=1}^{n} (\pi(v))_i = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

Thus, every vertex of $\mathcal{P}_n$ lies in the hyperplane H. Since the vertices span H, and H has dimension $n-1$, it follows that $\mathcal{P}_n$ is an $(n-1)$-dimensional polytope. □

Theorem 4.3 (Geometric Contraction via Linear Constraints)**.** *Consider a sequence of linear inequalities of the form*

$$x_i < x_j,$$

selected according to an algorithm's comparisons, that is satisfied by the sorted order $(1, 2, \ldots, n)$. *Then there exists a set of at most* $C \cdot n \log n$ *such independent constraints, for some constant* C, *whose intersection with* $\mathcal{P}_n$ *is exactly the vertex corresponding to the sorted order.*

Proof. Each comparison $l_i < l_j$ imposes the inequality $x_i < x_j$, which corresponds to an affine half-space. Let S be the set of such constraints made during sorting. Define

$$\mathcal{P}_n(S) = \mathcal{P}_n \cap \bigcap_{(i,j)\in S} \{x \in \mathbb{R}^n \mid x_i < x_j\}. \tag{6}$$

The vertex $(1, 2, \ldots, n)$ satisfies all constraints $x_1 < x_2 < \cdots < x_n$.

Suppose each comparison halves the space of feasible permutations. Then after k comparisons:

$$\frac{n!}{2^k} \leq 1 \quad \Rightarrow \quad 2^k \geq n!.$$

Taking logarithms:

$$k \geq \log_2(n!) \approx n \log_2 n - O(n).$$

So at least $\Omega(n \log n)$ constraints are needed.

Classical algorithms achieve $O(n \log n)$ comparisons. Hence, their comparison constraints define a polytope intersecting $\mathcal{P}_n$ in exactly one vertex, the sorted order. □

Figure 2 illustrates this process geometrically. Each comparison corresponds to a linear constraint that incrementally reduces the feasible region of the permutohedron, ultimately isolating the sorted permutation as the unique solution. Ziegler's influential work on polytopes provides the geometric foundations necessary for understanding how permutohedra represent the combinatorial structure of sorting problems [5]. The contraction of solution space via linear inequalities illustrates how structural constraints can reduce complexity and, in some cases, enable polynomial-time solutions. Goemans further formalized this idea by showing that sorting networks yield extended formulations of the permutohedron with $\Theta(n \log n)$ facets, reinforcing the role of comparisons as geometric constraints [6].

A related geometric approach is explored by Lee et al., who study stack-sorting algorithms via subpolytopes of the permutohedron called stack-sorting simplices [7]. These structures capture how the stack-sort process traverses faces and edges of the permutohedron to identify sortable permutations. Their work provides a detailed case study of structured decomposition within a constrained algorithm. In contrast, the present formulation considers general comparison sorts and treats comparisons as half-space constraints that carve through the full permutohedron. This broader perspective builds on such algorithm-specific insights to describe how constraint-based descent efficiently isolates the sorted order across all sorting procedures.

5. CONTINUOUS GRADIENT FLOW AND THE $\Omega(n \log n)$ LOWER BOUND

In this section we present our main contribution: a new continuous-time formulation of sorting as a gradient flow on the permutohedron. This formulation not only offers novel geometric insight into the exponential contraction of disorder under simple dynamics, but also yields an independent proof of the classical $\Omega(n \log n)$ lower bound for comparison-based sorting algorithms. In the broader context of this paper, this perspective complements our algebraic and geometric analyses and highlights how recursive logarithmic traversal enables efficient navigation of exponential spaces.

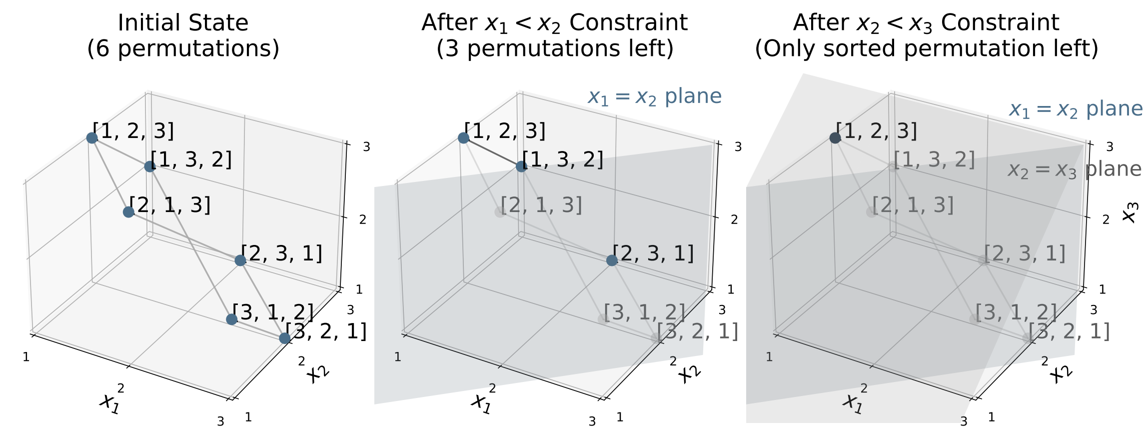

Figure 2: Geometric interpretation of sorting as structured contraction of the permutohedron $\mathcal{P}_3$, the convex hull of all permutations of $(1, 2, 3)$. Each vertex corresponds to a permutation, and edges connect permutations that differ by adjacent swaps. The panels illustrate how successive linear constraints—$x_1 < x_2$ and $x_2 < x_3$—shrink the feasible region of $\mathcal{P}_3$, ultimately isolating the sorted permutation. This sequence illustrates recursive logarithmic reduction, with comparisons acting as hyperplane cuts that halve the solution space, reducing factorial complexity to the optimal $O(n \log n)$.

5.1 Setup and Definitions

Let

$$v_s = (1, 2, \ldots, n) \in \mathbb{R}^n$$

denote the vertex of the permutohedron corresponding to the sorted permutation. To measure the disorder of any point $x \in \mathbb{R}^n$, define the potential

$$V(x) = \frac{1}{2}\|x - v_s\|^2 = \frac{1}{2}\sum_{i=1}^{n}(x_i - i)^2. \tag{7}$$

Note that $V(v_s) = 0$ and $V(x) > 0$ for $x \neq v_s$.

5.2 Main Theorem and Proof

Theorem 5.1. *Let $x(0) \in \mathbb{R}^n$ be an initial state corresponding to an unsorted permutation and $v_s = (1, 2, \ldots, n)$ the sorted vertex of the permutohedron P_n. Under the gradient flow*

$$\dot{x}(t) = -\nabla V(x(t)) = v_s - x(t), \tag{8}$$

the following statements hold:

1. *The solution satisfies*

$$x(t) = v_s + \big(x(0) - v_s\big)e^{-t}, \tag{9}$$

$$\|x(t) - v_s\|^2 = \|x(0) - v_s\|^2 e^{-2t}. \tag{10}$$

2. *For a fixed threshold ϵ with $\epsilon^2 = \Theta(1)$ defining an effectively sorted state (i.e., $V(x(t)) \le \epsilon^2$), it follows that*

$$t \ge \frac{1}{2}\log\Big(\frac{\|x(0) - v_s\|^2}{\epsilon^2}\Big). \tag{11}$$

In the worst-case scenario (e.g., the reverse permutation) we have $\|x(0) - v_s\|^2 = \Theta(n^3)$ and thus $t = \Theta(\log n)$.

3. *If we interpret each discrete comparison or swap as advancing time by an order of $\Theta(1/n)$, then the total number T of discrete operations required satisfies*

$$T \sim n \cdot t = \Theta(n \log n).$$

Proof. We begin by deriving the gradient flow dynamics directly from the potential V. A straightforward computation shows that

$$\nabla V(x) = \big(x_1 - 1,\ x_2 - 2,\ \ldots,\ x_n - n\big).$$

Hence, the gradient flow is given by

$$\dot{x}(t) = -\nabla V(x(t)) = \big(1 - x_1(t),\ 2 - x_2(t),\ \ldots,\ n - x_n(t)\big).$$

Because the system decouples, each coordinate x_i satisfies the first-order linear differential equation

$$\dot{x}_i(t) = i - x_i(t),$$

with initial condition $x_i(0)$. Its solution is

$$x_i(t) = i + \big(x_i(0) - i\big)e^{-t}.$$

Collecting these coordinate solutions, we obtain

$$x(t) = v_s + \big(x(0) - v_s\big)e^{-t},$$

so that

$$\|x(t) - v_s\|^2 = \|x(0) - v_s\|^2 e^{-2t}.$$

This establishes statement 1.

For statement 2, define the *initial disorder* $D_0 = \|x(0) - v_s\|^2$. The system is effectively sorted when

$$\|x(t) - v_s\|^2 = D_0 e^{-2t} \le \epsilon^2.$$

Taking logarithms yields

$$t \ge \frac{1}{2}\log\Big(\frac{D_0}{\epsilon^2}\Big).$$

Reverse-permutation radius If $x(0) = (n, n-1, \ldots, 1)$ then $x_i(0) - i = n + 1 - 2i$. A direct summation shows

$$D_0 = \sum_{i=1}^{n}\big(n + 1 - 2i\big)^2 = 2\sum_{k=1}^{\lfloor n/2 \rfloor}(2k-1)^2 = \frac{n(n^2-1)}{3} = \Theta(n^3). \tag{12}$$

Hence $t = \Theta(\log n)$.

For statement 3, if the continuous time t is distributed over T discrete steps, each comparison advancing time by $\Theta(1/n)$, then

$$t \sim \frac{T}{n}, \quad \text{so} \quad T \sim n \cdot t = \Theta(n \log n).$$

□

5.3 Discrete-Time Correspondence

The passage from continuous dynamics to discrete comparisons can be made precise.

Lemma 5.2. *Let Δt be the maximum time advance attributable to a single comparison. If each comparison resolves at most one inversion, then $\Delta t \le c/n$ for some constant $c > 0$; consequently*

$$T \;\ge\; \frac{t}{\Delta t} \;\ge\; \frac{n}{c} \cdot \frac{1}{2} \log\!\left(\frac{D_0}{\epsilon^2}\right). \tag{13}$$

In the worst case $D_0 = \Theta(n^3)$, whence $T = \Omega(n \log n)$.

Proof. There are $\Theta(n^2)$ inversions in the worst case. Exponential decay $\|x(t) - v_s\|^2 \propto e^{-2t}$ eliminates a constant fraction of inversions every $\Theta(1)$ time units. Thus a single inversion cannot correspond to more than c/n continuous time, establishing the bound on Δt. The stated inequality follows by substitution of t from Theorem 5.1. □

5.4 Worked Example ($n = 3$)

Consider the reverse permutation $x(0) = (3, 2, 1)$, for which $D_0 = \|x(0) - v_s\|^2 = 8$. The solution of $\dot{x} = v_s - x$ is

$$x(t) = (1, 2, 3) + (2, 0, -2)\, e^{-t}.$$

Boundary 1. The trajectory meets the face $x_1 = x_2$ at $t_1 = \log 2$. Projecting onto this face exchanges the first two coordinates, yielding the permutation $(2, 3, 1)$.

Boundary 2. Continuing the flow, we encounter $x_2(t) = x_3(t)$ at $t_2 = t_1 + \log 2$. The associated projection gives the permutation $(2, 1, 3)$.

Boundary 3. After another interval of length $\log 2$ the trajectory again reaches $x_1 = x_2$; the projection now delivers the sorted state $(1, 2, 3)$.

Thus the total continuous time is $t = \frac{3}{2} \log 2$. Applying Lemma 5.2 with $\Delta t = 1/3$ yields

$$T = \frac{t}{\Delta t} = 3 \quad \text{comparisons,}$$

matching the information-theoretic minimum $\lceil \log_2 3! \rceil = 3$.

5.5 Incorporating the Permutohedron Constraints

Although the above derivation considers the dynamics in $\mathbb{R}^n$, the actual sorting process is constrained to the vertices of the permutohedron P_n. A rigorous approach requires one to consider the projected gradient flow

$$\dot{x}(t) = \Pi_{T_{P_n}(x(t))}\Big[-\nabla V(x(t))\Big], \tag{14}$$

where $\Pi_{T_{P_n}(x(t))}$ denotes the projection onto the tangent cone to P_n at $x(t)$. In regions where the flow lies within the interior of a face, the dynamics mimic those of the unconstrained system. When a boundary (where $x_i = x_j$) is encountered, the projection reflects the resolution of an inversion, akin to a hyperplane cut in the discrete setting. Because the outward normal at any boundary point is orthogonal to $\Pi_{T_{P_n}}\big[-\nabla V\big]$, we still obtain $\frac{d}{dt}V(x(t)) \le -2V(x(t))$, so the exponential contraction

$$\|x(t) - v_s\|^2 = \|x(0) - v_s\|^2 e^{-2t}$$

remains valid. The well-posedness of this projected flow follows from classical results on projection onto convex tangent cones (see [8]).

5.6 Remarks

Theorem 5.1 demonstrates that continuous gradient descent under the potential $V(x) = \frac{1}{2}\|x - v_s\|^2$ captures not only the exponential contraction of disorder but also yields a lower bound of $\Omega(n \log n)$ discrete operations. Lemma 5.2 bridges the continuous and discrete viewpoints, showing that a microscopic time step of $\Theta(1/n)$ per comparison is both necessary and sufficient to match the information-theoretic bound.

By interpreting each comparison as an infinitesimal projection along this descent, we obtain a continuous geometric derivation of the classical sorting complexity bound. This mirrors the insight of Jordan, Kinderlehrer, and Otto, who reinterpreted the Fokker–Planck equation as a gradient flow minimizing a free energy functional under the Wasserstein metric, offering a variational foundation for diffusion dynamics [9].

This synthesis of continuous dynamics with the combinatorial structure of P_n offers a fresh perspective on sorting: rather than operating solely in the discrete domain, efficient sorting algorithms can be viewed as navigating a continuous geometric landscape in which disorder decays smoothly and predictably. In this sense, comparisons act as discretized steps along a natural gradient toward order.

Figure 3 visually illustrates this process. Vertices correspond to permutations, and the gradient flow traces continuous trajectories across the edges of the permutohedron, converging toward the sorted vertex. For example, beginning at the vertex $(3, 2, 1)$, the flow descends via adjacent edges to $(1, 2, 3)$, resolving inversions in a manner analogous to classical swap-based sorting. The figure thus provides a geometric intuition for how sorting can be viewed as an energy-minimizing process, one that unfolds over a combinatorial polytope with surprising smoothness. A similar mathematical structure appears in information geometry, where descent processes minimize divergence-based potentials over curved statistical manifolds, as developed by Amari [10].

This viewpoint may inspire analogous approaches in other domains where seemingly discrete or combinatorial problems conceal latent continuous structure. The ability to recast such problems in geometric or dynamical terms can reveal not only new proofs but deeper unifying principles governing efficiency and complexity.

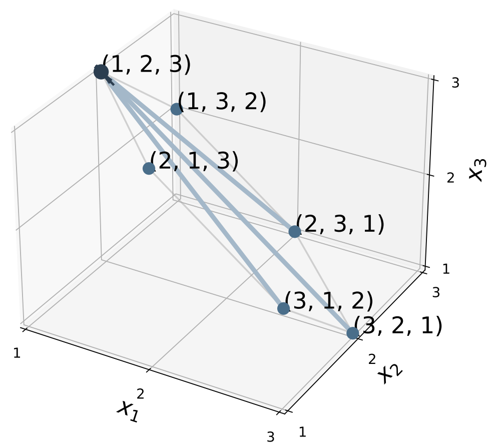

Figure 3: Gradient flow on the permutohedron illustrating exponential contraction toward the sorted permutation. Vertices correspond to unique permutations, with intermediate lines depicting trajectories of gradient descent driven by the disorder potential $V(x) = \frac{1}{2}\|x - v_s\|^2$. The highlighted paths demonstrate rapid convergence from initial unsorted states via exponential decay of disorder, linking continuous geometric dynamics directly to discrete sorting steps. For instance, starting at vertex $(3, 2, 1)$, the gradient flow moves continuously along adjacent edges toward the sorted vertex $(1, 2, 3)$, resolving inversions step-by-step. This visualization highlights the geometric foundations underlying the classical $\Omega(n \log n)$ sorting complexity bound established in Theorem 5.1.

6. CONCLUSION

The central challenge in efficient sorting is overcoming the exponential size of the permutation space. Classical algorithms achieve this by exploiting inherent structural constraints. For example, MergeSort uses the recurrence

$$T(n) = 2T(n/2) + O(n)$$

to obtain an $O(n \log n)$ running time via divide-and-conquer, while Von Neumann's pioneering implementation demonstrated how recursive design could yield both theoretical and practical efficiency [11]. Similarly, QuickSort uses pivot selection to probabilistically prune the search space with an expected runtime of $\mathbb{E}[T(n)] = O(n \log n)$, and HeapSort restricts orderings via a heap structure to maintain the same complexity. Together, these examples epitomize how recursive constraints and localized comparisons reduce the overwhelming $n!$ possibilities to a manageable process.

At each recursive step, the algorithm partitions the permutation space according to

$$|\Omega_{\text{subproblem}}| \leq \prod_{i=1}^{m} |\Omega_i|, \quad \text{with} \quad \sum_{i=1}^{m} n_i = n.$$

Taking logarithms, we have

$$\log_2\left(\prod_{i=1}^{m}|\Omega_i|\right) = \sum_{i=1}^{m}\log_2|\Omega_i| \approx \sum_{i=1}^{m} n_i\log_2(n_i),$$

so that the overall work sums to $O(n\log n)$. Hopcroft and Tarjan formalized this approach by showing that reducing combinatorial complexity via structured decomposition is a powerful method for making exponential problems tractable [12]. In this light, recursive constraints literally "collapse" the search space from size $n!$ to a polynomial number of possibilities.

Recent combinatorial analysis by Harris, Kretschmann, and Mori further underscores the hidden structure exploited by sorting algorithms [4]. Their work reveals that the number of comparisons performed by QuickSort over all permutations corresponds to the number of parking preference lists with $n-1$ "lucky cars." This surprising enumerative result links QuickSort's partitioning strategy to the geometry of the permutohedron, a connection that our framework generalizes. By modeling comparisons as linear constraints that slice through the permutohedron, we show that every efficient sorting algorithm leverages internal structure to descend through an exponential space. This geometric viewpoint not only broadens our understanding of sorting but also connects it to broader complexity-theoretic ideas.

In parallel, we have introduced a novel continuous-time perspective by modeling sorting as a gradient flow over the permutohedron. This formulation independently recovers the classical $\Omega(n\log n)$ lower bound, offering an intuitive geometric insight: each discrete comparison corresponds to a projection along a natural descent path that smooths out disorder. As the gradient flow contracts the distance to the sorted order according to

$$\|x(t)-v_s\|^2 = \|x(0)-v_s\|^2 e^{-2t},$$

we obtain an alternative derivation of the optimal discrete complexity when interpreting each infinitesimal time step as an elementary comparison. This reinforces the idea that efficient sorting is not purely combinatorial but also inherently geometric and dynamical in nature.

Furthermore, the method extends naturally to sorting networks, external-memory models, and parallel computation by adapting the underlying structure $\mathcal{F}$ and the associated cost metrics. The persistent theme, regardless of the specific model, is that structured reduction of the permutation group reduces the exponential complexity, a perspective supported by Shannon's information-theoretic analysis [2].

Ultimately, the synthesis of combinatorial, geometric, and continuous viewpoints invites continued reflection on one of the most enduring mysteries in computation: why do some problems yield to structure and others remain intractable? This perspective, echoing central themes in complexity theory [13], challenges the notion that an exponential search space necessarily implies NP-hardness. Mann's clarifying analysis [14] exemplifies this point by showing how tractability can emerge from structural insights—as illustrated by Dijkstra's algorithm in the realm of shortest paths. In our work, modeling comparisons as linear constraints that geometrically collapse the permutation polytope concretizes this idea.

In this broader context, recursive logarithmic pruning of structure serves as a unifying lens for sorting algorithms, capturing how inherent constraints progressively narrow an exponential search space. This perspective invites curiosity about whether other computational problems may conceal analogous structures - latent "permutohedra" whose facets encode tractable subsets of an otherwise exponential complexity.

REFERENCES

[1] Donald E. Knuth. *The art of computer programming, volume 3: sorting and searching*. Addison-Wesley, Reading, MA, 1973.

[2] Claude E. Shannon. A mathematical theory of communication. *Bell Syst Tech J*, 27:379–423, 623–656, 1948.

[3] Guy E. Blelloch and Magdalen Dobson. The geometry of tree-based sorting. In *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *LIPIcs*, pages 26:1–26:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[4] Pamela E. Harris, Jan Kretschmann, and J. Carlos Martínez Mori. "Lucky cars" and the quicksort algorithm. *The American Mathematical Monthly*, 131(5):417–423, 2024.

[5] Günter M. Ziegler. *Lectures on polytopes*. Springer-Verlag, Berlin, 1995.

[6] Michel X. Goemans. Smallest compact formulation for the permutahedron. *Math Program Ser B*, 153(1):5–11, 2015.

[7] Eon Lee, Carson Mitchell, and Andrés R. Vindas-Meléndez. Stack-sorting simplices: geometry and lattice-point enumeration. In *Proceedings of Combinatorics, Graph Theory and Computing (CGTC 2025)*, Springer Proc. Math. Stat., Cham, Switzerland, 2025. Springer. Forthcoming.

[8] R. Tyrrell Rockafellar. *Convex analysis*. Princeton University Press, Princeton, NJ, 1970.

[9] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the Fokker–Planck equation. *SIAM J Math Anal*, 29(1):1–17, 1998.

[10] Shun-ichi Amari. *Information geometry and its applications*. Springer, Tokyo, 2016.

[11] John von Neumann. First draft of a report on the EDVAC. Technical report, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, PA, 1945.

[12] John E. Hopcroft and Robert E. Tarjan. Efficient algorithms for graph manipulation. *Commun ACM*, 16(6):372–378, 1973.

[13] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, New York, 2009.

[14] Zoltán Mann. The top eight misconceptions about NP-hardness. *Computer*, 50(5):72–79, 2017.