COMPLEXITY BOUNDS FOR SEARCH PROBLEMS

by

Nicholas Recker

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Computer Science

West Lafayette, Indiana

May 2024

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Simina Brânzei, Chair
Department of Computer Science

Dr. Mikhail Atallah
Department of Computer Science

Dr. Kristoffer Hansen
Department of Computer Science, Aarhus University

Dr. Alex Pothen
Department of Computer Science

Approved by:
Dr. Kihong Park

ACKNOWLEDGMENTS

I would like to thank my advisor, Simina Branzei, as well as my collaborators Dimitris Paparas and Davin Choo, without whom the works covered in this thesis would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Search problems lie at the core of computer science. This thesis addresses two types of search problems. The first type is local search on graphs. In this problem there is a graph $G = (V, E)$ and a function $f : V \to \mathbb{R}$. The goal is to find a *local minimum* of f; i.e. a vertex v such that $f(v) \leq f(u)$ for all $(u, v) \in E$. This problem is relevant as a heuristic in a variety of fields where it is not computationally feasible to compute global minima. For example, many machine learning techniques compute a local minimum of loss.

The second type of search problem we consider is extremely fundamental: search for an element in an array. We consider this both when the array is sorted and unsorted, but in the context of computation in rounds. In each of a limited number of rounds some queries are issued, but the answers are not revealed until after the last query of the round has been issued. This model is relevant to parallel computing, where the work done by one machine is not available to other machines operating concurrently, effectively allowing one query per machine at any given time. Following this thread, we also study sorting in rounds and the connection this has to cake cutting in rounds, a form of fair division.

In studying these problems we focus primarily on proving *lower* bounds on their query complexities. While upper bounds can be shown simply by designing an algorithm that achieves that bound, it is often difficult to show a matching lower bound.

For local search, we use *relational adversary* methods to overcome this difficulty. Relational adversaries were first developed by Ambainis [1] for use in bounding the query complexities of quantum algorithms. We use the strong weighted version of the quantum relational adversary, developed by Zhang [2], for our quantum results. Aaronson [3] developed a classical variant of the relational adversary and applied it to local search on the hypercube and grid. We improve on his classical relational adversary and use our variant for all our classical results on local search.

Most classical lower bound techniques for randomized algorithms use Yao's minimax principle [4], either explicitly or implicitly through methods like the relational adversary. For

search problems on arrays, we show sharper bounds than Yao's minimax principle can show. We accomplish this via more direct proofs by induction and through showing polynomial inequalities.

The specific papers covered in this thesis are the following:

1. "The Sharp Power Law of Local Search on Expanders" [5], Simina Brânzei, Davin Choo, and Nicholas Recker. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, https://arxiv.org/abs/2305.08269, 2024.
2. "Spectral Lower Bounds for Local Search" [6], Simina Brânzei and Nicholas Recker. arXiv preprint arXiv:2403.06248 (2024).
3. "Searching Sorting and Cake Cutting in Rounds" [7], Simina Brânzei, Dimitris Paparas, and Nicholas Recker. arXiv preprint arXiv:2012.00738 (2020).

In addition, this thesis covers unpublished results on the quantum query complexity of local search on expanders.

1. THE SHARP POWER LAW OF LOCAL SEARCH ON EXPANDERS

This chapter is based on my paper of the same name, which can be found at https://arxiv.org/abs/2305.08269.

1.1 Introduction

Local search is a powerful heuristic for solving hard optimization problems, which works by starting with an initial solution to a problem and then iteratively improving it. Its simplicity and ability to handle large and complex search spaces make it a useful tool in a wide range of fields, including computer science, engineering, optimization, economics, and finance.

The complexity of local search has been extensively studied in both the white box model (see, e.g., [8]) and the black box model (see, e.g., [9]). The latter type of complexity, also known as query complexity, is well understood when the neighbourhood structure of the underlying graph is the Boolean hypercube or the d-dimensional grid, but much less is known for general graphs.

Many optimization techniques rely on gradient-based methods. The speed at which gradient methods find a stationary point of a function can be estimated by analyzing the complexity of local search on the corresponding discretized space. Constructions for analyzing the hardness of computing stationary points are often similar to those for local search, modulo handling the smoothness of the function (see, e.g., [10]). Meanwhile, the difficulty of local search itself is strongly related to the neighbourhood structure of the underlying graph. At one extreme, local search on a line graph on n nodes is easy and can be solved via binary search in $O(\log n)$ queries. At the other extreme, local search on a clique on n nodes takes $\Omega(n)$ queries, thus requiring brute force.

In this paper, we consider the following high level question: *How does the geometry of the graph influence the complexity of local search?* In general, the neighbourhood graph search structure in optimization settings may correspond to more general graphs beyond the well-studied Boolean hypercubes and d-dimensional grids. For example, when the data in low rank matrix estimation is subjected to adversarial corruptions, it is helpful to consider the function on a Riemannian manifold rather than Euclidean space. That is, the discretization of an optimization search space may not necessarily always correspond to some d-dimensional grid. Multiple works consider optimization in non-Euclidean spaces, such as that of [11], which adapts stochastic gradient descent to work on Riemannian manifolds. See [12] and [13] for more discussion.

Our paper tackles the challenge of understanding local search on general graphs and obtains several new results by considering a broader framework of graph features such as vertex congestion and separation number. A corollary is a lower bound of the right order for expanders with constant degree.

Our methodology is strongly inspired by, and can be seen as a variant of, the relational adversary method of [3]. However, where Aaronson's method focuses on the contribution of a query towards distinguishing two inputs from all others, our method considers the aggregate impact of a query across many inputs at once. This allows our method to be asymptotically at least as strong as the version in [3] for all randomized algorithms, as well as strictly stronger on some problems and easier to apply in our setting. This strength also comes at a cost: we get results for randomized algorithms, whilst Aaronson's method works in quantum settings.

Roadmap to the paper.

The model is defined in Section 1.2. Our contributions are given in Section 1.3. Related work is discussed in Section 1.4. Our variant of the relational adversary method is stated together with an example and discussion in Section 1.5 (with full material in Section 1.9).

Lower bounds via vertex congestion, as well as the corollary for expanders, are given in Section 1.6 (with full material in Section 1.11). Lower bounds via the separation number are in Section 1.7 (with full material in Section 1.12).

Finally, Section 1.8 reviews some known results from prior work that we use. Section 1.10 provides the proof for a lemma used throughout the paper.

1.2 Model

Let $G = (V, E)$ be a connected undirected graph and $f : V \to \mathbb{R}$ a function defined on the vertices. A vertex $v \in V$ is a local minimum if $f(v) \leq f(u)$ for all $\{u, v\} \in E$. We will write $V = [n] = \{1, \ldots, n\}$.

Given as input a graph G and oracle access to function f, the local search problem is to find a local minimum of f on G using as few queries as possible. Each query is of the form: "Given a vertex v, what is $f(v)$?".

Query complexity.

The *deterministic query complexity* of a task is the total number of queries necessary and sufficient for a correct deterministic algorithm to find a solution. The *randomized query complexity* is the expected number of queries required to find a solution with probability at least $9/10$ for each input, where the expectation is taken over the coin tosses of the protocol.

Congestion.

Let $\mathcal{P} = \{P^{u,v}\}_{u,v \in V}$ be an all-pairs set of paths in G, where $P^{u,v}$ is a path from u to v. For convenience, we assume $P^{u,u} = (u)$ for all $u \in V$; our results will hold even if $P^{u,u} = \emptyset$.

For a path $Q = (v_1, \ldots, v_s)$ in G, let c_v^Q be the number of times a vertex $v \in V$ appears in Q and c_{e}^Q the number of times an edge $\mathrm{e} \in E$ appears in Q. The *vertex congestion* of the set of paths $\mathcal{P}$ is $\max_{v \in V} \sum_{Q \in \mathcal{P}} c_v^Q$, while the *edge congestion* of $\mathcal{P}$ is $\max_{\mathrm{e} \in E} \sum_{Q \in \mathcal{P}} c_{\mathrm{e}}^Q$.

The *vertex congestion of* G is the smallest integer g for which the graph has an all-pairs set of paths $\mathcal{P}$ with vertex congestion g. Clearly, $g \geq n$ since each vertex belongs to at least n paths in $\mathcal{P}$ and $g \leq n^2$ since each vertex appears at most once on each path and there are n^2 paths in $\mathcal{P}$. The *edge congestion* g_{e} is similarly defined, but with respect to the edge congestion of a set of paths $\mathcal{P}$.

Separation number.

For each subset of vertices $A \subseteq V$, let $\delta(A) \subseteq V \setminus A$ be the set of vertices outside A and adjacent to vertices in A. The *separation number* s of G (see, e.g., [14]) is [1]:

$$s = \max_{H \subseteq V} \min_{\substack{A \subset H: \\ |H|/4 \leq |A| \leq 3|H|/4}} |\delta(A)| \ .$$

d**-regular expanders.**

For each set of vertices $S \subseteq V$, the edges with one endpoint in S and another in $V \setminus S$ are called *cut edges* and denoted $E(S, V \setminus S) = \{(u,v) \in E \mid u \in S, v \notin S\}$. The graph is a β-expander if $|E(S, V \setminus S)| \geq \beta \cdot |S|$, for all $S \subseteq V$ with $0 < |S| \leq n/2$ (see, e.g. [15]). The graph is *d-regular* if each vertex has degree d.

Distance.

For each $u, v \in V$, let $dist(u,v)$ be the length of the shortest path from u to v.

[1]↑For example, the separation number of a barbell graph (i.e., two cliques of size $n/2$ connected by a single edge) is $n/8$, since the maximization will choose H to be one clique of size $n/2 = 4n/8$ and the minimization will choose A to be an arbitrary subset of H of size $3n/8$; then $\delta(A)$ is the rest of the clique of size $4n/8 - 3n/8 = n/8$.

1.3 Our contributions

Guided by the high level question of understanding how graph geometry influences hardness of local search, we obtain the following results.

1.3.1 Our variant of the relational adversary method

Our first contribution is to design a new variant of the relational adversary method of [3]. While [3] relates the query complexity to the progress made on pairs of inputs, we relate the query complexity to progress made on subsets of inputs via a different expression. The precise statement of our variant is as follows:

Theorem 1.3.1. *Consider finite sets A and B, a set $\mathcal{X} \subseteq B^A$ of functions* [2]*, and a map $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ which assigns a label to each function in $\mathcal{X}$. Additionally, we get oracle access to an unknown function $F^* \in \mathcal{X}$. The problem is to compute $\mathcal{H}(F^*)$ using as few queries to F^* as possible.*[3]

Let $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a non-zero symmetric function of our choice with $r(F_1, F_2) = 0$ whenever $\mathcal{H}(F_1) = \mathcal{H}(F_2)$. For each $\mathcal{Z} \subseteq \mathcal{X}$, define

$$M(\mathcal{Z}) = \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2) \; ; \quad \textit{and} \quad q(\mathcal{Z}) = \max_{a \in A} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}} \, . \tag{1.1}$$

If there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$, then the randomized query complexity of the problem is at least

$$\min_{\mathcal{Z} \subseteq \mathcal{X} : q(\mathcal{Z}) > 0} \frac{M(\mathcal{Z})}{100 \cdot q(\mathcal{Z})} \, . \tag{1.2}$$

In Section 1.5, we also show an example on which our variant is strictly stronger, giving a tight lower bound for the query complexity of a simple "matrix game". Then we prove our variant is asymptotically at least as strong in general, for randomized algorithms.

[2]↑Each function $F \in \mathcal{X}$ has the form $F : A \to B$.

[3]↑In other words, we have free access to $\mathcal{H}$ and the only queries counted are the ones to F^*, which will be of the form: "What is $F^*(a)$?", for some $a \in A$. The oracle will return $F^*(a)$ in one computational step.

1.3.2 Lower bounds for local search via congestion

Next we give the first known lower bound for local search as a function of (vertex) congestion, which is enabled by our Theorem 1.3.1.

Theorem 1.3.2. *Let $G = (V, E)$ be a connected undirected graph with n vertices. Then the randomized query complexity of local search on G is $\Omega\left(\frac{n^{1.5}}{g}\right)$, where g is the vertex congestion of the graph.*

Since $g \in [n, n^2]$, Theorem 1.3.2 cannot be used to show a lower bound stronger than $\Omega(\sqrt{n})$ queries, matching a general upper bound of $O(\sqrt{n})$ for graphs with bounded degree ([9]). Theorem 1.3.2 gives meaningful results precisely when one can construct an all-pairs set of paths with vertex congestion $g = o(n^{1.5})$; e.g. our bound is vacuous on trees since $g \in \Theta(n^2)$ on trees.

Theorem 1.3.2 also implies a lower bound of $\Omega\left(\frac{n^{1.5}}{g_{\mathrm{e}} \cdot \Delta}\right)$ on any graph G, where g_{e} is the edge congestion and Δ the maximum degree of G.

High level approach.

To obtain the result in Theorem 1.3.2, we apply Yao's lemma and design a hard input distribution where the input is a random function $f : V \to \mathbb{R}$ induced by a "staircase". A staircase is a walk of vertices $v_1, v_2, \ldots, v_k$ of some length k where v_1 is the entrance and v_k is the end. The value of the function f outside the staircase is equal to the distance (in the graph) to the entrance of the staircase, while the value at vertices on the staircase decreases as one walks away from the entrance. The local minimum is unique and can be found at the end of the staircase. This type of construction is classical (see, e.g., [3, 9, 10]). We designate the space of such functions as $\mathcal{X}$.

A staircase is characterized by a sequence of vertices (milestones). Then we connect each pair w, z of consecutive milestones with the path $P^{w,z}$ from the all-pairs set of paths $\mathcal{P} =$

$\{P^{u,v}\}_{u,v\in[n]}$ with "low" congestion (that is, with congestion as low as possible given the graph).

Then, we design a function r — also called the relation — that "relates" any two such functions. Any such non-zero relation induces a distribution over functions where each function F_1 is sampled with likelihood equal to $\sum_{F_2\in\mathcal{X}} r(F_1, F_2)$.

Our choice of r increases according to the number of milestones the underlying staircases have in common (specifically, by the longest initial prefix of milestones shared by the two staircases). With our choice of r, the distribution over staircases is as though the milestones were chosen uniformly at random without replacement.

Any two functions F_1 and F_2 with long initial prefix in their corresponding staircases are very similar and so will be hard to distinguish by an algorithm. Roughly speaking, without querying sufficiently many vertices and chancing upon a vertex which F_1 and F_2 disagrees on, the algorithm will not be able to distinguish them. In order to capture this difficulty of distinguishing such F_1 and F_2, the relation $r(F_1, F_2)$ will be set to a high value. Our congestion lower bound then follows by invoking our variant of the relational adversary (Theorem 1.3.1) with this choice of r.

Our approach was inspired by previous works such as [3], who gave lower bounds for the d-dimensional grid and the Boolean hypercube, and of [16], who gave lower bounds for Cayley and vertex transitive graphs by using a system of *carefully chosen* shortest paths rather than arbitrary shortest paths; this inspired our choice of the set of paths $\mathcal{P}$. We explain in more detail the comparison with [3] and [16] in Section 1.4.1 of Related work.

1.3.3 Lower bounds for local search via separation number

We also give an *improved* lower bound for local search with respect to the graph separation number s. Our construction is heavily inspired by the one in [17] , who gave a lower bound of $\Omega\left(\sqrt[8]{\frac{s}{\Delta}}/\log n\right)$, for both the quantum and classical randomized algorithms. Adapting this construction within the framework of Theorem 1.3.1 is non-trivial however.

Theorem 1.3.3. *Let $G = (V, E)$ be a connected undirected graph with n vertices, maximum degree Δ, and separation number s. Then the randomized query complexity of local search on G is $\Omega\left(\sqrt[4]{\frac{s}{\Delta}}\right)$.*

The best known upper bound with respect to graph separation number is $O((s+\Delta)\cdot\log n)$ due to [17], which was obtained via a refinement of the divide-and-conquer procedure of [18]. It is an interesting open question whether the current upper and lower bounds can be improved.

1.3.4 Corollaries for expanders, Cayley graphs, and the hypercube

Since d-regular β-expanders with constant d and β admit an all-pairs set of paths with congestion $O(n \cdot \log n)$ (e.g., see [19]), we get the next lower bound for constant degree expanders.

Corollary 1. *Let $G = (V, E)$ be an undirected d-regular β-expander with n vertices, where d and β are constant. Then the randomized query complexity of local search on G is $\Omega\left(\frac{\sqrt{n}}{\log n}\right)$.*

The lower bound of Corollary 1 is tight within a logarithmic factor. A simple algorithm known as steepest descent with warm start ([9]) can be used to see this:

> First query t vertices $x_1, \ldots, x_t$ selected uniformly at random and pick the vertex x^* that minimizes the function among these[4]. Then run steepest descent from x^* and stop when no further improvement can be made, returning the final vertex reached. When $t = \sqrt{n\Delta}$, where Δ is the maximum degree of the graph, the algorithm issues $O(\sqrt{n\Delta})$ queries in expectation.

Thus steepest descent with a warm start has expected query complexity $O(\sqrt{n})$ on constant degree expanders. Our lower bound implies this algorithm is essentially optimal on such graphs.

[4]↑That is, the vertex x^* is defined as: $x^* = x_{\mathrm{j}}$, where $\mathrm{j} = \mathrm{argmin}_{\mathrm{i}=1}^{t} f(x_{\mathrm{i}})$.

We also get a lower bound as a function of the expansion and maximum degree of the graph G.

Corollary 2. *Let $G = (V, E)$ be an undirected β-expander with n vertices and maximum degree Δ. Then the randomized query complexity of local search on G is $\Omega\left(\frac{\beta\sqrt{n}}{\Delta \log^2 n}\right)$.*

The congestion framework also allows us to recover a lower bound for undirected Cayley graphs, which were studied before in [16]. An *undirected Cayley* graph is formed from a group $\mathcal{G}$ and a generating set S. The vertices are the elements of $\mathcal{G}$ and there is an edge between vertices u, v if $u = w \cdot v$ or $v = w \cdot u$ for some $w \in S$.

Corollary 3. *Let $G = (V, E)$ be an undirected Cayley graph with n vertices and diameter $diam(G)$. Then the randomized query complexity of local search on G is $\Omega\left(\frac{\sqrt{n}}{diam(G)}\right)$.*

We also get the next corollary for the query complexity of local search on the Boolean hypercube.

Corollary 4. *The randomized query complexity of local search on the Boolean hypercube $\{0, 1\}^n$ is $\Omega\left(\frac{2^{n/2}}{n}\right)$.*

The lower bound in Corollary 4 is sandwiched between the lower bound of $\Omega\left(\frac{2^{n/2}}{\sqrt{n}}\right)$ by [20] and the lower bound of $\Omega\left(\frac{2^{n/2}}{n^2}\right)$ by [3].

1.4 Related work

The query complexity of local search was first studied experimentally by [21]. The first breakthrough in the theoretical analysis of local search was obtained by [9]. Aldous stated the algorithm based on steepest descent with a warm start and showed the first nontrivial lower bound of $\Omega(2^{n/2-o(n)})$ on the query complexity for the Boolean hypercube $\{0, 1\}^n$.

The lower bound construction from [9] uses Yao's lemma and describes a hard distribution, such that if a deterministic algorithm receives a random function according to this distribution, the expected number of queries issued until finding a local minimum will be large. The random function is obtained as follows:

> Consider an initial vertex v_0 uniformly at random. Set the function value at v_0 to $f(v_0) = 0$. From this vertex, start an unbiased random walk $v_0, v_1, \ldots$ For each vertex v in the graph, set $f(v)$ equal to the first hitting time of the walk at v; that is, let $f(v) = \min\{t \mid v_t = v\}$.

The function f defined this way has a unique local minimum at v_0. By a very delicate analysis of this distribution, [9] showed a lower bound of $\Omega(2^{n/2-o(n)})$ on the hypercube $\{0,1\}^n$.

This almost matches the query complexity of steepest descent with a warm start, which was also analyzed in [9] and shown to take $O(\sqrt{n} \cdot 2^{n/2})$ queries in expectation on the hypercube. The steepest descent with a warm start algorithm applies to generic graphs too, resulting in $O(\sqrt{n \cdot \Delta})$ queries overall for any graph with maximum degree Δ.

Aldous' lower bound for the hypercube was later improved via more refined types of random walks and/or more careful analysis. [3] improved the bound to $\Omega(2^{n/2}/n^2)$ via his relational adversary method, which is a combinatorial framework that avoids analyzing the posterior distribution and also yielded a quantum bound of $\Omega(2^{n/4}/n)$. [20] improved the randomized lower bound to a tight bound of $\Theta(2^{n/2} \cdot \sqrt{n})$ via a "clock"-based random walk construction, which avoids self-intersections.

Meanwhile, [18] developed a deterministic divide-and-conquer approach to solving local search that is theoretically optimal over all graphs in the deterministic context, albeit hard to apply in practice. On the hypercube, their method yields a lower bound of $\Omega(2^n/\sqrt{n})$ and an upper bound of $O(2^n \log(n)/\sqrt{n})$: a mere $\log(n)$ factor gap.

Another commonly studied graph for local search is the d-dimensional grid $[n]^d$. [3] used his relational adversary method there to show a randomized lower bound of $\Omega(n^{d/2-1}/\log n)$ for every constant $d \geq 3$. [20] proved a randomized lower bound of $\Omega(n^{d/2})$ for every constant $d \geq 4$; this is tight as shown by Aldous' generic upper bound. Zhang also showed improved bounds of $\Omega(n^{2/3})$ and $\Omega(n^{3/2}/\sqrt{\log n})$ for $d = 2$ and $d = 3$ respectively, as well as some quantum results. The work of [22] closed further gaps in the quantum setting as well as the randomized $d = 2$ case. The problem of local search on the grid was also studied under the context of multiple limited rounds of adaptive interactions by [23].

More general results are few and far between. On many graphs, the simple bound from [9] of $\Omega(\Delta)$ queries is the best known lower bound: hiding the local minimum in one of the Δ leaves of a star subgraph requires checking about half the leaves in expectation to find it.

[17] gave a quantum lower bound of $\Omega\left(\sqrt[8]{\frac{s}{\Delta}}/\log(n)\right)$, where s is the separation number of the graph. This implies the same lower bound in a randomized context, using the spectral method. Meanwhile, the best known upper bound is $O((s + \Delta) \cdot \log n)$ due to [17], which was obtained via a refinement of the divide-and-conquer procedure of [18].

[16] studied Cayley and vertex transitive graphs and gave lower bounds for local search as a function of the number of vertices and the diameter of the graph. We explain the comparison with their work more precisely in Section 1.4.1. [24] obtained upper bounds as a function of the genus of the graph.

[25] studied the communication complexity of local search. This captures distributed settings, where data is stored in the cloud, on different computers.

There is a rich literature analyzing the congestion of graphs. E.g., the notion of edge congestion is important in routing problems, where systems of paths with low edge congestion can enable traffic with minimum delays (see, e.g., [19, 26, 27]). This problem is sometimes called *multicommodity flow* or *edge disjoint paths with congestion*. Others study routing with the goal of maximizing the number of demand pairs routed using *node* disjoint paths; this is the same as requiring vertex congestion equal to 1 (see, e.g., [28, 29]).

Local search is strongly related to the problem of local optimization where one is interested in finding an approximate local minimum of a function on $\mathbb{R}^d$. A common way to solve local optimization problems is to employ gradient-based methods, which find approximate stationary points. To show lower bounds for finding stationary points, one can similarly define a function that selects a walk in the underlying space and hide a stationary point at the end of the walk. Handling the requirement that the function is smooth and ensuring there is a unique stationary point are additional challenges.

For examples of works on algorithms and complexity of computing approximate stationary points, see, e.g., [10, 30–34]). Constructions where the function is induced by a hidden walk have first been designed for showing lower bounds on the query complexity of finding Brouwer fixed points in classical work by [35].

Works like [11] study stochastic gradient descent, which is one method of finding approximate local minima. Moreover, they do this on Riemann manifolds, which are a very broad class of spaces. This motivates the need to study local search not only on hypercubes and grids, but also on other, broader classes of graphs. For a more extensive survey, see, e.g., [12].

The computational complexity of local search is captured by the class PLS, defined by [8] to model the difficulty of finding locally optimal solutions to optimization problems. A related class is PPAD, introduced by [36] to study the computational complexity of finding a Brouwer fixed-point. Both PLS and PPAD are subsets of of the class TFNP.

The class PPAD contains many natural problems that are computationally equivalent to the problem of finding a Brouwer fixed point ([37]), such as finding an approximate Nash equilibrium in a multiplayer or two-player game ([38, 39]), an Arrow-Debreu equilibrium in a market ([40, 41]), and a local min-max point ([42]).

The query complexity of computing an ε-approximate Brouwer fixed point was studied in a series of papers starting with [35], later improved by [43] and [44]. Recently, [45] showed that the class CLS, introduced by [46] to capture continuous local search, is equal to PPAD $\cap$ PLS. The query complexity of continuous local search has also been studied (see, e.g., [47]).

1.4.1 Comparison with prior works and corrections

Our approach for the lower bound as a function of congestion was directly inspired by the relational adversary method of [3] and an ingenious application to vertex-transitive graphs by [16]. In both of these papers, a hard distribution is obtained by getting a random input function induced by a "staircase" (walk): the value of the function outside the staircase is equal to the distance to the entrance of the staircase, while the value of the function on the staircase is decreasing as one moves away from the entrance.

[16] choose a staircase by first selecting several random points (milestones) and then connecting them with a path from a system of paths. While one typically chooses *arbitrary* shortest paths between two endpoints when constructing lower bounds, the system of paths in their work consists of *carefully chosen* shortest paths as follows: (i) For paths that start from a fixed vertex v_0, fix arbitrary shortest paths; (ii) For paths starting from other vertices v_{i} with $\mathrm{i} > 0$, use one of the same paths as from v_0, but transformed by an automorphism mapping v_0 to v_{i} (which is defined when the graph is vertex transitive).

The high-level approach and the careful selection of paths in [16] inspired our choice for the set of paths $\mathcal{P}$ with low congestion when the graph is not necessarily vertex transitive.

Given the family of functions and staircases described, what remains is to define a relation between functions and compute the lower bound obtained by invoking the relational adversary method.

Corrections.

There appears to be a potential issue in the proof of Proposition 2 in [16], where the probability of an event is higher than would be required for the argument to go through [5]. We

[5]↑ In [16], for any vertex v that appears in a walk $\mathbf{x} = (x_0, x_1, x_2, \ldots, x_L)$, the function is defined as $f_{\mathbf{x}}(v) = L - \max\{\mathrm{i} : v = x_{\mathrm{i}}\}$, i.e. L minus the last index v last appears on $\mathbf{x}$. Then, a vertex v should be a disagreement between two walks $\mathbf{x}$ and $\mathbf{y}$ if v lies on both walks but $f_{\mathbf{x}}(v) \neq f_{\mathbf{y}}(v)$. Afterwards, in Proposition 2, if vertex v is a disagreement between two walks $\mathbf{x}$ and $\mathbf{y}$, then v must lie on at least one of the walks, and moreover, v was stated to be contained in a "segment" of the tail that is not the one immediately after the divergence place of $\mathbf{x}$ and $\mathbf{y}$ (quote: "*We can't have both $t < \mathrm{j} + s$ and $t' < \mathrm{j} + s$... either $t \geq \mathrm{j} + s$*

bypassed this using our Theorem 1.3.1, which enabled us to recover the randomized lower bound for Cayley graphs from [16]; see Corollary 3.

This also occurs in the proof [6] of Lemma 6.2 of [3], where it could be corrected by setting the function $f_X(v)$, for a vertex v, as $f_X(v) = \min\{t : x_t = v \text{ and } x_{t+1} \neq v\}$.

1.5 A variant of the relational adversary method

In this section we state our variant of the relational adversary method (Theorem 1.3.1). After stating the variant, we also design and analyze a "matrix game", to illustrate a simple problem for which our variant yields a better (in fact tight) lower bound for the randomized query complexity. The complete details and proofs are included in Section 1.9, together with the original theorem from [3] for comparison.

Theorem 1.3.1. *Consider finite sets A and B, a set $\mathcal{X} \subseteq B^A$ of functions [7], and a map $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ which assigns a label to each function in $\mathcal{X}$. Additionally, we get oracle access to an unknown function $F^* \in \mathcal{X}$. The problem is to compute $\mathcal{H}(F^*)$ using as few queries to F^* as possible.*[8]

Let $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a non-zero symmetric function of our choice with $r(F_1, F_2) = 0$ whenever $\mathcal{H}(F_1) = \mathcal{H}(F_2)$. For each $\mathcal{Z} \subseteq \mathcal{X}$, define

$$M(\mathcal{Z}) = \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2) \,; \quad \text{and} \quad q(\mathcal{Z}) = \max_{a \in A} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}} \,. \quad (1.1)$$

or $t' \geq \mathrm{j} + s$", where $t = \max\{\mathrm{i} : v = x_\mathrm{i}\}$ is the last index of v on $\mathbf{x}$, $t' = \max\{\mathrm{i} : v = y_\mathrm{i}\}$ is the last index of v on $\mathbf{y}$, j is the point of splitting, and s is the "segment length"). However, one can check the following setup which violates this assertion also makes v a disagreement: $\max\{\mathrm{i} : v = x_\mathrm{i}\} < \mathrm{j} < \max\{\mathrm{i} : v = y_\mathrm{i}\} < \mathrm{j} + s$, where v appears in the shared prefix and then only on $\mathbf{x}$ within the first segment after they diverge.

[6]↑In [3], Lemma 6.2: in the first case, where $t > \mathrm{j} - n$ and $t^* > \mathrm{j} - n$, the probability $P[x_t = y_{t^*}]$ is not zero, but rather can be nearly 1/2 (e.g. at $t = t^* = \mathrm{j}$), since the coordinate loop allows staying in place. With probability 1/2, exactly one of $x_{\mathrm{j}-1}$ and $y_{\mathrm{j}-1}$ is equal to x_j, which usually makes $f_X(x_\mathrm{j}) \neq f_Y(x_\mathrm{j})$. We have reached out to the author and he has acknowledged the error, as well as suggesting another possible fix.

[7]↑Each function $F \in \mathcal{X}$ has the form $F : A \to B$.

[8]↑In other words, we have free access to $\mathcal{H}$ and the only queries counted are the ones to F^*, which will be of the form: "What is $F^*(a)$?", for some $a \in A$. The oracle will return $F^*(a)$ in one computational step.

If there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$, then the randomized query complexity of the problem is at least

$$\min_{\mathcal{Z} \subseteq \mathcal{X} : q(\mathcal{Z}) > 0} \frac{M(\mathcal{Z})}{100 \cdot q(\mathcal{Z})} . \tag{1.2}$$

Theorem 1.3.1 uses Yao's lemma (see Section 1.8), thus the algorithm can be assumed to be deterministic and receive as input a random function sampled from some distribution P.

The theorem considers the probability distribution P where each function $F \in \mathcal{X}$ is given as input with probability $P(F) = \frac{M(\{F\})}{M(\mathcal{X})}$, where $\mathcal{X}$ is the space of possible functions.

The term $M(\mathcal{Z})$ represents the likelihood that the function F^* given as input comes from the set $\mathcal{Z} \subseteq \mathcal{X}$, while $q(\mathcal{Z})$ is proportional to a lower bound on the number of queries needed in the worst case to narrow down the function F^* within $\mathcal{Z}$, conditioned on F^* being in $\mathcal{Z}$.

Clearly some choices of r are more useful than others, and so the challenge when giving lower bounds is to design the function r and estimate the expression in equation (1.2).

1.5.1 Matrix game

In this section, we describe a toy problem upon for which our new variant (Theorem 1.3.1) can show a stronger lower bound than the original relational adversary theorem.

Setup

Let $n \in \mathbb{N}$ be a perfect square and $\mathcal{X}$ be a subset of square $\sqrt{n} \times \sqrt{n}$ matrices with entries from $\{0, 1, 2\}$. There are two types of matrices within $\mathcal{X}$: "row" matrices and "column" matrices. Row matrices have one row of 1s with all other entries 0 while column matrices have one column of 2s with all other entries 0. For example:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ is a row matrix, and } \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix} \text{ is a column matrix.}$$

So, $|\mathcal{X}| = 2\sqrt{n}$ since there are $\sqrt{n}$ distinct row matrices and $\sqrt{n}$ distinct column matrices.

The game.

Given n and oracle access to a matrix $F \in \mathcal{X}$, the goal is to correctly declare whether F is a row or column matrix.

Lemma 1. *The randomized query complexity of the matrix game is* $\Theta(\sqrt{n})$.

Proof sketch. We give a high level explanation of the proof, while the complete details can be found in Section 1.9. For the upper bound, we can check that $\sqrt{n}$ queries suffice. Query the entries of the main diagonal and then proceed as follows: if any "1" is detected, declare "row"; if any "2" is found, declare "column".

For the lower bound, one intuitively expects that $\Omega(\sqrt{n})$ queries are necessary even allowing randomization. In fact, this is what we can show using Theorem 1.3.1. Choose the function r so that $r(F_1, F_2)$ represents the indicator function for whether the two matrices are "opposite types", for any two matrices $F_1, F_2 \in \mathcal{X}$. This means that the probability to sample any subset $\mathcal{Z} \subseteq \mathcal{X}$ will be proportional to its size $|\mathcal{Z}|$.

Meanwhile, for any subset $\mathcal{Z} \subseteq \mathcal{X}$ of matrices, *any single query* on a matrix coordinate will distinguish at most two matrices from all the others *within* $\mathcal{Z}$. Thus, we can show that the ratio $M(\mathcal{Z})/q(\mathcal{Z}) \in \Omega(\sqrt{n})$ for *any* subset $\mathcal{Z} \subseteq \mathcal{X}$ of matrices, which implies a $\Omega(\sqrt{n})$ lower bound of the matrix game via Theorem 1.3.1. □

On the other hand, one can only show a lower bound of $\Omega(1)$ for the matrix game using the version of the relational adversary method from [3]; see details in Section 1.9.

1.5.2 Advantage of our variant for randomized algorithms

In addition to being strictly stronger on some problems (like this matrix game), our variant is at least as strong in general, for randomized algorithms.

Proposition 1.5.1. *Consider any problem and let T be the expected number of queries required in the worst case by the best randomized algorithm to succeed with probability* $9/10$.

If the relational adversary method from [3] provides a lower bound of $T \geq \Lambda$ for some $\Lambda > 0$, then Theorem 1.3.1 can prove a lower bound of $T \geq \Lambda/40$.

1.6 Lower bound for local search via congestion

In this section, we explain at a high level the proof of Theorem 1.3.2, which gives a lower bound as a function of congestion, as well as the corollary for expanders. See Section 1.11 for the details.

1.6.1 Proof sketch for the congestion lower bound

The proof of Theorem 1.3.2 is sketched in the next sequence of steps. We define the following with the intention of invoking Theorem 1.3.1.

Fixing a set of paths $\mathcal{P}$.

Since the graph $G = ([n], E)$ has vertex congestion g, we can fix an all-pairs set of paths $\mathcal{P} = \{P^{u,v}\}_{u,v \in [n]}$, such that $P^{u,v} \in \mathcal{P}$ is a simple path from u to v and $P^{u,u} = (u)$, for each $u, v \in [n]$. Moreover, each vertex is used at most g times across paths in $\mathcal{P}$.

Staircases.

We fix a parameter $L \in [n]$ to be set later. We consider sequences of vertices of the form $\mathbf{x} = (x_1, \ldots, x_{L+1}) \in \{1\} \times [n]^L$, i.e. with $x_1 = 1$. The *staircase* induced by $\mathbf{x}$ is a walk $S_{\mathbf{x}} = S_{\mathbf{x},1} \circ \ldots \circ S_{\mathbf{x},L}$, where each $S_{\mathbf{x},\mathrm{i}}$ is a path in G starting at vertex x_i and ending at $x_{\mathrm{i}+1}$. Each vertex x_i is called a *milestone* and each path $S_{\mathbf{x},\mathrm{i}}$ a *quasi-segment.*

The staircase $S_{\mathbf{x}}$ is *induced by* $\mathbf{x}$ *and* $\mathcal{P}$ if we additionally have $S_{\mathbf{x},\mathrm{i}} = P^{x_\mathrm{i}, x_{\mathrm{i}+1}}$ for all $\mathrm{i} \in [L]$. In other words, to build such a staircase we first decide on the sequence of "milestones" $\mathbf{x}$; then to get from each milestone x_i to the next milestone $x_{\mathrm{i}+1}$, we travel using the path $P^{x_\mathrm{i}, x_{\mathrm{i}+1}}$ from the set of paths $\mathcal{P}$. Note that $P^{x_\mathrm{i}, x_{\mathrm{i}+1}}$ may not be the shortest path between x_i and $x_{\mathrm{i}+1}$ since $\mathcal{P}$ does not necessarily only consist of shortest paths.

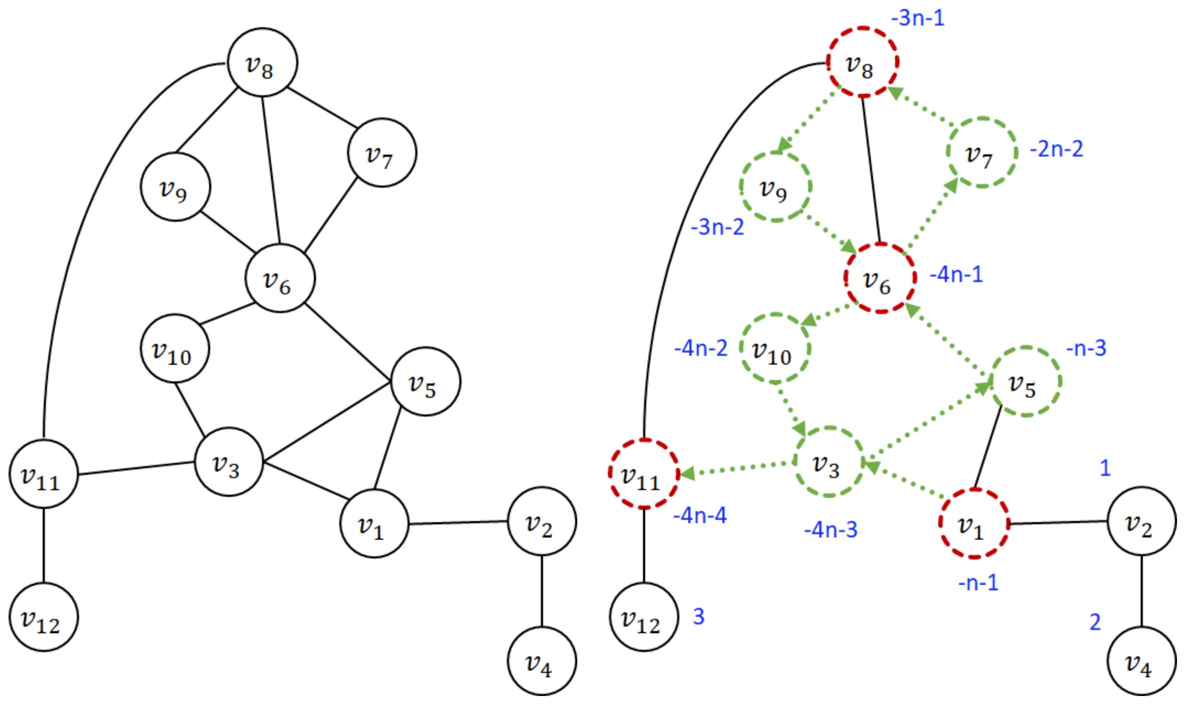
The value function $f_{\mathbf{x}}$.

For each staircase $S_{\mathbf{x}}$ induced by $\mathbf{x}$ and $\mathcal{P}$, we define a corresponding function $f_{\mathbf{x}} : [n] \to \mathbb{R}$ as follows. For each vertex v in G:

(a) If $v \notin S_{\mathbf{x}}$, then set $f_{\mathbf{x}}(v) = dist(v, 1)$.

(b) If $v \in S_{\mathbf{x}}$, then set $f_{\mathbf{x}}(v) = -\mathrm{i} \cdot n - \mathrm{j}$, where i is the maximum index with $v \in P^{x_\mathrm{i}, x_{\mathrm{i}+1}}$, and v is the j-th vertex in $P^{x_\mathrm{i}, x_{\mathrm{i}+1}}$.

The following example gives a visual illustration of an induced staircase $S_{\mathbf{x}}$ and its associated function $f_{\mathbf{x}}$. Such a function $f_{\mathbf{x}}$ has a unique local minimum at the end of the staircase $S_{\mathbf{x}}$. Our lower bound construction uses such functions.



(a) Graph with $n = 12$ vertices (b) Staircase with induced value function.

Figure 1.1. Graph G with $n = 12$ vertices labelled $\{v_1, \ldots, v_{12}\}$. The staircase shown consists of the walk given by the dotted vertices $(v_1, v_3, v_5, v_6, v_7, v_8, v_9, v_6, v_{10}, v_3, v_{11})$. The value of the function at each vertex is shown in blue near that vertex. The local minimum is at the end of the staircase, at vertex v_{11}.

Example 1 (Staircase with the associated value function.)**.** *Consider the graph G in Figure 1.1, with $n = 12$ vertices labelled $\{v_1, \ldots, v_{12}\}$.*

Let the set of paths $\mathcal{P} = \{P^{u,v}\}_{u,v \in [n]}$ be

- $P^{v_1,v_6} = (v_1, v_3, v_5, v_6)$*;* $P^{v_6,v_8} = (v_6, v_7, v_8)$*;* $P^{v_8,v_6} = (v_8, v_9, v_6)$*;* $P^{v_6,v_{11}} = (v_6, v_{10}, v_3, v_{11})$*.*
- *For all other pairs of vertices (u, w), we set $P^{u,w}$ as the shortest path from u to w.*

Let $L = 4$. Consider a sequence $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5) = (v_1, v_6, v_8, v_6, v_{11})$, where each milestone is highlighted by a red dotted circle. Observe we allow repeated vertices. The staircase induced by $\mathbf{x}$ and $\mathcal{P}$, given by the green dotted walk, is

$$S_{\mathbf{x}} = (v_1, v_3, v_5, v_6, v_7, v_8, v_9, v_6, v_{10}, v_3, v_{11})\,.$$

The value function $f_{\mathbf{x}}$, computed using the definition (a-b), of each vertex is given in blue.

For technical reasons, in the lower bound proof we will actually work with a decision problem. There is a simple way to turn a search problem into a decision problem (see [3, 16]): associate with each function $f_{\mathbf{x}}$ a function $g_{\mathbf{x},b}$ that hides a bit at the local minimum vertex (while hiding -1 at every other vertex). Formally, $g_{\mathbf{x},b}$ is defined next.

Let $A = [n]$ and $B = \{-n^2 - n, \dots, 0, \dots, n\} \times \{-1, 0, 1\}$ be finite sets. For each vertex $\mathbf{x} \in \{1\} \times [n]^L$ and bit $b \in \{0,1\}$, let $g_{\mathbf{x},b} : A \to B$ be such that, for all $v \in [n]$:

$$g_{\mathbf{x},b}(v) = \begin{cases} \Big(f_{\mathbf{x}}(v), b\Big) & \text{if } v = x_{L+1} \\ \Big(f_{\mathbf{x}}(v), -1\Big) & \text{if } v \neq x_{L+1} \end{cases}. \tag{1.3}$$

The set of functions we consider is $\mathcal{X} = \Big\{ g_{\mathbf{x},b} \mid \mathbf{x} \in \{1\} \times [n]^L \text{ and } b \in \{0,1\} \Big\}$.

The decision problem is: given a graph G and oracle access to a function $g_{\mathbf{x},b} \in \mathcal{X}$, return the value $\mathcal{H}(g_{\mathbf{x},b}) = b$. This means: find the hidden bit, which only exists at the vertex corresponding to the local minimum of $f_{\mathbf{x}}$. Measuring the query complexity of this decision problem will give the answer for local search, as the next two problems have query complexity within additive 1:

- *search problem:* given oracle access to a function $f_{\mathbf{x}}$, find a vertex v that is a local minimum;
- *decision problem:* given oracle access to the function $g_{\mathbf{x},b}$, find $\mathcal{H}(g_{\mathbf{x},b})$.

Good/bad sequences of vertices; Good/bad functions.

We divide the set $\mathcal{X}$ of functions into "good" and "bad" functions. Our analysis later will focus on "good" sequences and functions.

A sequence of vertices $\mathbf{x} = (x_1, \ldots, x_{L+1})$ is *good* if $x_{\mathrm{i}} \neq x_{\mathrm{j}}$ for all $\mathrm{i}, \mathrm{j} \in [L+1]$ with $\mathrm{i} < \mathrm{j}$; otherwise, $\mathbf{x}$ is *bad*. That is, $\mathbf{x}$ only involves distinct milestones.

For each bit $b \in \{0, 1\}$, the function $g_{\mathbf{x},b} \in \mathcal{X}$ is *good* if $\mathbf{x}$ is good, and *bad* otherwise.

The relation function r.

To be able to invoke Theorem 1.3.1, we need to also define the relation function r whose role is to "relate" pairs of input functions F_1 and F_2 in order to roughly capture the difficulty of differentiating F_1 from F_2.

Intuitively, an algorithm Γ will query vertices of the graph to eliminate options to figure out which is the underlying input function, from which it can query the local minimum to retrieve the hidden bit b. However, if two functions $F_1 = g_{\mathbf{x},b_1}$ and $F_2 = g_{\mathbf{y},b_2}$ are very "similar" (i.e. their underlying staircases $S_{\mathbf{x}}$ and $S_{\mathbf{y}}$ are almost identical), then it may take Γ many queries to learn whether the input is F_1 or F_2, even if it knows the input can only be one of these two functions. Consequently, Γ will have great difficulty finding the local minimum on certain inputs.

Formally, for all sequences of vertices $\mathbf{x}$ let $\mathbf{x}_{\mathrm{i}\to\mathrm{j}}$ be the sequence $(x_{\mathrm{i}}, x_{\mathrm{i}+1}, \ldots, x_{\mathrm{j}})$. Then define $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ as a symmetric function such that for each $\mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L$ and $b_1, b_2 \in \{0, 1\}$:

$$r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = \begin{cases} 0 & \text{if at least one of the following holds: } b_1 = b_2 \text{ or } \mathbf{x} \text{ is bad or } \mathbf{y} \text{ is bad.} \\ n^{\mathrm{j}} & \text{otherwise, where j is the maximum index for which } \mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}\,. \end{cases}$$

The choice of r is deliberate, as we see next.

Invoking Theorem 1.3.1.

We are now ready to invoke Theorem 1.3.1 using the definitions of $f_{\mathbf{x}}$, $g_{\mathbf{x},b}$, $\mathcal{X}, \mathcal{H}$, and r from above. We will show there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$, and so we get that the randomized query complexity of local search is:

$$\Omega\left(\min_{\substack{\mathcal{Z}\subseteq\mathcal{X}:\\ q(\mathcal{Z})>0}} \frac{M(\mathcal{Z})}{q(\mathcal{Z})}\right), \text{ where } \quad q(\mathcal{Z}) = \max_{v\in[n]} \sum_{F_1\in\mathcal{Z}} \sum_{F_2\in\mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v)\neq F_2(v)\}}\,. \tag{1.4}$$

Estimating the lower bound in (1.4) precisely is quite challenging. Instead, for any arbitrary $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$, we show a lower bound $M(\mathcal{Z})$ and an upper bound for $q(\mathcal{Z})$ using our choice of the function r. The two bounds we show only depend on $\mathcal{Z}$ via its size $|\mathcal{Z}|$. For any fixed $\mathcal{Z}$, this dependency will be cancelled out through the division, and thus our result follows.

Lower bounding $M(\mathcal{Z})$.

By our choice of r, only *good* functions affect the value of $M(\mathcal{Z})$. Furthermore, we know that a function is good only if it was constructed using a good sequence of milestones $\mathcal{X}$. For any good function $F_1 \in \mathcal{X}$, a counting argument tells us that there are roughly $n^{L+1-\mathrm{j}}$ good functions $F_2 \in \mathcal{X}$ with $r(F_1, F_2) = n^{\mathrm{j}}$, i.e. F_1 and F_2 have the same milestones from the first to the j^{th} one. This approximation holds for $\mathrm{j} \in O(\sqrt{n})$; we will eventually have $\mathrm{j} \le L \in O(\sqrt{n})$, so this is fine. Therefore, for any good function $F_1 \in \mathcal{X}$, we have

$$\sum_{F_2\in\mathcal{X}:\, r(F_1,F_2)=n^{\mathrm{j}}} r(F_1, F_2) \approx \sum_{\mathrm{j}=1}^{L} n^{\mathrm{j}} \cdot n^{L+1-\mathrm{j}} = L \cdot n^{L+1}\,.$$

Therefore $M(\mathcal{Z}) \in \Omega(|\mathcal{Z}| \cdot L \cdot n^{L+1})$.

Upper bounding $q(\mathcal{Z})$.

Let $v \in [n]$ be a vertex and $F_1 = g_{\mathbf{x},b} \in \mathcal{Z}$ a good function with hidden bit $b \in \{0,1\}$. By our choice of r, it suffices to relate F_1 to functions $F_2 \in \mathcal{Z}$ that are also good but have hidden bit $1-b$, i.e. of the form $F_2 = g_{\mathbf{y},1-b}$, for some good sequence $\mathbf{y}$.

To upper bound the inner summation of $q(\mathcal{Z})$, we partition the functions F_2 based on the length of the shared prefix of $\mathbf{y}$ and $\mathbf{x}$. We get

$$\sum_{F_2 \in \mathcal{Z}} r(F_1,F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} = \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{F_2 \in \mathcal{Z}: \\ \mathrm{j}=\max\{\mathrm{i}: \mathbf{x}_{\mathrm{i}\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}\}}} r(F_1,F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} \,. \tag{1.5}$$

When $\mathrm{j} = L+1$, there is exactly one function $F_2 \in \mathcal{Z}$ with such a shared prefix. Thus we get a contribution of n^{L+1} by our definition of r.

Thus from now on, we focus on $1 \leq \mathrm{j} \leq L$. Let $Tail(\mathrm{j}, S_{\mathbf{y}})$ denote the suffix of the staircase $S_{\mathbf{y}}$ after y_{j}. We can then upper bound the j-th term from (1.5) as follows:

$$\sum_{\substack{F_2 \in \mathcal{Z}: \\ \mathrm{j}=\max\{\mathrm{i}: \mathbf{x}_{\mathrm{i}\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}\}}} r(F_1,F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} \leq 2 \cdot \sum_{\substack{F_2 \in \mathcal{Z}: \\ \mathrm{j}=\max\{\mathrm{i}: \mathbf{x}_{\mathrm{i}\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}\} \\ v \in Tail(\mathrm{j}, S_{\mathbf{y}})}} r(F_1,F_2) \,. \tag{1.6}$$

Recall the staircase has the form $S_{\mathbf{y}} = P^{y_1,y_2} \circ \ldots \circ P^{y_{\mathrm{j}},y_{\mathrm{j}+1}} \circ \ldots \circ P^{y_L,y_{L+1}}$, where each $P^{y_{\mathrm{i}},y_{\mathrm{i}+1}}$ is a path from $\mathcal{P}$. When v is in $Tail(\mathrm{j}, S_{\mathbf{y}})$, it means that $v \in P^{y_{\mathrm{i}},y_{\mathrm{i}+1}}$ for some $\mathrm{j} \leq \mathrm{i} \leq L$. We will upper bound the number of sequences $\mathbf{y}$ depending on the location of v.

Case $\mathrm{i} = \mathrm{j}$**.** Let $\mathfrak{q}_v(u)$ denote the number of paths in the set $\mathcal{P}$ that start at vertex u and contain v. There are $\mathfrak{q}_v(x_{\mathrm{j}})$ choices of vertices for $y_{\mathrm{j}+1}$ and $n^{L-\mathrm{j}}$ choices for sequences $(y_{\mathrm{j}+2}, \ldots, y_{L+1})$, yielding a total count of $\mathfrak{q}_v(x_{\mathrm{j}}) \cdot n^{L-\mathrm{j}}$.

Case $\mathrm{j} < \mathrm{i} \leq L$**.** There are at most L choices for i such that $\mathrm{j} < \mathrm{i} \leq L$. For fixed i, there are at most g choices for a pair $(y_{\mathrm{i}}, y_{\mathrm{i}+1})$ such that $v \in P^{y_{\mathrm{i}},y_{\mathrm{i}+1}}$ since each vertex appears in at most g paths within $\mathcal{P}$. For fixed i and $(y_{\mathrm{i}}, y_{\mathrm{i}+1})$, there are $n^{L-\mathrm{j}-1}$ tuples of the form $(y_{\mathrm{j}+1}, \ldots, y_{\mathrm{i}-1}, y_{\mathrm{i}+2}, \ldots, y_{L+1})$. That is, the total count is at most $L \cdot g \cdot n^{L-\mathrm{j}-1}$.

The two cases are not mutually exclusive, so we will combine the counts in (1.5) and (1.6) by summing them. Then we obtain the following upper bound for the inner summation of $q(\mathcal{Z})$:

$$\sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} \leq n^{L+1} + 2 \cdot \sum_{\mathrm{j}=1}^{L} n^{\mathrm{j}} \cdot \left(\mathfrak{q}_v(x_{\mathrm{j}}) \cdot n^{L-\mathrm{j}} + L \cdot g \cdot n^{L-\mathrm{j}-1} \right) .$$

Since sequence $\mathbf{x}$ is good, it has no repeated vertices. Thus the elements of $\mathbf{x}$ represent a subset of $[n]$. This yields the inequality $\sum_{\mathrm{j}=1}^{L} \mathfrak{q}_v(x_{\mathrm{j}}) \leq \sum_{u \in [n]} \mathfrak{q}_v(u) \leq g$ for any vertex v, where g is the vertex congestion of $\mathcal{P}$ and $\mathfrak{q}_v(u)$ is the number of paths in $\mathcal{P}$ that start at u and contain v. We thus obtain the next bound on the inner summation of $q(\mathcal{Z})$:

$$\sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} \leq n^{L+1} + 2 \cdot L \cdot n^{L} \cdot g \cdot \left(1 + \frac{L \cdot g}{n}\right) .$$

Finally, since $n \leq g$, we get $q(\mathcal{Z}) \in O(|\mathcal{Z}| \cdot g \cdot n^L \cdot (1 + L^2/n))$.

Wrapping up.

Since we showed $M(\mathcal{Z}) \in \Omega(|\mathcal{Z}| \cdot L \cdot n^{L+1})$ and $q(\mathcal{Z}) \in O(|\mathcal{Z}| \cdot g \cdot n^L \cdot (1 + L^2/n))$ for arbitrary $\mathcal{Z}$ with $q(\mathcal{Z}) > 0$, our randomized query complexity bound Eq. (1.4) yields

$$\Omega\left(\min_{\substack{\mathcal{Z} \subseteq \mathcal{X}: \\ q(\mathcal{Z}) > 0}} \frac{M(\mathcal{Z})}{q(\mathcal{Z})} \right) \subseteq \Omega\left(\frac{|\mathcal{Z}| \cdot L \cdot n^{L+1}}{|\mathcal{Z}| \cdot g \cdot n^L \cdot (1 + L^2/n)} \right) \subseteq \Omega\left(\frac{n \cdot L}{g \cdot (1 + L^2/n)} \right) .$$

Setting $L \approx \sqrt{n}$ gives a query complexity of local search of $\Omega(n^{1.5}/g)$.

1.6.2 Corollary for expanders

To obtain the lower bound for expanders from Corollary 1, we use a result from [19]. Their work shows that constant-degree constant-expansion graphs have an all-pairs set of paths with vertex congestion $g \in O(n \ln n)$; see Section 1.8 for details. Theorem 1.3.2 then implies the randomized query complexity of local search on such graphs is $\Omega\left(\frac{\sqrt{n}}{\log n}\right)$.

1.7 Lower bound for local search via separation number

We briefly discuss how to obtain the lower bound of $\Omega\left(\sqrt[4]{\frac{s}{\Delta}}\right)$ of Theorem 1.3.3, where s is the separation number and Δ the maximum degree of the graph. For details, see Section 1.12.

We apply Theorem 1.3.1 with a similar strategy as the one discussed in Section 1.6 of lower bounding $M(\mathcal{Z})$ and upper bounding $q(\mathcal{Z})$ for arbitrary subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$.

However, we use a slightly different r function and now construct staircases with respect to another graph-theoretic notion known as "path arrangement parameter" and "cluster walks" (instead of using a pre-defined set of pairs $\mathcal{P}$ with congestion g). Our construction is heavily inspired by that in [17]. While the construction is highly similar, using it within Theorem 1.3.1 is non-trivial.

1.8 Theorems from prior work

In this section we include several theorems from prior work.

The first is a result from [19] about systems of paths with low congestion for expanders.

Theorem 1.8.1 ([19], Theorem 1)**.** *Let $G = (V, E)$ be a d-regular β-expander where $d \in \mathbb{N}$ and $\beta \in \mathbb{R}_+$ are constant. Let $\alpha : \mathbb{N} \to \mathbb{R}_+$ be a function. Consider a collection of $K = \alpha(n)n/\ln(n)$ pairs of vertices denoted $\{(a_1, b_1), \ldots, (a_K, b_K)\}$ such that no vertex in V participates in more than s pairs.*

Then there is a set of K paths $\{P_1, \ldots, P_K\}$ such that P_i connects a_i to b_i and the congestion on each edge is at most

$$g = \begin{cases} O\left(s + \left\lceil \frac{\ln \ln n}{\ln(1/\min\{\alpha, 1/\ln\ln n\})} \right\rceil\right) & \text{for } \alpha < \frac{1}{2}\,. \\ O(s + \alpha + \ln \ln n) & \text{for } \alpha \geq \frac{1}{2}\,. \end{cases} \tag{1.7}$$

Next is a result from [48] on multi-commodity flow, which gives a corollary for finding systems of paths with low congestion for expanders. We state the corollary as described in [27].

Theorem 1.8.2 ([27], Corollary C.2)**.** *Let $G = ([n], E)$ be a β-expander with maximum vertex degree Δ and let M be any partial matching over the vertices of G. Then there is an efficient randomized algorithm that finds, for every pair $(u, v) \in M$, a set $\mathcal{P}_{u,v}$ of $\lceil \ln n \rceil$ paths of length $O(\Delta \cdot \ln(n)/\beta)$ each, such that the set $\mathcal{P} = \bigcup_{(u,v) \in M} \mathcal{P}_{u,v}$ has edge congestion $O(\ln^2(n)/\beta)$. The algorithm succeeds with high probability.*

Next we present a lemma reducing local search to a decision problem. This reduction is not new; see for example [16].

Lemma 2. *Suppose the randomized query complexity of local search on G is χ, recalling that in local search we have a graph G and a function $f : V \to \mathbb{R}$, and the problem is to find a local minimum. Then the query complexity of the following decision problem is at most $\chi + 1$:*

- *Input: graph G and oracle access to a function $h_b : [n] \to \mathbb{R} \times \{-1, 0, 1\}$ for some $b \in \{0, 1\}$ with the property that $h_b(v) = (f(v), -1)$ when v is not a local minimum of f, and $h_b(v) = (f(v), b)$ when v is a local minimum of f.*

- *Output: the bit b.*

To clarify, the algorithm is given oracle access to h_b, but it is not given b itself.

Proof. Let Γ be a randomized algorithm that can solve local search on G such that

- Γ has success probability at least p on every input $\langle G, h_b \rangle$.
- Γ issues at most χ queries in expectation.

We will use the local search algorithm Γ to construct an algorithm Γ_d that solves the decision problem. To simulate a query on $f(v)$ for Γ, algorithm Γ_d will query the function h_b at v, obtain $(f(v), c)$, with $c \in \{-1, 0, 1\}$, and will pass $f(v)$ to Γ. Whenever Γ locates a local minimum $v_{\min}$, we know that $h_b(v_{\min})$ contains the hidden bit output required by Γ_d. Obtaining that hidden bit then requires only one additional query, at $v_{\min}$. Since Γ locates a local minimum $v_{\min}$ with χ queries in expectation and succeeds with probability at least p, we see that Γ_d uses $\chi + 1$ queries in expectation and succeeds with probability at least p. $\square$

1.9 The relational adversary method and our variant

In this section we show our variant of the original relational adversary method from [3]. First, we introduce some preliminaries. Consider any two functions $f, g : A \to B$, for some sets A, B. An element $a \in A$ is said to *distinguish* the function f from g if $f(a) \neq g(a)$.

Next, we include the original statement from [3], written with our notation.

Theorem 1.9.1 ([3], Theorem 5)**.** *Consider finite sets A or B. Let $\mathcal{H} : B^A \to \{0,1\}$ be a map that labels each function $F : A \to B$ with 0 or 1. Let $\mathcal{A} \subseteq \mathcal{H}^{-1}(0)$ and $\mathcal{B} \subseteq \mathcal{H}^{-1}(1)$. The problem is: given $A, B, \mathcal{H}, \mathcal{A}, \mathcal{B}$, and oracle access to a function F^* [9] from $\mathcal{A}$ or $\mathcal{B}$, return the label $\mathcal{H}(F^*)$.*

Let $r : \mathcal{A} \times \mathcal{B} \to \mathbb{R}_{\geq 0}$ be a non-zero real-valued function of our choice. For $F_1 \in \mathcal{A}, F_2 \in \mathcal{B}$, and $a \in A$, define

$$\theta(F_1, a) = \frac{\sum_{F_3 \in \mathcal{B}\,:\,F_1(a) \neq F_3(a)} r(F_1, F_3)}{\sum_{F_3 \in \mathcal{B}} r(F_1, F_3)} \qquad \textit{and} \qquad \theta(F_2, a) = \frac{\sum_{F_3 \in \mathcal{A}\,:\,F_2(a) \neq F_3(a)} r(F_3, F_2)}{\sum_{F_3 \in \mathcal{A}} r(F_3, F_2)} \tag{1.8}$$

[9]In other words, we have free access to $\mathcal{H}$ and the only queries counted are the ones to F^*, which will be of the form: "What is $F^*(a)$?", for some $a \in A$. The oracle will return $F^*(a)$ in one computational step.

whenever the denominators in (1.8) are all nonzero. Then the randomized query complexity [10] *of the problem is* $1/(5 \cdot v_{min})$*, where*

$$v_{min} = \max_{F_1 \in \mathcal{A}, F_2 \in \mathcal{B}, a \in A \,:\, r(F_1,F_2)>0, F_1(a) \neq F_2(a)} \min\Big\{\theta(F_1, a), \theta(F_2, a)\Big\}.$$

The proof centers on a difficult input distribution under which the denominator of $\theta(F_1, a)$ (respectively $\theta(F_2, a)$) is the likelihood that F_1 (respectively F_2) is sampled, conditioned on the input being sampled from $\mathcal{A}$ (respectively $\mathcal{B}$). The numerator of $\theta(F_1, a)$ (respectively $\theta(F_2, a)$) is the progress that is made via querying a towards distinguishing F_1 (respectively F_2) from the other functions.

Our variant uses the same framework of relating pairs of inputs through some "relation" r, but the lower bound expression is based on another average type of argument.

1.9.1 Our variant of the relational adversary method

Now we restate our variant of the relational adversary method.

Theorem 1.3.1. *Consider finite sets A and B, a set $\mathcal{X} \subseteq B^A$ of functions* [11]*, and a map $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ which assigns a label to each function in $\mathcal{X}$. Additionally, we get oracle access to an unknown function $F^* \in \mathcal{X}$. The problem is to compute $\mathcal{H}(F^*)$ using as few queries to F^* as possible.*[12]

Let $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a non-zero symmetric function of our choice with $r(F_1, F_2) = 0$ whenever $\mathcal{H}(F_1) = \mathcal{H}(F_2)$. For each $\mathcal{Z} \subseteq \mathcal{X}$, define

$$M(\mathcal{Z}) = \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2)\,; \quad \textit{and} \quad q(\mathcal{Z}) = \max_{a \in A} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}}\,. \quad (1.1)$$

[10]↑Recall we defined the randomized query complexity as the expected number of queries required to achieve success probability at least 9/10.
[11]↑Each function $F \in \mathcal{X}$ has the form $F : A \to B$.
[12]↑In other words, we have free access to $\mathcal{H}$ and the only queries counted are the ones to F^*, which will be of the form: "What is $F^*(a)$?", for some $a \in A$. The oracle will return $F^*(a)$ in one computational step.

If there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$, then the randomized query complexity of the problem is at least

$$\min_{\mathcal{Z} \subseteq \mathcal{X} : q(\mathcal{Z}) > 0} \frac{M(\mathcal{Z})}{100 \cdot q(\mathcal{Z})} \,. \tag{1.2}$$

Let us briefly interpret Theorem 1.3.1. We use Yao's lemma, and so it will suffice to design a "hard" distribution of input functions and analyze the performance of a deterministic algorithm when given inputs from this distribution.

The theorem considers the probability distribution P where each function $F \in \mathcal{X}$ is given as input with probability $P(F) = \frac{M(\{F\})}{M(\mathcal{X})}$, where $\mathcal{X}$ is the space of possible functions.

The quantity $M(\mathcal{Z})$ is the likelihood that the function F^* sampled from this distribution lies in $\mathcal{Z}$. The quantity $q(\mathcal{Z})$ is the largest amount of progress possible in a single query once the algorithm already knows that the given function F^* lies in $\mathcal{Z}$.

Now we are ready to prove Theorem 1.3.1.

Proof of Theorem 1.3.1. Given a relation r with the properties required by the theorem, define

$$M(\{F_1\}) = \sum_{F_2 \in \mathcal{X}} r(F_1, F_2) \;\; \forall F_1 \in \mathcal{X} \quad \text{and} \quad M(\mathcal{X}) = \sum_{F_1 \in \mathcal{X}} M(\{F_1\}) \,. \tag{1.9}$$

We consider the distribution P over functions in $\mathcal{X}$ that selects each function $F \in \mathcal{X}$ with probability $P(F) = M(\{F\})/M(\mathcal{X})$. The theorem claims a lower bound when there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$, so we may assume such a subset $\mathcal{Z}$ exists. By Lemma 3, this implies $M(\mathcal{Z}) > 0$, and so

$$M(\mathcal{X}) > 0 \,. \tag{1.10}$$

Thus $P(F) = M(\{F\})/M(\mathcal{X})$ is well defined.

We say an algorithm succeeds on an input function F if it outputs the correct label $\mathcal{H}(F)$. Let R be the randomized query complexity (on the worst case input $F \in \mathcal{X}$) for success probability 19/20. Let Γ_d be the best deterministic algorithm [13] that succeeds with probability at least 9/10 when the input is a random function drawn from distribution P. Let D be the expected number of queries issued by Γ_d on input distribution P. Yao's lemma ([49], Theorem 3) yields $2R \geq D$. Thus to lower bound T, it will suffice to lower bound D. Let $T = 10D$.

Let Γ^* be the truncation of Γ_d after T queries. We will analyze the expected number of queries made by Γ^* when facing distribution P. Let X be the random variable representing the number of queries issued by Γ_d. Then $\mathbb{E}[X] = D$. By Markov's inequality,

$$\Pr[X \geq T] \leq \frac{\mathbb{E}[X]}{T} = \frac{D}{T} = \frac{1}{10}. \tag{1.11}$$

We have

$$\begin{aligned}\Pr[X > T] + &\Pr[\Gamma_d \text{ succeeds and } X \leq T] \\ &\geq \Pr[\Gamma_d \text{ succeeds and } X > T] + \Pr[\Gamma_d \text{ succeeds and } X \leq T] \\ &= \Pr[\Gamma_d \text{ succeeds}]. \end{aligned} \tag{1.12}$$

Then

$$\begin{aligned}\Pr[\Gamma^* \text{ succeeds}] &= \Pr[\Gamma_d \text{ succeeds and } X \leq T] \\ &\geq \Pr[\Gamma_d \text{ succeeds}] - \Pr[X > T] && \text{(By (1.12))} \\ &\geq 9/10 - 1/10 = 4/5 && \text{(By choice of } \Gamma_d \text{ and (1.11))}\end{aligned}$$

[13]↑That is, with smallest expected number of queries possible.

Algorithm Γ^* is said to *distinguish* $F_1 \in \mathcal{X}$ from $F_2 \in \mathcal{X}$ within the first t queries if Γ^* queries an element $a \in A$ with the property that $F_1(a) \neq F_2(a)$ within the first t queries. For all $t \in \mathbb{N}$, $F_1 \in \mathcal{X}$, and $F_2 \in \mathcal{X}$, define

$$I^{(t)}(F_1, F_2) = \begin{cases} 1 & \text{if algorithm } \Gamma^* \text{ distinguishes } F_1 \text{ from } F_2 \text{ within the first } t \text{ queries} \\ 0 & \text{otherwise}\,. \end{cases}$$

For each function $F_1 \in \mathcal{X}$ and index $t \in \{0, \ldots, T\}$, we define a "local progress measure" $S^{(t)}(F_1)$ that counts the number of elements $F_2 \in \mathcal{X}$ distinguished from F_1 by the t-th query, weighted by the relation $r(F_1, F_2)$. Formally, for each $F_1 \in \mathcal{X}$ and $0 \leq t \leq T$, let

$$S^{(t)}(F_1) = \sum_{\substack{F_2 \in \mathcal{X}: \\ I^{(t)}(F_1, F_2) = 1}} r(F_1, F_2)\,.$$

Summing over all the functions in $\mathcal{X}$, we obtain a "global progress measure" $S^{(t)}$:

$$S^{(t)} = \sum_{F_1 \in \mathcal{X}} S^{(t)}(F_1)\,.$$

The difference in progress between consecutive queries can then be defined as:

$$\Delta S^{(t)} = S^{(t)} - S^{(t-1)}\,. \tag{1.13}$$

To lower bound the progress, we show in three steps that (a) the initial value of the progress measure, $S^{(0)}$, is zero; (b) the final value of the progress measure, $S^{(T)}$, is "large"; and (c) the difference in the progress measure between consecutive rounds, $\Delta S^{(t)}$, is "small" for each round t.

Step (a)

No queries have been issued at time zero, so nothing has been distinguished. Therefore

$$S^{(0)} = 0\,. \tag{1.14}$$

Step (b)

Define $\mathcal{W} = \{F_1 \in \mathcal{X} \mid \Gamma^*(F_1) = \mathcal{H}(F_1)\}$. This is the set of functions on which Γ^* succeeds, i.e. finds the correct label.

We claim algorithm Γ^* distinguishes each pair of functions $F_1, F_2 \in \mathcal{W}$ for which $\mathcal{H}(F_1) \neq \mathcal{H}(F_2)$. That is, Γ^* must have queried an index $a \in A$ such that $F_1(a) \neq F_2(a)$ within the first T queries. To see this, suppose towards a contradiction there exists a pair $F_1, F_2 \in \mathcal{W}$ with $\mathcal{H}(F_1) \neq \mathcal{H}(F_2)$ such that Γ^* only queries indices $a \in A$ with $F_1(a) = F_2(a)$. Then, Γ^* cannot differentiate whether the input function is F_1 or F_2, and so must have the same output $\Gamma^*(F_1) = \Gamma^*(F_2)$ since Γ^* is deterministic. Thus Γ^* makes a mistake on one of F_1 or F_2 because $\mathcal{H}(F_1) \neq \mathcal{H}(F_2)$. This contradicts the choice of $F_1, F_2 \in \mathcal{W}$ as inputs on which Γ^* is successful. Thus, the algorithm Γ^* distinguishes each pair of inputs $F_1, F_2 \in \mathcal{W}$ within the first T queries.

Formally, we have

$$I^{(T)}(F_1, F_2) = 1, \qquad \text{for all } F_1, F_2 \in \mathcal{W} \text{ where } \mathcal{H}(F_1) \neq \mathcal{H}(F_2). \tag{1.15}$$

Moreover, since Γ^* succeeds when the input is a function from $\mathcal{W}$ and fails otherwise, the success probability on input distribution P is at least $4/5$, and the distribution P samples each function $F_1 \in \mathcal{X}$ with probability $P(F_1) = M(\{F_1\})/M(\mathcal{X})$, we have

$$\sum_{F_1 \in \mathcal{W}} \frac{M(\{F_1\})}{M(\mathcal{X})} = \sum_{F_1 \in \mathcal{W}} P(F_1) \geq 4/5 \quad \text{and} \quad \sum_{F_2 \in \mathcal{X} \setminus \mathcal{W}} \frac{M(\{F_2\})}{M(\mathcal{X})} \leq 1/5\,. \tag{1.16}$$

Thus,

$$\begin{aligned}
S^{(T)} &= \sum_{\substack{F_1,F_2\in\mathcal{X}:\\ I^{(T)}(F_1,F_2)=1}} r(F_1,F_2) && \text{(By definition of } S^{(T)}\text{)}\\
&\geq \sum_{\substack{F_1,F_2\in\mathcal{W}:\\ I^{(T)}(F_1,F_2)=1}} r(F_1,F_2) && \text{(Since } \mathcal{W}\subseteq\mathcal{X} \text{ and } r \text{ is non-negative)}\\
&= \sum_{F_1,F_2\in\mathcal{W}} r(F_1,F_2) && \text{(By Eq. (1.15) and } r(F_1,F_2)=0 \text{ if } \mathcal{H}(F_1)=\mathcal{H}(F_2)\text{)}\\
&= \sum_{F_1\in\mathcal{W},F_2\in\mathcal{X}} r(F_1,F_2) - \sum_{F_1\in\mathcal{W},F_2\in\mathcal{X}\setminus\mathcal{W}} r(F_1,F_2)\,. && (1.17)
\end{aligned}$$

By definition of M, we have $\sum_{F_1\in\mathcal{W},F_2\in\mathcal{X}} r(F_1,F_2) = \sum_{F_1\in\mathcal{W}} M(\{F_1\})$, which substituted in (1.17) yields

$$S^{(T)} \geq \sum_{F_1\in\mathcal{W}} M(\{F_1\}) - \sum_{F_1\in\mathcal{W},F_2\in\mathcal{X}\setminus\mathcal{W}} r(F_1,F_2)\,. \tag{1.18}$$

Since $\mathcal{W}\subseteq\mathcal{X}$ and r is non-negative, we have

$$\sum_{F_1\in\mathcal{W},F_2\in\mathcal{X}\setminus\mathcal{W}} r(F_1,F_2) \leq \sum_{F_1\in\mathcal{X},F_2\in\mathcal{X}\setminus\mathcal{W}} r(F_1,F_2)\,. \tag{1.19}$$

Furthermore, using the symmetry of r and the definition of $M(\{F_2\})$, we can rewrite the right hand side of (1.19) as

$$\begin{aligned}
\sum_{F_1\in\mathcal{X},F_2\in\mathcal{X}\setminus\mathcal{W}} r(F_1,F_2) &= \sum_{F_1\in\mathcal{X},F_2\in\mathcal{X}\setminus\mathcal{W}} r(F_2,F_1) && \text{(By symmetry of } r\text{)}\\
&= \sum_{F_2\in\mathcal{X}\setminus\mathcal{W}} M(\{F_2\})\,. && (1.20)
\end{aligned}$$

Combining (1.19) and (1.20) yields

$$\sum_{F_1\in\mathcal{W},F_2\in\mathcal{X}\setminus\mathcal{W}} r(F_1,F_2) \leq \sum_{F_2\in\mathcal{X}\setminus\mathcal{W}} M(\{F_2\})\,. \tag{1.21}$$

Combining (1.18) and (1.21), we obtain

$$\begin{aligned} S^{(T)} &\geq \sum_{F_1 \in \mathcal{W}} M(\{F_1\}) - \sum_{F_2 \in \mathcal{X} \setminus \mathcal{W}} M(\{F_2\}) \\ &\geq \frac{4}{5} M(\mathcal{X}) - \frac{1}{5} M(\mathcal{X}) = \frac{3}{5} M(\mathcal{X})\,. && \text{(By Eq. (1.16))} \end{aligned}$$

Therefore,

$$S^{(T)} \geq 3 \cdot M(\mathcal{X})/5\,. \tag{1.22}$$

Step (c)

By Lemma 4, for each $t \in [T]$, the maximum progress made by the t-th query, $\Delta S^{(t)}$, can be bounded as follows:

$$\Delta S^{(t)} \leq M(\mathcal{X}) \cdot \max_{\mathcal{Z} \subseteq \mathcal{X}\colon M(\mathcal{Z})>0} \frac{q(\mathcal{Z})}{M(\mathcal{Z})}\,. \tag{1.23}$$

Combining steps (a,b,c).

We obtain:

$$\begin{aligned} \frac{3}{5} \cdot M(\mathcal{X}) &\leq S^{(T)} && \text{(By Eq. (1.22))} \\ &= S^{(0)} + \sum_{t=1}^{T} \Delta S^{(t)} && \text{(By definition of } S^{(T)}\text{)} \\ &= 0 + \sum_{t=1}^{T} \Delta S^{(t)} && \text{(Since } S^{(0)} = 0 \text{ by Eq. (1.14))} \\ &\leq T \cdot M(\mathcal{X}) \cdot \max_{\mathcal{Z} \subseteq \mathcal{X}\colon M(\mathcal{Z})>0} \frac{q(\mathcal{Z})}{M(\mathcal{Z})} && \text{(By Eq. (1.23))} \\ &= T \cdot M(\mathcal{X}) \cdot \max_{\mathcal{Z} \subseteq \mathcal{X}\colon M(\mathcal{Z})>0 \text{ and } q(\mathcal{Z})>0} \frac{q(\mathcal{Z})}{M(\mathcal{Z})} \\ & && \text{(Since there exists } \mathcal{Z} \subseteq \mathcal{X} \text{ with } q(\mathcal{Z}) > 0.\text{)} \end{aligned}$$

$$= T \cdot M(\mathcal{X}) \cdot \max_{\mathcal{Z} \subseteq \mathcal{X}:\, q(\mathcal{Z})>0} \frac{q(\mathcal{Z})}{M(\mathcal{Z})}\,. \qquad \text{(Since } q(\mathcal{Z}) > 0 \text{ implies } M(\mathcal{Z}) > 0.\text{)}$$

Rearranging and denoting $\zeta = \min_{\mathcal{Z} \subseteq \mathcal{X}:\, q(\mathcal{Z})>0} \frac{M(\mathcal{Z})}{q(\mathcal{Z})}$, we get

$$T \geq \frac{3}{5} \cdot \frac{1}{\max_{\mathcal{Z} \subseteq \mathcal{X}:\, q(\mathcal{Z})>0} \frac{q(\mathcal{Z})}{M(\mathcal{Z})}} = \frac{3\zeta}{5}\,. \tag{1.24}$$

In summary, the randomized query complexity R when the success probability is 19/20 can be bounded as follows:

$$R \geq \frac{D}{2} = \frac{T}{20} \geq \frac{3\zeta}{100}\,. \tag{1.25}$$

Let $\mathcal{A}$ be an arbitrary randomized algorithm that succeeds with probability at least 9/10. If $\mathcal{A}$ issues fewer than $\zeta/100$ queries in expectation, then $\mathcal{A}$ could be repeated three times and the majority answer returned; this would achieve a greater than 19/20 probability of success and fewer than $3\zeta/100$ queries in expectation, which would contradict Eq. (1.25). Thus $\mathcal{A}$ issues at least $\zeta/100$ queries in expectation on its worst case input. Thus the randomized query complexity is lower bounded as required by the theorem statement. ☐

Lemma 3. *Let $\mathcal{Z} \subseteq \mathcal{X}$ be a subset with $q(\mathcal{Z}) > 0$. Then $M(\mathcal{Z}) > 0$.*

Proof. By definition of $M(\mathcal{Z})$, we have

$$\begin{aligned} M(\mathcal{Z}) &= \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2) \\ &\geq \max_{v \in [n]} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} && \text{(Since } \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} \leq 1 \text{ and } r(F_1, F_2) \geq 0.\text{)} \\ &= q(\mathcal{Z})\,. && \text{(By definition of } q(\mathcal{Z}).\text{)} \end{aligned}$$

Since $\mathcal{Z}$ was chosen such that $q(\mathcal{Z}) > 0$ and $M(\mathcal{Z}) \geq q(\mathcal{Z})$, we conclude that $M(\mathcal{Z}) > 0$ as required. ☐

Lemma 4. *In the setting of step (c) of Theorem 1.3.1, for each $t \in [T]$, we have*

$$\Delta S^{(t)} \leq M(\mathcal{X}) \cdot \max_{\mathcal{Z} \subseteq \mathcal{X} \colon M(\mathcal{Z}) > 0} \frac{q(\mathcal{Z})}{M(\mathcal{Z})} .$$

Proof. Let $\Psi = A^{t-1} \times B^{t-1}$ be the set of possible sequences of the first $t-1$ queries and their answers; each $\psi \in \Psi$ a "history". Let $\psi_1, \psi_2, \ldots, \psi_k$ be the histories in Ψ, where $k = |A|^{t-1} \cdot |B|^{t-1}$.

For all $\mathrm{i} \in [k]$, let $\mathcal{X}_\mathrm{i} \subseteq \mathcal{X}$ be the set of inputs on which Γ^* would have history ψ_i for the first $t-1$ queries; this is well-defined because Γ^* is deterministic. In other words, we can partition $\mathcal{X}$ into equivalence classes $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ so that two inputs have the same history over the first $t-1$ queries if and only if they are in the same equivalence class. This induces three useful facts:

Fact 1.

For each $F_1, F_2 \in \mathcal{X}_\mathrm{i}$ for some i, we know that F_1 and F_2 must receive the same t-th query since Γ^* is deterministic. Let $a_\mathrm{i}^{(t)} \in A$ be the location of this query. For $\mathrm{i} \neq \mathrm{j}$, we may have $a_\mathrm{i}^{(t)} \neq a_\mathrm{j}^{(t)}$; i.e. each history may have a different query in the same round.

For each $F_1 \in \mathcal{X}_\mathrm{i}$ and $F_2 \in \mathcal{X}_\mathrm{j}$ with $\mathrm{i} \neq \mathrm{j}$, functions F_1 and F_2 have already been distinguished since they have different histories. This implies that all future progress must come from pairs from the same equivalence class. For all $a \in A$ and $F_1, F_2 \in \mathcal{X}$, let us define

$$r_a(F_1, F_2) = r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}} .$$

Then

$$\begin{aligned} \Delta S^{(t)} &= S^{(t)} - S^{(t-1)} \\ &= \sum_{F_1 \in \mathcal{X}} \left(S^{(t)}(F_1) - S^{(t-1)}(F_1) \right) && \text{(By definition of } S^{(t)} \text{ and } S^{(t-1)}.) \end{aligned}$$

$$= \sum_{F_1 \in \mathcal{X}} \sum_{\substack{F_2 \in \mathcal{X} \\ I^{(t)}(F_1,F_2)=1 \\ I^{(t-1)}(F_1,F_2)=0}} r(F_1, F_2) \qquad \text{(By definition of } S^{(t)}(F_1) \text{ and } S^{(t-1)}(F_1)\text{)}$$

$$= \sum_{i=1}^{k} \sum_{F_1 \in \mathcal{X}_i} \left(\sum_{\substack{F_2 \in \mathcal{X}_i \\ I^{(t)}(F_1,F_2)=1 \\ I^{(t-1)}(F_1,F_2)=0}} r(F_1, F_2) + \sum_{\substack{F_2 \notin \mathcal{X}_i \\ I^{(t)}(F_1,F_2)=1 \\ I^{(t-1)}(F_1,F_2)=0}} r(F_1, F_2) \right)$$

$$= \sum_{i=1}^{k} \sum_{F_1 \in \mathcal{X}_i} \sum_{\substack{F_2 \in \mathcal{X}_i \\ I^{(t)}(F_1,F_2)=1}} r(F_1, F_2)$$

(By definition of $\mathcal{X}_i$, each function F_2 with $I^{(t-1)}(F_1, F_2) = 0$ is in $\mathcal{X}_i$.)

$$= \sum_{i=1}^{k} \sum_{F_1 \in \mathcal{X}_i} \sum_{F_2 \in \mathcal{X}_i} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a_i^{(t)}) \neq F_2(a_i^{(t)})\}}$$

(Since F_1 and F_2 are distinguished at time t, when query $a_i^{(t)}$ is issued.)

$$= \sum_{i=1}^{k} \sum_{F_1 \in \mathcal{X}_i} \sum_{F_2 \in \mathcal{X}_i} r_{a_i^{(t)}}(F_1, F_2)\,. \tag{1.26}$$

Fact 2.

Since $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k$ is a partition of $\mathcal{X}$, we have

$$M(\mathcal{X}) = \sum_{i=1}^{k} M(\mathcal{X}_i)\,. \tag{1.27}$$

Fact 3.

For all $i \in [k]$, we have

$$M(\mathcal{X}_i) = \sum_{F_1 \in \mathcal{X}_i} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2) \geq \sum_{F_1 \in \mathcal{X}_i} \sum_{F_2 \in \mathcal{X}_i} r_{a_i^{(t)}}(F_1, F_2)\,.$$

Therefore, since r is non-negative,

$$\left(M(\mathcal{X}_{\mathrm{i}}) = 0\right) \implies \left(\sum_{F_1 \in \mathcal{X}_{\mathrm{i}}} \sum_{F_2 \in \mathcal{X}_{\mathrm{i}}} r_{a_{\mathrm{i}}^{(t)}}(F_1, F_2) = 0\right). \tag{1.28}$$

Combining facts 1, 2, and 3.

We get

$$\begin{aligned}
\frac{\Delta S^{(t)}}{M(\mathcal{X})} &= \frac{\sum_{\mathrm{i}\in[k]} \sum_{F_1\in\mathcal{X}_{\mathrm{i}}} \sum_{F_2\in\mathcal{X}_{\mathrm{i}}} r_{a_{\mathrm{i}}^{(t)}}(F_1,F_2)}{\sum_{\mathrm{i}\in[k]} M(\mathcal{X}_{\mathrm{i}})} && \text{(From Eq. (1.26) and Eq. (1.27))} \\
&= \frac{\sum_{\substack{\mathrm{i}\in[k]:\\ M(\mathcal{X}_{\mathrm{i}})>0}} \left(\sum_{F_1\in\mathcal{X}_{\mathrm{i}}} \sum_{F_2\in\mathcal{X}_{\mathrm{i}}} r_{a_{\mathrm{i}}^{(t)}}(F_1,F_2)\right)}{\sum_{\substack{\mathrm{i}\in[k]:\\ M(\mathcal{X}_{\mathrm{i}})>0}} M(\mathcal{X}_{\mathrm{i}})} && \text{(By Eq. (1.28))} \\
&\le \max_{\substack{\mathrm{i}\in[k]:\\ M(\mathcal{X}_{\mathrm{i}})>0}} \frac{\sum_{F_1\in\mathcal{X}_{\mathrm{i}}} \sum_{F_2\in\mathcal{X}_{\mathrm{i}}} r_{a_{\mathrm{i}}^{(t)}}(F_1,F_2)}{M(\mathcal{X}_{\mathrm{i}})} && \text{(Maximizing over } \mathrm{i}\in[k] \text{ with } M(\mathcal{X}_{\mathrm{i}})>0\text{)} \\
&\le \max_{\substack{\mathcal{Z}\subseteq\mathcal{X},\, a\in A:\\ M(\mathcal{Z})>0}} \frac{\sum_{F_1\in\mathcal{Z}} \sum_{F_2\in\mathcal{Z}} r_a(F_1,F_2)}{M(\mathcal{Z})} && \text{(Maximizing over } \mathcal{Z}\subseteq\mathcal{X} \text{ and } a\in A\text{)} \\
&= \max_{\mathcal{Z}\subseteq\mathcal{X}:\, M(\mathcal{Z})>0} \frac{q(\mathcal{Z})}{M(\mathcal{Z})}. && \text{(By definition of } q(\mathcal{Z})\text{)}
\end{aligned}$$

Therefore $\Delta S^{(t)} \le M(\mathcal{X}) \cdot \max_{\mathcal{Z}\subseteq\mathcal{X}:\, M(\mathcal{Z})>0} \frac{q(\mathcal{Z})}{M(\mathcal{Z})}$, as required by the lemma. □

1.9.2 Matrix game

In this section, we describe a toy problem which highlights the advantage of Theorem 1.3.1 over Theorem 1.9.1.

Setup

Let $n \in \mathbb{N}$ be a square number and $\mathcal{X}$ be a subset of square $\sqrt{n} \times \sqrt{n}$ matrices with entries from $\{0, 1, 2\}$. There are two types of matrices within $\mathcal{X}$: "row" matrices and "column" matrices. Row matrices have one row of 1s with all other entries 0 while column matrices have one column of 2s with all other entries 0. For example:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ is a row matrix; } \qquad \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix} \text{ is a column matrix.}$$

So, $|\mathcal{X}| = 2\sqrt{n}$ since there are $\sqrt{n}$ distinct row matrices and $\sqrt{n}$ distinct column matrices.

The game

Given n and oracle access to a matrix $F \in \mathcal{X}$, the goal is to correctly declare whether F is a row or column matrix.

One can check that $\sqrt{n}$ queries suffices by querying the main diagonal: if any "1" is detected, declare "row"; if any "2" is found, declare "column". Even with randomization, one intuitively expects that $\Omega(\sqrt{n})$ queries are necessary. In fact, this is what we can show using Theorem 1.3.1.

Lemma 1. *The randomized query complexity of the matrix game is* $\Theta(\sqrt{n})$.

Proof. The upper bound of $O(\sqrt{n})$ follows by querying the main diagonal: if any "1" is detected, declare "row"; if any "2" is found, declare "column".

To show the lower bound of $\Omega(\sqrt{n})$, we instantiate Theorem 1.3.1 with the following definitions:

- Finite set A is the set of n coordinates within $\sqrt{n} \times \sqrt{n}$ matrix.
- Finite set B is the space of all possible $\sqrt{n} \times \sqrt{n}$ matrices with cell values from $\{0, 1, 2\}$.

- Set $\mathcal{X} \subseteq B^A$ of $\sqrt{n}$ row matrices and $\sqrt{n}$ column matrices, for a total of $2\sqrt{n}$ matrices. So, for any given $F \in \mathcal{X}$ and $a \in A$, we have that $F(a) \in \{0,1,2\}$ is the value of the matrix at the coordinate indicated by a.
- Mapping $\mathcal{H} : \mathcal{X} \to \{0,1\}$ refers to deciding whether the matrix from $\mathcal{X}$ is a row or column matrix: output 0 if "row" and output 1 if "column".
- For any two $F_1, F_2 \in \mathcal{X}$, we define $r(F_1, F_2) = \mathbb{1}_{\{\mathcal{H}(F_1) \neq \mathcal{H}(F_2)\}}$ to be the indicator whether F_1 and F_2 are of the same type.

There exists $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$; in particular, $\mathcal{Z} = \{F_{row}, F_{col}\}$ for any row function F_{row} and column function F_{col} has $q(\mathcal{Z}) = 2$. Therefore we may invoke Theorem 1.3.1.

Under this instantiation, we have $\sum_{F_2 \in \mathcal{X}} r(F_1, F_2) = \sqrt{n}$ for any $F \in \mathcal{X}$. Thus

$$M(\mathcal{Z}) = \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2) = |\mathcal{Z}| \cdot \sqrt{n} \quad \text{for any } \mathcal{Z} \subseteq \mathcal{X} \,. \tag{1.29}$$

Meanwhile, let $\mathcal{Z}_{row} = \{F_1 \in \mathcal{Z} : \mathcal{H}(F_1) = 0\}$ and $\mathcal{Z}_{col} = \{F_2 \in \mathcal{Z} : \mathcal{H}(F_2) = 1\}$. For every $F_1, F_2 \in \mathcal{X}$ with $\mathcal{H}(F_1) \neq \mathcal{H}(F_2)$, we have $F_1(a) \neq F_2(a)$ if and only if a lies on F_1's row/column or F_2's row/column. Furthermore, because for each row/column there is only one corresponding input in $\mathcal{X}$, we have

$$|\{F \in \mathcal{Z} \mid F(v) = 1\}| \leq 1 \quad \text{and} \quad |\{F \in \mathcal{Z} \mid F(v) = 2\}| \leq 1 \,. \tag{1.30}$$

For any arbitrary $\mathcal{Z} \subseteq \mathcal{X}$ and $a \in A$, we have

$$\begin{aligned}
&\sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}} && (1.31)\\
&= \sum_{\substack{F_1 \in \mathcal{Z}: \\ F_1(a) \neq 0}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}} + \sum_{\substack{F_1 \in \mathcal{Z}: \\ F(a) = 0}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} && (1.32)\\
&\leq (|\mathcal{Z}_{row}| + |\mathcal{Z}_{col}|) + \sum_{\substack{F_1 \in \mathcal{Z}: \\ F_1(a) = 0}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}} && \text{(By Eq. (1.30))}
\end{aligned}$$

$$= |\mathcal{Z}_{row}| + |\mathcal{Z}_{col}| + \sum_{\substack{F_1 \in \mathcal{Z}: \\ F_1(a)=0}} \sum_{\substack{F_2 \in \mathcal{Z}: \\ F_2(a)\neq 0}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}} \tag{1.33}$$

$$\leq |\mathcal{Z}_{row}| + |\mathcal{Z}_{col}| + \sum_{\substack{F_1 \in \mathcal{Z}: \\ F_1(a)=0}} 1 \tag{1.34}$$

$$\leq 2 \cdot |\mathcal{Z}| \,. \tag{1.35}$$

In the equations above, Eq. (1.33) holds since $F_1(a) = F_2(a) = 0$ implies $\mathbb{1}_{\{F_1(a)\neq F_2(a)\}} = 0$.

Next we explain why Eq. (1.34) holds. Take arbitrary $F_1, F_2 \in \mathcal{Z}$ with $F_1(a) = 0$ and $F_2(a) \neq 0$. Then $r(F_1, F_2) = 1$ when $\mathcal{H}(F_1) \neq \mathcal{H}(F_2)$ and $r(F_1, F_2) = 0$ when $\mathcal{H}(F_1) = \mathcal{H}(F_2)$. By Eq. (1.30), there are at most two functions $F_2 \in \mathcal{Z}$ with $F_2(a) \neq 0$: one with $\mathcal{H}(F_2) = 0$ and one with $\mathcal{H}(F_2) = 1$. So no matter what $\mathcal{H}(F_1)$ is, there is at most one $F_2 \in \mathcal{Z}$ with $F_2(a) \neq 0$ such that $r(F_1, F_2) = 1$. Thus Eq. (1.34) holds.

Therefore by Eq. (1.35),

$$q(\mathcal{Z}) \leq \max_{a \in A} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a)\neq F_2(a)\}} \leq 2 \cdot |\mathcal{Z}| \quad \text{for any } \mathcal{Z} \subseteq \mathcal{X} \,. \tag{1.36}$$

By Eq. (1.29) and Eq. (1.36), the statement of Theorem 1.3.1 implies the query complexity is at least

$$\Omega\left(\min_{\substack{\mathcal{Z} \subseteq \mathcal{X}: \\ q(\mathcal{Z})>0}} \frac{M(\mathcal{Z})}{q(\mathcal{Z})} \right) \subseteq \Omega\left(\frac{|\mathcal{Z}| \cdot \sqrt{n}}{2 \cdot |\mathcal{Z}|} \right) \subseteq \Omega(\sqrt{n}) \,.$$

□

On the other hand, one can only show that the randomized query complexity of the matrix game is $\Omega(1)$ by using the original relational adversary theorem of [3], illustrating the advantage of our new variant on this matrix game. Intuitively, the reason why Theorem 1.9.1 cannot handle this problem is because *every* pair of row matrix F_{row} and column matrix F_{col} has *some* location x that distinguishes both F_{row} and F_{col} from *many* matrices.

Lemma 5. *Using Theorem 1.9.1 gives a lower bound of $\Omega(1)$ on the randomized query complexity of the matrix game.*

Proof. We instantiate Theorem 1.9.1 with the next parameters:

- The finite set A is the set of n coordinates within $\sqrt{n} \times \sqrt{n}$ matrix.
- The finite set B is $\{0, 1, 2\}$.
- The set $\mathcal{X} \subseteq B^A$ consists of $\sqrt{n}$ row matrices and $\sqrt{n}$ column matrices, for a total of $2\sqrt{n}$ matrices. So, for any given $F \in \mathcal{X}$ and $a \in A$, we have that $F(a) \in \{0, 1, 2\}$ is the value of the matrix at the coordinate indicated by a.
- Mapping $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ refers to deciding whether the matrix from $\mathcal{X}$ is a row or column matrix: output 0 if "row" and output 1 if "column".

Consider an arbitrary choice of $r : \mathcal{A} \times \mathcal{B} \to \mathbb{R}_{\geq 0}$. Consider an arbitrary row matrix $F_{row} \in \mathcal{A}$ and an arbitrary column matrix $F_{col} \in \mathcal{B}$ such that $r(F_{row}, F_{col}) > 0$. Let a_{int} be the unique intersecting coordinate in the matrix within F_{row}'s non-zero row and F_{col}'s non-zero column. That is, $F_{row}(a_{\text{int}}) = 1$ and $F_{col}(a_{\text{int}}) = 2$. Meanwhile, $F(a_{\text{int}}) = 0$ for any other matrix $F \in \mathcal{A}$ or $F \in \mathcal{B}$. Therefore,

$$\theta(F_{row}, a_{\text{int}}) = \frac{\sum_{F_3 \in \mathcal{B} \,:\, F_{row}(a_{\text{int}}) \neq F_3(a_{\text{int}})} r(F_{row}, F_3)}{\sum_{F_3 \in \mathcal{B}} r(F_{row}, F_3)} = \frac{\sum_{F_3 \in \mathcal{B}} r(F_{row}, F_3)}{\sum_{F_3 \in \mathcal{B}} r(F_{row}, F_3)} = 1\,.$$

Similarly, we have $\theta(F_{col}, a_{\text{int}}) = 1$ and we see that

$$\min\{\theta(F_{row}, a_{\text{int}}), \theta(F_{col}, a_{\text{int}})\} = 1\,.$$

Since v_{min} is a maximum over choices of F_{row}, F_{col}, and v, this suffices to show

$$v_{min} \geq 1\,.$$

Thus $1/v_{min} \leq 1$ and since the choice of r was arbitrary, one cannot hope to show a stronger lower bound than $\Omega(1)$ via Theorem 1.9.1 for this problem. $\square$

1.9.3 Our variant is stronger than the original relational adversary method

In fact, we now show that our new variant Theorem 1.3.1 is always at least asymptotically as good as Theorem 1.9.1.

Proposition 1.5.1. *Consider any problem and let T be the expected number of queries required in the worst case by the best randomized algorithm to succeed with probability* $9/10$.

If the relational adversary method from [3] provides a lower bound of $T \geq \Lambda$ for some $\Lambda > 0$, then Theorem 1.3.1 can prove a lower bound of $T \geq \Lambda/40$.

Proof. Recall that for every $a \in A$ and $F_1, F_2 \in \mathcal{X}$, we define $r_a(F_1, F_2) = r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}}$. Theorem 1.9.1 provides a lower bound of $\Lambda = 1/(5 \cdot v_{min})$ for the expected number of queries issued by a randomized algorithm that succeeds with probability at least $9/10$, where

$$v_{min} = \max_{F_1 \in \mathcal{A}, F_2 \in \mathcal{B}, a \in A \,:\, r(F_1,F_2)>0, F_1(a) \neq F_2(a)} \min\{\theta(F_1, a), \theta(F_2, a)\}\,.$$

Then for all $F_1 \in \mathcal{A}, F_2 \in \mathcal{B}, a \in A$ where $r_a(F_1, F_2) > 0$, we have

$$\frac{1}{\min\{\theta(F_1, a), \theta(F_2, a)\}} \geq \frac{1}{v_{min}}\,. \tag{1.37}$$

By rearranging and invoking the definition of θ (Eq. (1.8)) we get

$$\max\left(\frac{\sum_{F_3 \in \mathcal{B}} r(F_1, F_3)}{\sum_{F_3 \in \mathcal{B}} r_a(F_1, F_3)}, \frac{\sum_{F_3 \in \mathcal{A}} r(F_3, F_2)}{\sum_{F_3 \in \mathcal{A}} r_a(F_3, F_2)}\right) \geq \frac{1}{v_{min}}\,. \tag{1.38}$$

By extending r to $\mathcal{X} \times \mathcal{X}$, with $r(F_1, F_2) = 0$ if both $F_1, F_2 \in \mathcal{A}$ or both $F_1, F_2 \in \mathcal{B}$, we can write Eq. (1.38) as:

$$\max \left(\frac{M(\{F_1\})}{\sum_{F_3 \in \mathcal{X}} r_a(F_1, F_3)}, \frac{M(\{F_2\})}{\sum_{F_3 \in \mathcal{X}} r_a(F_2, F_3)} \right) \geq \frac{1}{v_{min}} \,. \tag{1.39}$$

Keeping the same choice of r, take an arbitrary choice of $\mathcal{Z} \subseteq \mathcal{X}$ and $a \in A$ such that $q(\mathcal{Z}) > 0$. Such a choice exists by the following argument: Theorem 1.9.1 provided a bound, so there must exist $F_1 \in \mathcal{A}$ and $F_2 \in \mathcal{B}$ and $a \in A$ with $r(F_1, F_2) > 0$ and $F_1(a) \neq F_2(a)$; the set $\mathcal{Z} = \{F_1, F_2\}$ has $q(\mathcal{Z}) > 0$. Let $\mathcal{C} \subseteq \mathcal{Z}$ be the subset of functions $F_1 \in \mathcal{Z}$ defined as follows:

$$\mathcal{C} = \left\{ F_1 \in \mathcal{Z} : M(\{F_1\}) \geq \frac{1}{v_{min}} \cdot \sum_{F_3 \in \mathcal{X}} r_a(F_1, F_3) \right\} .$$

By Eq. (1.39), we know that for every pair of functions $F_1, F_2 \in \mathcal{Z}$ with $r_a(F_1, F_2) > 0$, at least one of F_1 and F_2 is in $\mathcal{C}$. Therefore

$$\sum_{F_1 \in \mathcal{C}} \sum_{F_2 \in \mathcal{Z}} r_a(F_1, F_2) \geq \frac{1}{2} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r_a(F_1, F_2) = \frac{q(\mathcal{Z})}{2} \,. \tag{1.40}$$

Meanwhile, $M(\mathcal{Z}) \geq M(\mathcal{C})$. Therefore

$$\begin{aligned} \frac{M(\mathcal{Z})}{q(\mathcal{Z})} &\geq \frac{1}{2} \cdot \frac{M(\mathcal{C})}{\sum_{F_1 \in \mathcal{C}} \sum_{F_2 \in \mathcal{Z}} r_a(F_1, F_2)} && \text{(By Eq. (1.40))} \\ &\geq \frac{1}{2} \cdot \frac{\sum_{F_1 \in \mathcal{C}} M(\{F_1\})}{\sum_{F_1 \in \mathcal{C}} \sum_{F_2 \in \mathcal{X}} r_a(F_1, F_2)} \,. && \text{(since } \mathcal{Z} \subseteq \mathcal{X}\text{)} \end{aligned}$$

By an averaging argument, there exists a function $F_1 \in \mathcal{C}$ such that

$$\frac{M(\mathcal{Z})}{q(\mathcal{Z})} \geq \frac{1}{2} \cdot \frac{M(\{F_1\})}{\sum_{F_2 \in \mathcal{X}} r_a(F_1, F_2)} \,. \tag{1.41}$$

But then since $F_1 \in \mathcal{C}$ we get

$$\frac{M(\{F_1\})}{\sum_{F_2 \in \mathcal{X}} r_a(F_1, F_2)} \geq \frac{1}{v_{min}} \,. \tag{1.42}$$

Combining Eq. (1.41) and Eq. (1.42),

$$\frac{M(\mathcal{Z})}{q(\mathcal{Z})} \geq \frac{1}{2} \cdot \frac{M(\{F_1\})}{\sum_{F_2 \in \mathcal{X}} r_a(F_1, F_2)} \geq \frac{1}{2 \cdot v_{min}} \,. \tag{1.43}$$

Since Eq. (1.43) holds for an arbitrary choice of $\mathcal{Z}$ and a, it follows that Theorem 1.3.1 shows that a randomized algorithm that succeeds with probability at least $9/10$ issues at least $1/(200 \cdot v_{min})$ queries in expectation. Since $\Lambda = 1/(5 \cdot v_{min})$, the lower bound given by Theorem 1.3.1 is at least $\Lambda/40$, which completes the proof. □

1.10 Valid functions have a unique local minimum

In this section, we define an abstract class of functions called "valid functions". Such functions have a unique local minimum and will be used in both the congestion and separation lower bounds.

Definition 1.10.1 (Valid function). *Let $W = (w_1, \ldots, w_s)$ be a walk in the graph G. A function f is* valid *with respect to the walk W if it satisfies the next conditions:*

1. *For all $u, v \in W$, if $\max\{\mathrm{i} \in [s] \mid v = w_\mathrm{i}\} < \max\{\mathrm{i} \in [s] \mid u = w_\mathrm{i}\}$, then $f(v) > f(u)$.*

   In other words, as one walks along the walk W starting from w_1 until w_s, if the last time the vertex v appears is before the last time that vertex u appears, then $f(v) > f(u)$.

2. *For all $v \in V \setminus W$, we have $f(v) = dist(w_1, v) > 0$.*

3. *$f(w_\mathrm{i}) \leq 0$ for all $\mathrm{i} \in [s]$.*

An illustration of a valid function is shown in Figure 1.2.

Next we prove every valid function has a unique local minimum.

Lemma 6. *Suppose $W = (w_1, \ldots, w_s)$ is a walk in G and $f : V \to \mathbb{R}$ is a valid function for the walk W. Then f has a unique local minimum at w_s, the last vertex on the walk.*

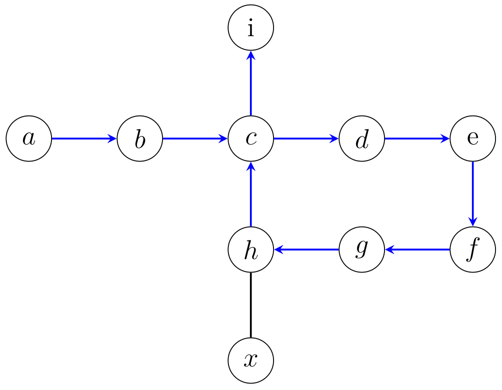


Figure 1.2. Consider the walk $W = (a, b, c, d, \mathrm{e}, f, g, h, c, \mathrm{i})$ in blue, where $f(a), \ldots, f(\mathrm{i}) \le 0$. Observe that $w_5 = \mathrm{e}$ and $w_9 = c$ so $f(\mathrm{e}) > f(c)$. Since the vertex x is *not* in the walk, $f(x) = dist(w_1, x) = dist(a, x) = 4$.

Proof. Let $v \in V$ be an arbitrary vertex. We consider two cases:

Case 1: $v \notin W$**.**

By condition 2 in the definition of a valid function, we have

$$f(v) = dist(w_1, v) > 0\,.$$

Let u be the neighbor of v on a shortest path from v to w_1, breaking ties lexicographically if multiple such neighbors exist. We have two subcases:

(a) If $u \in W$, then $f(u) \le 0 < f(v)$.

(b) If $u \notin W$, then $f(v) = f(u) + 1 > f(u)$.

In both subcases, vertex v has a neighbour with a smaller value, so v is not a local minimum of f.

Case 2: $v \in W$.

Let $\mathrm{i} = \max\{\mathrm{j} \in [s] \mid v = w_\mathrm{j}\}$ be the last index where vertex v appears on the walk W. We have two subcases:

(a) If $\mathrm{i} < s$, let $u = w_{\mathrm{i}+1}$ be the next vertex along the walk. By maximality of the index i, the walk W does not visit vertex v anymore in $(w_{\mathrm{i}+1}, \ldots, w_s)$. Since $v = w_\mathrm{i}$, condition 1 of a valid function implies

$$f(v) = f(w_\mathrm{i}) > f(w_{\mathrm{i}+1}) = f(u),$$

since $\max\{\mathrm{j} \in [s] \mid u = w_\mathrm{j}\} \geq \mathrm{i} + 1 > \mathrm{i} = \max\{\mathrm{j} \in [s] \mid v = w_\mathrm{j}\}$. Thus vertex w_i is not a local minimum.

(b) If $\mathrm{i} = s$, then $v = w_s$. By the analysis in the previous cases (1.a, 1.b, and 2.a), each vertex $u \in [n] \setminus \{w_s\}$ has a neighbor with a strictly smaller value than u, and so u cannot be a local minimum. Since the graph G must have at least one local minimum, at the global minimum, it follows that w_s is the unique local and global minimum of f.

$\square$

1.11 Lower bound for local search via congestion: proofs

In this section we prove the lower bound of $\Omega\left(n^{1.5}/g\right)$, where g is the vertex congestion. We start with the basic definitions. Then we state and prove the lower bound. Afterwards, we show the helper lemmas used in the proof of the theorem.

1.11.1 Basic definitions for congestion

Recall we have a graph $G = ([n], E)$ with vertex congestion g. This means there exists an all-pairs set of paths $\mathcal{P} = \{P^{u,v}\}_{u,v \in [n]}$ with vertex congestion g [14], but no such set of paths exists for $g - 1$. We fix the set $\mathcal{P}$, requiring $P^{u,u} = (u)\ \forall u \in [n]$.

For each $u, v \in [n]$, let $\mathfrak{q}_v(u)$ be the number of paths in $\mathcal{P}$ that start at vertex u and contain v:

$$\mathfrak{q}_v(u) = \left|\{P^{u,w} \in \mathcal{P} : w \in [n], v \in P^{u,w}\}\right| . \tag{1.44}$$

Let $L \in [n]$, with $L \geq 2$, be a parameter that we set later.

Given a sequence of k vertices $\mathbf{x} = (x_1, \ldots, x_k)$, we write $\mathbf{x}_{1 \to \mathrm{j}} = (x_1, \ldots, x_{\mathrm{j}})$ to refer to a prefix of the sequence, for an index $\mathrm{j} \in [k]$.

Given a walk $Q = (v_1, \ldots, v_k)$ in G, let Q_{i} refer to the i-th vertex in the walk (i.e. $Q_{\mathrm{i}} = v_{\mathrm{i}}$). For each vertex $u \in [n]$, let $\mu(Q, u)$ be the number of times that vertex u appears in Q.

Definition 1.11.1 (Staircase). *Given a sequence $\mathbf{x} = (x_1, \ldots, x_k)$ of vertices in G, a* staircase *induced by $\mathbf{x}$ is a walk $S_{\mathbf{x}} = S_{\mathbf{x},1} \circ \ldots \circ S_{\mathbf{x},k-1}$, where each $S_{\mathbf{x},\mathrm{i}}$ is a path in G starting at vertex x_{i} and ending at $x_{\mathrm{i}+1}$. Each vertex x_{i} is called a* milestone *and each path $S_{\mathbf{x},\mathrm{i}}$ a* quasi-segment.

The staircase $S_{\mathbf{x}}$ is said to be induced by $\mathbf{x}$ and $\mathcal{P} = \{P^{u,v}\}_{u,v \in [n]}$ *if additionally we have $S_{\mathbf{x},\mathrm{i}} = P^{x_{\mathrm{i}}, x_{\mathrm{i}+1}}$ for all* $\mathrm{i} \in [k-1]$.

Definition 1.11.2 (Tail of a staircase). *Let $S_{\mathbf{x}} = S_{\mathbf{x},1} \circ \ldots \circ S_{\mathbf{x},k-1}$ be a staircase induced by some sequence $\mathbf{x} = (x_1, \ldots, x_k) \in [n]^k$. For each $\mathrm{j} \in [k-1]$, let $T = S_{\mathbf{x},\mathrm{j}} \circ \ldots \circ S_{\mathbf{x},k-1}$. Then $Tail(\mathrm{j}, S_{\mathbf{x}})$ is obtained from T by removing the first occurrence of x_{j} in T (and only the first occurrence). We also define $Tail(k, S_{\mathbf{x}})$ to be the empty sequence.*

Next we define the set of functions, $\mathcal{X}$, that will be used when invoking Theorem 1.3.1.

[14]↑That is, each vertex is used at most g times across all the paths.

Definition 1.11.3 (The functions $f_{\mathbf{x}}$ and $g_{\mathbf{x},b}$; the set $\mathcal{X}$)**.** *Suppose $\mathcal{P} = \{P^{u,v}\}_{u,v \in [n]}$ is an all-pairs set of paths in G. For each sequence of vertices $\mathbf{x} \in \{1\} \times [n]^L$, define a function $f_{\mathbf{x}} : [n] \to \{-n^2 - n, \ldots, 0, \ldots, n\}$ such that for each $v \in [n]$:*

- *If $v \notin S_{\mathbf{x}}$, then set $f_{\mathbf{x}}(v) = dist(v, 1)$, where $S_{\mathbf{x}}$ is the staircase induced by $\mathbf{x}$ and $\mathcal{P}$.*
- *If $v \in S_{\mathbf{x}}$, then set $f_{\mathbf{x}}(v) = -\mathrm{i} \cdot n - \mathrm{j}$, where i is the maximum index with $v \in P^{x_{\mathrm{i}}, x_{\mathrm{i}+1}}$, and v is the j-th vertex in $P^{x_{\mathrm{i}}, x_{\mathrm{i}+1}}$.*

Also, for each $\mathbf{x} \in \{1\} \times [n]^L$ and $b \in \{0, 1\}$, let $g_{\mathbf{x},b} : [n] \to \{-n^2 - n, \ldots, 0, \ldots, n\} \times \{-1, 0, 1\}$ be such that, for all $v \in [n]$:

$$g_{\mathbf{x},b}(v) = \begin{cases} \Big(f_{\mathbf{x}}(v), b\Big) & \text{if } v = x_{L+1} \\ \Big(f_{\mathbf{x}}(v), -1\Big) & \text{if } v \neq x_{L+1} \end{cases}. \tag{1.45}$$

Let $\mathcal{X} = \Big\{ g_{\mathbf{x},b} \mid \mathbf{x} \in \{1\} \times [n]^L$ and $b \in \{0, 1\} \Big\}$.

We will show later that for each sequence $\mathbf{x}$, the function $f_{\mathbf{x}}$ has a unique local minimum at the end of the staircase $S_{\mathbf{x}}$.

An example of a staircase $S_{\mathbf{x}}$ for some sequence of vertices $\mathbf{x}$ is shown in the next figure, together with the accompanying value function $f_{\mathbf{x}}$.

(46) Let G be the grid graph on $n = 16$ nodes from Figure 1.3. Consider the sequence of vertices

$$\mathbf{x} = (x_1, x_2, x_3, x_4) = (v_1, v_6, v_{11}, v_{16}),$$

We fix an all-pairs set of paths $\mathcal{P} = \{P^{u,v}\}_{u,v \in [n]}$, such that

- $P^{v_1,v_6} = (v_1, v_2, v_3, v_7, v_6)$; $P^{v_6,v_{11}} = (v_6, v_{10}, v_{11})$; $P^{v_{11},v_{16}} = (v_{11}, v_7, v_8, v_{12}, v_{16})$, where
  $P^{v_1,v_6}, P^{v_6,v_{11}}, P^{v_{11},v_{16}} \in \mathcal{P}$.

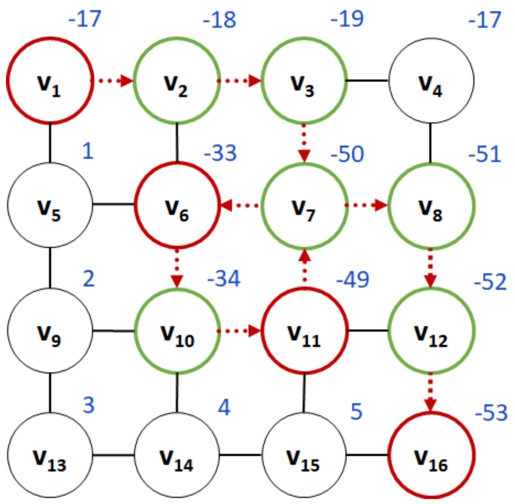


Figure 1.3. Example of a staircase with the accompanying value function. The sequence of milestones is $(v_1, v_6, v_{11}, v_{16})$, which are shown in red. The vertices of the staircase are shown in red and green vertices, connected by red dotted edges. For each node v, the value of the function at v is shown in blue.

- For each other pair of vertices (u, w), we set $P^{u,w}$ as the shortest path between u and w, breaking ties lexicographically (vertices with lower index come first).

Then the staircase induced by $\mathbf{x}$ and $\mathcal{P}$ is

$$S_{\mathbf{x}} = P^{v_1,v_6} \circ P^{v_6,v_{11}} \circ P^{v_{11},v_{16}} = (v_1, v_2, v_3, v_7, v_6, v_{10}, v_{11}, v_7, v_8, v_{12}, v_{16}) .$$

For example, $f(v_4) = dist(v_1, v_4) = 3$ since $v_4 \notin S_{\mathbf{x}}$, and $f_{\mathbf{x}}(v_7) = -3n - 2 = -50$ since v_7 is the second node in $P^{x_3,x_4} = P^{v_{11},v_{16}}$ (even though v_7 is also included in the path P^{x_1,x_2}).

Definition 1.11.4 (The map $\mathcal{H}$)**.** *Suppose $\mathcal{P} = \{P^{u,v}\}_{u,v\in[n]}$ is an all-pairs set of paths in G and $\mathcal{X}$ is the set of functions $g_{\mathbf{x},b}$ from Definition 1.11.3. Define $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ as*

$$\mathcal{H}(g_{\mathbf{x},b}) = b \qquad \forall \mathbf{x} \in \{1\} \times [n]^L \textit{ and } b \in \{0, 1\} .$$

To define a suitable r function, we only concern ourselves with pairs of staircases that do not have repeated milestones within themselves and with different hidden bits. For such a pair of staircases, we assign a relative difficulty of distinguishing them that scales with the length of their common prefix: the longer their common prefix, the more function values they agree on, and so we assign a higher r value.

Definition 1.11.5 (Good/bad sequences of vertices; Good/bad functions)**.** *A sequence of k vertices $\mathbf{x} = (x_1, \ldots, x_k)$ is* good *if $x_{\mathrm{i}} \neq x_{\mathrm{j}}$ for all i, j with $1 \leq \mathrm{i} < \mathrm{j} \leq k$; otherwise, $\mathbf{x}$ is* bad*. Moreover, for each $b \in \{0, 1\}$ a function $F = g_{\mathbf{x},b} \in \mathcal{X}$ is good if $\mathbf{x}$ is good, and* bad *otherwise.*

Definition 1.11.6 (The function r)**.** *Let $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a symmetric function defined as follows. For each $\mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L$ and $b_1, b_2 \in \{0, 1\}$, we have*

$$r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = \begin{cases} 0 & \textit{if at least one of the following holds: } b_1 = b_2 \textit{ or } \mathbf{x} \textit{ is bad or } \mathbf{y} \textit{ is bad.} \\ n^{\mathrm{j}} & \textit{otherwise, where } \mathrm{j} \textit{ is the maximum index for which } \mathbf{x}_{1\to \mathrm{j}} = \mathbf{y}_{1\to \mathrm{j}}\,. \end{cases}$$

(47) Suppose $L = 3$. Consider the graph $G = ([n], E)$ and consider the sequences of vertices $\mathbf{x} = (1, 2, 3, 4)$, $\mathbf{y} = (1, 3, 5, 3)$, and $\mathbf{z} = (1, 2, 5, 4)$. Since $\mathbf{y}$ has repeated elements while $\mathbf{x}$ and $\mathbf{z}$ do not, we see that $\mathbf{x}$ is good, $\mathbf{y}$ is bad, and $\mathbf{z}$ is good. Then, for each $b \in \{0, 1\}$, we have

- $r(\cdot, g_{\mathbf{y},b}) = r(g_{\mathbf{y},b}, \cdot) = 0$ since $\mathbf{y}$ is bad.
- $r(g_{\mathbf{x},b}, g_{\mathbf{z},1-b}) = r(g_{\mathbf{z},1-b}, g_{\mathbf{x},b}) = n^2$ since $\mathbf{x}$ is good, $\mathbf{z}$ is good, $\mathbf{x}_{1\to 2} = \mathbf{z}_{1\to 2}$, and $x_3 \neq z_3$.

The function r will be used directly to invoke Theorem 1.3.1, but we also define here some related helper functions to use r in conjunction with certain indicator variables.

Definition 1.11.7 (The function r_v)**.** *For each $v \in [n]$, define $r_v : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ as follows:*

$$r_v(F_1, F_2) = r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} \qquad \forall F_1, F_2 \in \mathcal{X}\,.$$

Definition 1.11.8 (The function $\widetilde{r}_v$)**.** *For each $v \in [n]$, define $\widetilde{r}_v : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ as follows:*

$$\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot \mathbb{1}_{\{\mu(S_{\mathbf{x}},v) \leq \mu(S_{\mathbf{y}},v)\}} \qquad \forall \mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L \;\; \forall b_1, b_2 \in \{0,1\}\,.$$

Observation 1.11.1. *For each $\mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L$ and $b \in \{0,1\}$, we have*

$$\widetilde{r}_v(g_{\mathbf{x},b}, g_{\mathbf{y},b}) = r_v(g_{\mathbf{x},b}, g_{\mathbf{y},b}) = r(g_{\mathbf{x},b}, g_{\mathbf{y},b}) = 0\,. \tag{1.48}$$

Proof. By definition of r, we have $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = 0$ when $b_1 = b_2$. Then for all $b \in \{0,1\}$:

$$\begin{aligned}\widetilde{r}_v(g_{\mathbf{x},b}, g_{\mathbf{y},b}) &= r_v(g_{\mathbf{x},b}, g_{\mathbf{y},b}) \cdot \mathbb{1}_{\{\mu(S_{\mathbf{x}},v) \leq \mu(S_{\mathbf{y}},v)\}} \\ &= r(g_{\mathbf{x},b}, g_{\mathbf{y},b}) \cdot \mathbb{1}_{\{g_{\mathbf{x},b}(v) = g_{\mathbf{y},b}(v)\}} \cdot \mathbb{1}_{\{\mu(S_{\mathbf{x}},v) \leq \mu(S_{\mathbf{y}},v)\}} = 0\,.\end{aligned} \tag{1.49}$$

□

1.11.2 Proof of the congestion lower bound

In this section we include the proof of Theorem 1.3.2. The proofs of lemmas used in the theorem are included afterwards, in Section 1.11.3.

Theorem 1.3.2. *Let $G = (V, E)$ be a connected undirected graph with n vertices. Then the randomized query complexity of local search on G is $\Omega\left(\frac{n^{1.5}}{g}\right)$, where g is the vertex congestion of the graph.*

Proof. Consider the following setting of parameters:

(a) $L = \lfloor \sqrt{n} \rfloor - 1$.

(b) Fix an all-pairs set of paths $\mathcal{P} = \{P^{u,v}\}_{u,v \in [n]}$ for G, such that $\mathcal{P}$ has vertex congestion g.

(c) The finite set A is the set of vertices $[n]$.

(d) The finite set B is $\{-n^2 - n, \dots, 0, \dots, n\} \times \{-1, 0, 1\}$.

(e) The functions $f_{\mathbf{x}}$, $g_{\mathbf{x},b}$, and the set $\mathcal{X}$ given by Definition 1.11.3. *(Recall $g_{\mathbf{x},b}(v) = (f_{\mathbf{x}}, c)$ for all $v \in [n]$, where $c = -1$ if $v \neq x_{L+1}$, and $c = b$ if $v = x_{L+1}$, i.e. $c = b$ if and only if v is a local minimum of $f_{\mathbf{x}}$. Also, $\mathcal{X} = \{g_{\mathbf{x},b} \mid \mathbf{x} \in \{1\} \times [n]^L$ and $b \in \{0,1\}\}$.)*

(f) Map $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ as in Definition 1.11.4. *(Recall $\mathcal{H}(g_{\mathbf{x},b}) = b$ for all $\mathbf{x} \in \{1\} \times [n]^L$ and $b \in \{0, 1\}$.)*

(g) The function r as in Definition 1.11.6.

By Lemma 7, each function $f_{\mathbf{x}}$ is valid for all $\mathbf{x} \in \{1\} \times [n]^L$, so Lemma 6 implies that each function $f_{\mathbf{x}}$ has a unique local minimum (at x_{L+1}). Therefore by Lemma 2 invoked with $f = f_{\mathbf{x}}$ and $h_b = g_{\mathbf{x},b}$, it suffices to show a lower bound for the corresponding decision problem: return the hidden bit $b \in \{0, 1\}$ given oracle access to the function $g_{\mathbf{x},b}$.

For each $\mathcal{Z} \subseteq \mathcal{X}$, let

$$M(\mathcal{Z}) = \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2)\,. \tag{1.50}$$

By Lemma 8, there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$. Thus the conditions required by Theorem 1.3.1 are met. By invoking Theorem 1.3.1 with the parameters in (a-g), we get that the randomized query complexity of the decision problem, and thus also of local search on G, is

$$\Omega\left(\min_{\mathcal{Z} \subseteq \mathcal{X} : q(\mathcal{Z}) > 0} \frac{M(\mathcal{Z})}{q(\mathcal{Z})}\right), \text{ where } q(\mathcal{Z}) = \max_{v \in [n]} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}}\,.$$

To get an explicit lower bound in terms of congestion, we will upper bound $q(\mathcal{Z})$ and lower bound $M(\mathcal{Z})$ for subsets $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$.

Fix an arbitrary subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$. Since $r(F_1, F_2) = 0$ when F_1 or F_2 is bad, it suffices to consider subsets $\mathcal{Z} \subseteq \mathcal{X}$ where each function $F \in \mathcal{Z}$ is good.

Upper bounding $q(\mathcal{Z})$.

Let $v \in [n]$ be arbitrary.

Fix an arbitrary function $F_1 \in \mathcal{Z}$. Since F_1 is good, there exist $\mathbf{x} \in \{1\} \times [n]^L$ and $b_1 \in \{0,1\}$ such that $F_1 = g_{\mathbf{x},b_1}$ and $\mathbf{x}$ is good. Since $\mathcal{Z} \subseteq \mathcal{X}$ and $\widetilde{r}_v \geq 0$, we have

$$\sum_{F_2 \in \mathcal{Z}} \widetilde{r}_v(F_1, F_2) \leq \sum_{F_2 \in \mathcal{X}} \widetilde{r}_v(F_1, F_2)\,. \tag{1.51}$$

Using the definition of $\mathcal{X} = \{g_{\mathbf{y},b_2} \mid \mathbf{y} \in \{1\} \times [n]^L,\ b_2 \in \{0,1\}\}$, the fact that $F_1 = g_{\mathbf{x},b_1}$, and partitioning the space of functions $F_2 \in \mathcal{X}$ by the length of the prefix that the staircase corresponding to F_2 shares with the staircase corresponding to F_1, we can upper bound the right hand side of Eq. (1.51):

$$\sum_{F_2 \in \mathcal{X}} \widetilde{r}_v(F_1, F_2) = \sum_{\mathbf{y} \in \{1\} \times [n]^L,\ b_2 \in \{0,1\}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \leq \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L,\ b_2 \in \{0,1\} \\ \mathrm{j} = \max\{\mathrm{i}\,:\, \mathbf{x}_{1\to \mathrm{i}} = \mathbf{y}_{1\to \mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2})\,. \tag{1.52}$$

Combining Eq. (1.51) and Eq. (1.52), we get

$$\sum_{F_2 \in \mathcal{Z}} \widetilde{r}_v(F_1, F_2) \leq \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L,\ b_2 \in \{0,1\} \\ \mathrm{j} = \max\{\mathrm{i}\,:\, \mathbf{x}_{1\to \mathrm{i}} = \mathbf{y}_{1\to \mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2})\,. \tag{1.53}$$

For each $\mathrm{j} \in [L+1]$, let

$$T_{\mathrm{j}} = \left\{ \mathbf{y} \in \{1\} \times [n]^L \mid \max\{\mathrm{i} : \mathbf{x}_{1\to \mathrm{i}} = \mathbf{y}_{1\to \mathrm{i}}\} = \mathrm{j} \text{ and } v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}}) \right\}.$$

Then for each $\mathrm{j} \in [L]$, we can bound the part of the sum in Eq. (1.53) corresponding to index j via the next chain of inequalities:

$$\begin{aligned}
\sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L,\, b_2 \in \{0,1\}: \\ \mathrm{j} = \max\{\mathrm{i} \,:\, \mathbf{x}_{1 \to \mathrm{i}} = \mathbf{y}_{1 \to \mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) &\leq \sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L: \\ \mathrm{j} = \max\{\mathrm{i} \,:\, \mathbf{x}_{1 \to \mathrm{i}} = \mathbf{y}_{1 \to \mathrm{i}}\} \\ v \in Tail(\mathrm{j}, S_{\mathbf{y}})}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) && \text{(By Lemma 11)} \\
&\leq n^{\mathrm{j}} \cdot |T_{\mathrm{j}}| && (\text{Since } r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \leq n^{\mathrm{j}} \text{ when } \mathrm{j} = \max\{\mathrm{i} : \mathbf{x}_{1 \to \mathrm{i}} = \mathbf{y}_{1 \to \mathrm{i}}\}) \\
&\leq n^{\mathrm{j}} \cdot \left(\mathfrak{q}_v(x_{\mathrm{j}}) \cdot n^{L-\mathrm{j}} + L \cdot g \cdot n^{L-\mathrm{j}-1} \right), && \text{(By Lemma 12.)}
\end{aligned}$$

where we recall that $\mathfrak{q}_v(u)$ is the number of paths in $\mathcal{P}$ that start at vertex u and contain v.

Using the identity $n^{\mathrm{j}} \cdot \left(\mathfrak{q}_v(x_{\mathrm{j}}) \cdot n^{L-\mathrm{j}} + L \cdot g \cdot n^{L-\mathrm{j}-1} \right) = n^L \cdot \left(\mathfrak{q}_v(x_{\mathrm{j}}) + \frac{L \cdot g}{n} \right)$, we obtain

$$\sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L,\, b_2 \in \{0,1\}: \\ \mathrm{j} = \max\{\mathrm{i} \,:\, \mathbf{x}_{1 \to \mathrm{i}} = \mathbf{y}_{1 \to \mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \leq n^L \cdot \left(\mathfrak{q}_v(x_{\mathrm{j}}) + \frac{L \cdot g}{n} \right). \tag{1.54}$$

When $\mathrm{j} = L + 1$, since $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$ implies $b_2 = 1 - b_1$ (see Observation 1.11.1), we have

$$\sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L,\, b_2 \in \{0,1\}: \\ L+1 = \max\{\mathrm{i} : \mathbf{x}_{1 \to \mathrm{i}} = \mathbf{y}_{1 \to \mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \leq n^{L+1}. \tag{1.55}$$

Summing Eq. (1.54) for all $\mathrm{j} \in [L]$ and adding it to Eq. (1.55) for $\mathrm{j} = L + 1$, we can now upper bound the right hand side of Eq. (1.53) as follows:

$$\begin{aligned}
\sum_{F_2 \in \mathcal{Z}} \widetilde{r}_v(F_1, F_2) &\leq \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L, b_2 \in \{0,1\}: \\ \max\{\mathrm{i} : \mathbf{x}_{1 \to \mathrm{i}} = \mathbf{y}_{1 \to \mathrm{i}}\} = \mathrm{j}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) && \text{(By Eq. (1.53))} \\
&\leq n^L \cdot \sum_{\mathrm{j}=1}^{L} \left(\mathfrak{q}_v(x_{\mathrm{j}}) + \frac{L \cdot g}{n} \right) + n^{L+1} && \text{(By Eq. (1.54) and Eq. (1.55))} \\
&\leq n^L \left(\sum_{u \in [n]} \mathfrak{q}_v(u) \right) + n^L \left(\sum_{\mathrm{j}=1}^{L} \frac{gL}{n} \right) + n^{L+1} \\
&\qquad (\text{Since } \mathbf{x} \text{ is good, i.e. } \mathbf{x} \text{ has no repeated vertices})
\end{aligned}$$

$$\leq n^L \cdot g + n^L \cdot \frac{g \cdot L^2}{n} + n^{L+1}$$

(Since $\sum_{u\in[n]} \mathfrak{q}_v(u)$ equals the number of paths in $\mathcal{P}$ that contain v, which is the congestion at v.)

$$\leq 3 \cdot g \cdot n^L . \qquad \text{(Since } L \leq \sqrt{n}-1 \text{ and } g \geq n\text{)}$$

Thus, for each good function $F_1 \in \mathcal{Z}$, we have

$$\sum_{F_2\in\mathcal{Z}} \widetilde{r}_v(F_1, F_2) \leq 3 \cdot g \cdot n^L . \tag{1.56}$$

Summing Eq. (1.56) over all $F_1 \in \mathcal{Z}$ (each of which is good, since $\mathcal{Z}$ was chosen to have good functions only), and invoking Lemma 9 yields

$$\begin{aligned}\sum_{F_1,F_2\in\mathcal{Z}} r_v(F_1, F_2) &\leq 2 \cdot \sum_{F_1,F_2\in\mathcal{Z}} \widetilde{r}_v(F_1, F_2) && \text{(By Lemma 9)}\\ &\leq 2 \cdot |\mathcal{Z}| \cdot 3 \cdot g \cdot n^L && \text{(By Eq. (1.56))}\\ &= |\mathcal{Z}| \cdot 6g \cdot n^L . && (1.57)\end{aligned}$$

Since we had considered an arbitrary vertex $v \in [n]$, taking the maximum over all $v \in [n]$ in Eq. (1.57) yields

$$q(\mathcal{Z}) = \max_{v\in[n]} \sum_{F_1\in\mathcal{Z}} \sum_{F_2\in\mathcal{Z}} r_v(F_1, F_2) \leq |\mathcal{Z}| \cdot 6g \cdot n^L . \tag{1.58}$$

Lower bounding $M(\mathcal{Z})$.

Since each function $F_1 \in \mathcal{Z}$ is good by choice of $\mathcal{Z}$, Lemma 14 yields

$$\sum_{F_2\in\mathcal{X}} r(F_1, F_2) \geq \frac{1}{2\mathrm{e}} \cdot (L+1) \cdot n^{L+1} \qquad \forall F_1 \in \mathcal{Z} . \tag{1.59}$$

Using Eq. (1.59) and recalling the definition of $M(\mathcal{Z})$ from Eq. (1.50), we get

$$M(\mathcal{Z}) = \sum_{F_1\in\mathcal{Z}} \sum_{F_2\in\mathcal{X}} r(F_1, F_2) \geq \frac{|\mathcal{Z}|}{2\mathrm{e}} \cdot (L+1) \cdot n^{L+1} . \tag{1.60}$$

Combining the bounds.

Combining the bounds from Eq. (1.58) and Eq. (1.60), we can now estimate the bound from Theorem 1.3.1:

$$\begin{aligned}\min_{\substack{\mathcal{Z}\subseteq\mathcal{X}:\\ q(\mathcal{Z})>0}} \frac{M(\mathcal{Z})}{q(\mathcal{Z})} &\geq \frac{\frac{|\mathcal{Z}|}{2\mathrm{e}}\cdot(L+1)n^{L+1}}{|\mathcal{Z}|\cdot 6g\cdot n^L} && \text{(By Eq. (1.58) and Eq. (1.60))}\\ &\geq \frac{n^{1.5}}{24\mathrm{e}\cdot g}\,. && \text{(Since } L+1=\lfloor\sqrt{n}\rfloor\geq\sqrt{n}/2\text{)}\end{aligned}$$

Therefore, the randomized query complexity of local search is

$$\Omega\left(\min_{\substack{\mathcal{Z}\subseteq\mathcal{X}:\\ q(\mathcal{Z})>0}} \frac{M(\mathcal{Z})}{q(\mathcal{Z})}\right)\subseteq\Omega\left(\frac{n^{1.5}}{g}\right)\,.$$

This completes the proof of the theorem. □

1.11.3 Helper lemmas

In this section we prove the helper lemmas that are used in the proof of Theorem 1.3.2. All the lemmas assume the setup of the parameters from Theorem 1.3.2.

Lemma 7. *For each $\mathbf{x}\in\{1\}\times[n]^L$, the function $f_{\mathbf{x}}$ is valid for the staircase $S_{\mathbf{x}}$ induced by $\mathbf{x}$ and $\mathcal{P}$, where $\mathcal{P}=\{P^{u,v}\}_{u,v\in[n]}$ is the all-pairs set of paths in G.*

Proof. Let $S_{\mathbf{x}}=(w_1,\dots,w_s)$ be the vertices of the staircase $S_{\mathbf{x}}$ induced by $\mathbf{x}$ and $\mathcal{P}$. We show that all the three conditions required by the definition of a valid function (Definition 2.4.6) hold.

To show the first condition of validity, consider two vertices $v_1,v_2\in S_{\mathbf{x}}$. Define

$$\mathrm{i}_1=\max\{k\in[L]\mid v_1\in P^{x_k,x_{k+1}}\}\,.$$

Define i_2 similarly for v_2. Let j_1 and j_2 be the indices of v_1 and v_2 in $P^{x_{\mathrm{i}_1},x_{\mathrm{i}_1+1}}$ and $P^{x_{\mathrm{i}_2},x_{\mathrm{i}_2+1}}$ respectively. Note that $\mathrm{j}_1, \mathrm{j}_2 \in [n]$. By definition of the function $f_{\mathbf{x}}$ (Definition 1.11.3), we have

$$f_{\mathbf{x}}(v_1) = -n \cdot \mathrm{i}_1 - \mathrm{j}_1 \quad \text{and} \quad f_{\mathbf{x}}(v_2) = -n \cdot \mathrm{i}_2 - \mathrm{j}_2 \,. \tag{1.61}$$

Without loss of generality, the last time vertex v_1 appears on the path $S_{\mathbf{x}}$, starting from w_1 towards w_s, is earlier than the last time vertex v_2 appears, that is:

$$\max\{k \in [s] \mid v_1 = w_k\} < \max\{k \in [s] \mid v_2 = w_k\} \,.$$

Then $\mathrm{i}_1 \leq \mathrm{i}_2$. We consider two cases:

- If $\mathrm{i}_1 = \mathrm{i}_2$, then

$$\begin{aligned} f_{\mathbf{x}}(v_1) - f_{\mathbf{x}}(v_2) &= \mathrm{j}_2 - \mathrm{j}_1 && \text{(By Eq. (1.61))} \\ &> 0 && \text{(Since by assumption } v_1 \text{ last appears on } S_{\mathbf{x}} \text{ before } v_2.) \end{aligned}$$

- If $\mathrm{i}_1 < \mathrm{i}_2$, then

$$\begin{aligned} f_{\mathbf{x}}(v_1) - f_{\mathbf{x}}(v_2) &\geq n + (\mathrm{j}_2 - \mathrm{j}_1) && \text{(By Eq. (1.61))} \\ &> 0 && \text{(Since } \mathrm{j}_1, \mathrm{j}_2 \in [n].) \end{aligned}$$

Therefore the first condition of validity is satisfied.

Also by Definition 1.11.3, of the function $f_{\mathbf{x}}$, we have that:

- $f_{\mathbf{x}}(v) = dist(1, v)$ for all $v \notin S_{\mathbf{x}}$, so the second condition of validity is satisfied.
- $f_{\mathbf{x}}(v) \leq 0$ for all $v \in S_{\mathbf{x}}$, so the third condition of validity is satisfied.

Therefore $f_{\mathbf{x}}$ is valid for the staircase $S_{\mathbf{x}}$ induced by $\mathbf{x}$ and $\mathcal{P}$. □

Lemma 8. *The next two properties hold:*

- *Let $F_1, F_2 \in \mathcal{X}$. Then $r(F_1, F_2) = 0$ when $\mathcal{H}(F_1) = \mathcal{H}(F_2)$.*
- *There exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ such that*

$$q(\mathcal{Z}) = \max_{v\in[n]} \sum_{F_1\in\mathcal{Z}} \sum_{F_2\in\mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v)\neq F_2(v)\}} > 0 \,.$$

Proof. We first show that $r(F_1, F_2) = 0$ when $\mathcal{H}(F_1) = \mathcal{H}(F_2)$. To see this, suppose $\mathcal{H}(F_1) = \mathcal{H}(F_2)$ for some functions $F_1, F_2 \in \mathcal{X}$. Then by definition of the set of functions $\mathcal{X}$, there exist sequences of vertices $\mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L$ and bits $b_1, b_2 \in \{0, 1\}$ such that $F_1 = g_{\mathbf{x},b_1}$ and $F_2 = g_{\mathbf{y},b_2}$. By definition of $\mathcal{H}$, we have $\mathcal{H}(g_{\mathbf{x},b_1}) = b_1$ and $\mathcal{H}(g_{\mathbf{y},b_2}) = b_2$. Since $\mathcal{H}(F_1) = \mathcal{H}(F_2)$, we have $b_1 = b_2$. Then $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = 0$ by definition of r, or equivalently, $r(F_1, F_2) = 0$.

Next we show there is a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$. To see this, consider two disjoint sets of vertices $U_1, U_2 \subset [n]$ such that $U_1 = \{u^1_2, \ldots, u^1_{L+1}\}$, $U_2 = \{u^2_2, \ldots, u^2_{L+1}\}$, each vertex $u^{\mathrm{i}}_{\mathrm{j}}$ appears exactly once in U_{i}, and $u^{\mathrm{i}}_{\mathrm{j}} \neq 1$ for all i, j. Such sets U_1, U_2 exist since

$$|U_1| + |U_2| + |\{1\}| = 2L + 1 = 2(\lfloor\sqrt{n}\rfloor - 1) + 1 \leq 2\sqrt{n} \leq n \quad \text{for } n \geq 4\,.$$

Form the sequences of vertices $W^1 = (1, u^1_2, \ldots, u^1_{L+1})$ and $W^2 = (1, u^2_2, \ldots, u^2_{L+1})$. Then both W^1 and W^2 are good. Consider now the functions $g_{W^1,0}$ and $g_{W^2,1}$. By definition of r, we have $r(g_{W^1,0}, g_{W^2,1}) = n$, since the maximum index j for which $W^1_{1\to\mathrm{j}} = W^2_{1\to\mathrm{j}}$ is $\mathrm{j} = 1$. Then

$$q(\{g_{W^1,0}, g_{W^2,1}\}) \geq r(g_{W^1,0}, g_{W^2,1}) \cdot \mathbb{1}_{\{g_{W^1,0}(u^1_{L+1})\neq g_{W^2,1}(u^1_{L+1})\}} = n > 0\,.$$

Thus there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$ as required. □

Lemma 9. *For each $v \in [n]$ and subset $\mathcal{Z} \subseteq \mathcal{X}$, we have*

$$\sum_{F_1,F_2\in\mathcal{Z}} r_v(F_1, F_2) \leq 2 \sum_{F_1,F_2\in\mathcal{Z}} \widetilde{r}_v(F_1, F_2)\,.$$

Proof. By Definition 1.11.7, we have $r_v(F_1, F_2) = r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}}$ for all $v \in [n]$ and $F_1, F_2 \in \mathcal{X}$. Then r_v is symmetric since both the function r and the indicator $\mathbb{1}_{\{F_1(v) \neq F_2(v)\}}$ are symmetric. Also recalling that for a walk Q, the number of times that a vertex u appears in Q is denoted $\mu(Q, u)$, we have:

$$\begin{aligned}
\sum_{F_1,F_2 \in \mathcal{Z}} r_v(F_1, F_2) &= \sum_{\substack{F_1,F_2 \in \mathcal{Z}: \\ F_1 = g_{\mathbf{x},b_1}; F_2 = g_{\mathbf{y},b_2} \\ \mu(S_{\mathbf{x}},v) \le \mu(S_{\mathbf{y}},v)}} r_v(F_1, F_2) + \sum_{\substack{F_1,F_2 \in \mathcal{Z}: \\ F_1 = g_{\mathbf{x},b_1}; F_2 = g_{\mathbf{y},b_2} \\ \mu(S_{\mathbf{y}},v) < \mu(S_{\mathbf{x}},v)}} r_v(F_1, F_2) \\
&= \sum_{\substack{F_1,F_2 \in \mathcal{Z}: \\ F_1 = g_{\mathbf{x},b_1}; F_2 = g_{\mathbf{y},b_2} \\ \mu(S_{\mathbf{x}},v) \le \mu(S_{\mathbf{y}},v)}} r_v(F_1, F_2) + \left(\sum_{\substack{F_1,F_2 \in \mathcal{Z}: \\ F_1 = g_{\mathbf{x},b_1}; F_2 = g_{\mathbf{y},b_2} \\ \mu(S_{\mathbf{y}},v) \le \mu(S_{\mathbf{x}},v)}} r_v(F_2, F_1) - \sum_{\substack{F_1,F_2 \in \mathcal{Z}: \\ F_1 = g_{\mathbf{x},b_1}; F_2 = g_{\mathbf{y},b_2} \\ \mu(S_{\mathbf{x}},v) = \mu(S_{\mathbf{y}},v)}} r_v(F_2, F_1) \right) \\
&= 2 \sum_{\substack{F_1,F_2 \in \mathcal{Z}: \\ F_1 = g_{\mathbf{x},b_1}; F_2 = g_{\mathbf{y},b_2} \\ \mu(S_{\mathbf{x}},v) \le \mu(S_{\mathbf{y}},v)}} r_v(F_1, F_2) - \sum_{\substack{F_1,F_2 \in \mathcal{Z}: \\ F_1 = g_{\mathbf{x},b_1}; F_2 = g_{\mathbf{y},b_2} \\ \mu(S_{\mathbf{y}},v) = \mu(S_{\mathbf{x}},v)}} r_v(F_2, F_1)\,. \qquad (1.62)
\end{aligned}$$

Recall from Definition 1.11.8, of the function $\widetilde{r}_v$, that

$$\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_1}) = r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot \mathbb{1}_{\{\mu(S_{\mathbf{x}},v) \le \mu(S_{\mathbf{y}},v)\}} \quad \forall \mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L, \forall b_1, b_2 \in \{0, 1\}.$$

Then $\widetilde{r}_v(F_1, F_2) = 0$ when $\mu(S_{\mathbf{x}}, v) > \mu(S_{\mathbf{y}}, v)$, which substituted in Eq. (1.62) gives

$$\begin{aligned}
\sum_{F_1,F_2 \in \mathcal{Z}} r_v(F_1, F_2) &= 2 \sum_{F_1,F_2 \in \mathcal{Z}} \widetilde{r}_v(F_1, F_2) - \sum_{\substack{F_1,F_2 \in \mathcal{Z}: \\ F_1 = g_{\mathbf{x},b_1}; F_2 = g_{\mathbf{y},b_2} \\ \mu(S_{\mathbf{y}},v) = \mu(S_{\mathbf{x}},v)}} r_v(F_2, F_1) \qquad (1.63) \\
&\le 2 \sum_{F_1,F_2 \in \mathcal{Z}} \widetilde{r}_v(F_1, F_2)\,. \qquad (\text{Since } r_v(F_2, F_1) \ge 0 \ \forall F_1, F_2 \in \mathcal{X}, v \in [n])
\end{aligned}$$

This completes the proof of the lemma. □

Lemma 10. *Let* $\mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L$, $b_1, b_2 \in \{0,1\}$, $v \in [n]$. *Let* $\mathrm{j} \in [L+1]$ *be the maximum index for which* $\mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}$. *If* $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$, *then at least one of the next two properties holds:*

(i) $v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$.

(ii) $\mathbf{x} = \mathbf{y}$.

Proof. We start with a few observations.

Recall from Definition 1.11.8 that $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_1}) = r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot \mathbb{1}_{\{\mu(S_{\mathbf{x}},v) \le \mu(S_{\mathbf{y}},v)\}}$. By the lemma statement, we have $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$, and so both of the next inequalities hold:

$$r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0 \tag{1.64}$$

$$\mu(S_{\mathbf{x}}, v) \le \mu(S_{\mathbf{y}}, v)\,. \tag{1.65}$$

By definition of r_v, we have $r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot \mathbb{1}_{\{g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)\}}$. Then Eq. (1.64) implies

$$g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)\,. \tag{1.66}$$

To prove that $v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$ or $\mathbf{x} = \mathbf{y}$ we consider two cases:

Case 1: $v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{x}})$**.**

Let us decompose the staircase $S_{\mathbf{x}}$ into the initial segment $S_{\mathbf{x}_{1\to\mathrm{j}}}$ and the remainder $\mathit{Tail}(\mathrm{j}, S_{\mathbf{x}})$. Similarly, we decompose the staircase $S_{\mathbf{y}}$ into initial segment $S_{\mathbf{y}_{1\to\mathrm{j}}}$ and the remainder $\mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$. We get:

$$\begin{aligned} & \mu(S_{\mathbf{x}}, v) \le \mu(S_{\mathbf{y}}, v) && \text{(By Eq. (1.65).)} \\ \iff & \mu(S_{\mathbf{x}_{1\to\mathrm{j}}}, v) + \mu(\mathit{Tail}(\mathrm{j}, S_{\mathbf{x}}), v) \le \mu(S_{\mathbf{y}_{1\to\mathrm{j}}}, v) + \mu(\mathit{Tail}(\mathrm{j}, S_{\mathbf{y}}), v) \end{aligned}$$

$$\iff \mu(Tail(\mathrm{j}, S_{\mathbf{x}}), v) \leq \mu(Tail(\mathrm{j}, S_{\mathbf{y}}), v) \qquad \text{(Since } \mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}.\text{)} \tag{1.67}$$

Since $v \in Tail(\mathrm{j}, S_{\mathbf{x}})$, we have $\mu(Tail(\mathrm{j}, S_{\mathbf{x}}), v) \geq 1$, and so

$$1 \leq \mu(Tail(\mathrm{j}, S_{\mathbf{y}}), v)\,.$$

Thus $v \in Tail(\mathrm{j}, S_{\mathbf{y}})$, so property (i) from the lemma statement holds. This completes Case 1.

Case 2: $v \notin Tail(\mathrm{j}, S_{\mathbf{x}})$**.**

If $\mathbf{x} = \mathbf{y}$, then property (ii) from the lemma statement holds.

Now suppose $\mathbf{x} \neq \mathbf{y}$. Then $Tail(\mathrm{j}, S_{\mathbf{y}}) \neq \emptyset$. We claim $v \in S_{\mathbf{x}} \cup S_{\mathbf{y}}$. Suppose towards a contradiction that $v \notin S_{\mathbf{x}} \cup S_{\mathbf{y}}$.

Then for each $b \in \{0, 1\}$, $u \in [n]$, and sequence $\mathbf{z} = (1, z_2, z_3 \ldots, z_{L+1}) \in \{1\} \times [n]^L$, we have by Eq. (1.45) (which defines the function $g_{\mathbf{z},b}$) that

$$g_{\mathbf{z},b}(u) = \begin{cases} \Big(f_{\mathbf{z}}(u), b\Big) & \text{if } u = z_{L+1} \\ \Big(f_{\mathbf{z}}(u), -1\Big) & \text{otherwise}\,. \end{cases}$$

Since $v \notin S_{\mathbf{x}}$, we have $v \neq \mathbf{x}_{L+1}$, and so $g_{\mathbf{x},b_1}(v) = (f_{\mathbf{x}}(v), -1)$. Moreover, since $x_1 = y_1$ and $v \notin S_{\mathbf{x}} \cup S_{\mathbf{y}}$, we have $f_{\mathbf{x}}(v) = dist(v, x_1) = dist(v, y_1) = f_{\mathbf{y}}(v)$. Combining these observations yields $g_{\mathbf{x},b_1}(v) = (f_{\mathbf{x}}(v), -1) = (f_{\mathbf{y}}(v), -1) = g_{\mathbf{y},b_2}(v)$, which contradicts Eq. (1.66). Thus the assumption must have been false and $v \in S_{\mathbf{x}} \cup S_{\mathbf{y}}$.

To summarize, we have $\mathbf{x}_{1\to \mathrm{j}} = \mathbf{y}_{1\to \mathrm{j}}$, $\mathbf{x} \neq \mathbf{y}$, $v \in S_{\mathbf{x}} \cup S_{\mathbf{y}}$, and $v \notin \mathit{Tail}(\mathrm{j}, S_{\mathbf{x}})$. Suppose towards a contradiction that $v \notin \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$. Then

$$\begin{aligned} g_{\mathbf{x},b_1}(v) &= (f_{\mathbf{x}}(v), -1) && \text{(Since } v \neq x_{L+1}, \text{ as } v \notin \mathit{Tail}(\mathrm{j}, S_{\mathbf{x}}).) \\ &= (f_{\mathbf{y}}(v), -1) && \text{(Since } v \in S_{\mathbf{x}} \cup S_{\mathbf{y}} \text{ and } \mathbf{x}_{1\to \mathrm{j}} = \mathbf{y}_{1\to \mathrm{j}} \text{ and } (v \notin \mathit{Tail}(\mathrm{j}, S_{\mathbf{x}}),\ v \notin \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})).) \\ &= g_{\mathbf{y},b_2}(v), && \text{(Since } v \neq y_{L+1}, \text{ as } v \notin \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}}).) \end{aligned}$$

which contradicts Eq. (1.66).

Thus the assumption must have been false and $v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$, so property (i) from the lemma statement holds.

We conclude that at least one of properties (i) and (ii) holds. This completes Case 2, as well as the proof of the lemma. □

Lemma 11. *For each* $\mathbf{x} \in \{1\} \times [n]^L$, $b_1 \in \{0, 1\}$, $v \in [n]$, *and* $\mathrm{j} \in [L]$, *we have*

$$\sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L,\ b_2 \in \{0,1\}: \\ \max\{\mathrm{i} : \mathbf{x}_{1\to \mathrm{i}} = \mathbf{y}_{1\to \mathrm{i}}\} = \mathrm{j}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \leq \sum_{\substack{\mathbf{y} \in \{1\} \times [n]^L: \\ \max\{\mathrm{i} : \mathbf{x}_{1\to \mathrm{i}} = \mathbf{y}_{1\to \mathrm{i}}\} = \mathrm{j} \\ v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \,.$$

Proof. Let $\mathbf{x} \in \{1\} \times [n]^L$, $b_1 \in \{0, 1\}$, $v \in [n]$, and $\mathrm{j} \in [L]$.

Recall the function r from Definition 1.11.6, the function r_v from Definition 1.11.7, and the function $\widetilde{r}_v$ from Definition 1.11.8. In particular, we have

- $r_v(F_1, F_2) = r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}}$ for all $F_1, F_2 \in \mathcal{X}$.
- $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot \mathbb{1}_{\{\mu(S_{\mathbf{x}},v) \leq \mu(S_{\mathbf{y}},v)\}}$ for each $\mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L$ and $b_1, b_2 \in \{0, 1\}$.

Then we get the next chain of identities:

$$\begin{aligned}
\sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L,\ b_2\in\{0,1\}:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) &= \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L,\ b_2\in\{0,1\}:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2})>0}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) && \text{(Since } \widetilde{r}_v \text{ is non-negative.)}\\
&= \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) && \text{(Since } \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_1}) = 0.)\\
&= \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \mu(S_{\mathbf{x}},v)\le\mu(S_{\mathbf{y}},v)\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) && \text{(By definition of } \widetilde{r}_v.)\\
&= \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \mu(S_{\mathbf{x}},v)\le\mu(S_{\mathbf{y}},v)\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \cdot \mathbb{1}_{\{g_{\mathbf{x},b_1}(v)\neq g_{\mathbf{y},1-b_1}(v)\}} && \text{(By definition of } r_v.)\\
&= \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \mu(S_{\mathbf{x}},v)\le\mu(S_{\mathbf{y}},v)\\ g_{\mathbf{x},b_1}(v)\neq g_{\mathbf{y},1-b_1}(v)\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1})\,. && (1.68)
\end{aligned}$$

Consider an arbitrary $\mathbf{y} \in \{1\} \times [n]^L$ meeting the properties from the last sum of Eq. (1.68):

- $\max\{\mathrm{i} : \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} = \mathrm{j}$
- $\mu(S_{\mathbf{x}}, v) \le \mu(S_{\mathbf{y}}, v)$
- $g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},1-b_1}(v)$
- $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) > 0$

By Lemma 10, the inequality $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) > 0$ implies that at least one of $v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$ or $\mathbf{x} = \mathbf{y}$ holds. However, $\mathbf{x}$ cannot be equal to such $\mathbf{y}$ as $\mathrm{j} < L + 1$ is the maximum index for which $\mathbf{x}_{1\to \mathrm{j}} = \mathbf{y}_{1\to \mathrm{j}}$. Thus $v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$. We can continue to bound the sum from Eq. (1.68) as follows:

$$\sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \mu(S_{\mathbf{x}},v)\le\mu(S_{\mathbf{y}},v)\\ g_{\mathbf{x},b_1}(v)\neq g_{\mathbf{y},1-b_1}(v)\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} r\big(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}\big) \le \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ v\in\mathit{Tail}(\mathrm{j},S_{\mathbf{y}})}} r\big(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}\big)\,. \tag{1.69}$$

Combining Eq. (1.68) and Eq. (1.69), we get the inequality required by the lemma statement:

$$\sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L,\ b_2\in\{0,1\}:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}}} \widetilde{r}_v\big(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}\big) \le \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ v\in\mathit{Tail}(\mathrm{j},S_{\mathbf{y}})}} r\big(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}\big)\,.$$

□

Lemma 12. *Let $\mathbf{x} \in \{1\} \times [n]^L$, $v \in [n]$, and $\mathrm{j} \in [L]$. Then*

$$\Big|\Big\{\mathbf{y} \in \{1\} \times [n]^L \mid \max\{\mathrm{i} : \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} = \mathrm{j} \text{ and } v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})\Big\}\Big| \le \mathfrak{q}_v(x_\mathrm{j}) \cdot n^{L-\mathrm{j}} + L \cdot g \cdot n^{L-\mathrm{j}-1}\,,$$

recalling that $\mathfrak{q}_v(u)$ is the number of paths in the set $\mathcal{P}$ that start at vertex u and contain v.

Proof. There are two ways v could be in $\mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$: either $v \in P^{y_\mathrm{j},y_{\mathrm{j}+1}}$ or $v \in P^{y_\mathrm{i},y_{\mathrm{i}+1}}$ for some $\mathrm{j} < \mathrm{i} \le L$. We will now upper bound the number of $\mathbf{y}$ for each of these possibilities separately.

(i) We count the number of sequences of vertices $\mathbf{y} = (y_1, \ldots, y_{L+1})$ with

$$\mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}} \text{ and } v \in P^{y_\mathrm{j},y_{\mathrm{j}+1}}\,. \tag{1.70}$$

There are $\mathfrak{q}_v(x_\mathrm{j})$ choices of vertices for $y_{\mathrm{j}+1}$ and $n^{L-\mathrm{j}}$ choices for sequences $(y_{\mathrm{j}+2}, \ldots, y_{L+1})$. Thus there are $\mathfrak{q}_v(x_\mathrm{j}) \cdot n^{L-\mathrm{j}}$ choices of $\mathbf{y}$ for which Eq. (1.70) holds.

(ii) For each i with $\mathrm{j} < \mathrm{i} \leq L$, we count the number of sequences of vertices $\mathbf{y} = (y_1, \ldots, y_{L+1})$ with

$$\mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}} \text{ and } v \in P^{y_\mathrm{i}, y_{\mathrm{i}+1}}\,. \tag{1.71}$$

There are $n^{L-\mathrm{j}-1}$ tuples of the form $(y_{\mathrm{j}+1}, \ldots, y_{\mathrm{i}-1}, y_{\mathrm{i}+2}, \ldots, y_{L+1})$, as these vertices can be chosen arbitrarily. Since there are at most g paths in $\mathcal{P}$ that contain v, the number of choices for the pair $(y_\mathrm{i}, y_{\mathrm{i}+1})$ is at most g as well. Thus there are $g \cdot n^{L-\mathrm{j}-1}$ choices of $\mathbf{y}$ for which Eq. (1.71) holds.

There are at most $L - \mathrm{j} \leq L$ possible locations for i when $\mathrm{j} < \mathrm{i} \leq L$. So, there are at most $L \cdot g \cdot n^{L-\mathrm{j}-1}$ choices for $\mathbf{y}$ such that Eq. (1.71) holds with some index $\mathrm{i} \in \{\mathrm{j}+1, \ldots, L\}$.

Combining the analysis from (i)-(ii), we get that:

$$\left|\left\{\mathbf{y} \in \{1\} \times [n]^L \mid \max\{\mathrm{i} : \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} = \mathrm{j} \text{ and } v \in \mathit{Tail}(\mathrm{j}, S_\mathbf{y})\right\}\right| \leq \mathfrak{q}_v(x_\mathrm{j}) \cdot n^{L-\mathrm{j}} + L \cdot g \cdot n^{L-\mathrm{j}-1}\,.$$

This completes the proof of the lemma. □

Lemma 13. *Let* $\mathrm{j} \in [L]$ *and* $\mathbf{x} \in \{1\} \times [n]^L$ *be an arbitrary good sequence of vertices. Then*

$$\left|\mathbf{y} \in \{1\} \times [n]^L \mid \mathbf{y} \textit{ is good and } \mathrm{j} = \max\{\mathrm{i} : \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}\right| = (n - \mathrm{j} - 1) \prod_{\mathrm{i}=\mathrm{j}+2}^{L+1} (n - \mathrm{i} + 1)\,.$$

Proof. Let $\mathbf{y} = (y_1, \ldots, y_{L+1}) \in [n]^{L+1}$ be such that $x_1 = y_1 = 1$, $\mathrm{j} = \max\{\mathrm{i} : \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}$, and $\mathbf{y}$ is good. Recall that for $\mathbf{y}$ to be good, each element in $(y_1, \ldots, y_{L+1})$ has to be distinct.

We will count by choosing each vertex of $\mathbf{y}$ in order from y_1 to y_{L+1}. There is only one choice for each of $y_1, \ldots, y_\mathrm{j}$ since we need $\mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}$.

Consider the number of possible vertices for $y_{\mathrm{j}+1}$:

- j vertices already appeared in $\{y_1, \dots, y_{\mathrm{j}}\}$ so they cannot be reused, otherwise $\mathbf{y}$ becomes bad.
- $\mathrm{j} = \max\{\mathrm{i} : \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}$ means $x_{\mathrm{j}+1} \neq y_{\mathrm{j}+1}$, so $x_{\mathrm{j}+1}$ cannot be reused either.

Thus, there are $n - \mathrm{j} - 1$ choices of vertices for $y_{\mathrm{j}+1}$.

For all $\mathrm{i} > \mathrm{j}+1$, there are $n - \mathrm{i} + 1$ choices of vertices for y_{i} since $\mathrm{i} - 1$ options have already been used. Thus the final count is $(n - \mathrm{j} - 1)\prod_{\mathrm{i}=\mathrm{j}+2}^{L+1}(n - \mathrm{i} + 1)$, as required. □

Lemma 14. *If $F_1 \in \mathcal{X}$ is good, then* $\sum_{F_2\in\mathcal{X}} r(F_1, F_2) \geq \frac{1}{2\mathrm{e}} \cdot (L+1) \cdot n^{L+1}$.

Proof. Since F_1 is good, there exists a good sequence $\mathbf{x} \in \{1\} \times [n]^L$ and a bit $b_1 \in \{0,1\}$ such that $F_1 = g_{\mathbf{x},b_1}$. Then we have the following chain of identities:

$$\begin{aligned}
\sum_{F_2\in\mathcal{X}} r(F_1, F_2) &= \sum_{F_2\in\mathcal{X}} r(g_{\mathbf{x},b_1}, F_2) \\
&= \sum_{\mathbf{y}\in\{1\}\times[n]^L,\ b_2\in\{0,1\}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) && \text{(By definition of the set } \mathcal{X}) \\
&= \sum_{\mathbf{y}\in\{1\}\times[n]^L} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) && (\text{Since } r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_1}) = 0) \\
&= \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L \\ \mathrm{j}=\max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \,. && (1.72)
\end{aligned}$$

Since $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) = 0$ if $\mathbf{y}$ is bad, we have

$$\sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L: \\ \mathrm{j}=\max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) = \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L: \\ \mathrm{j}=\max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\} \\ \mathbf{y} \text{ is good}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \,. \tag{1.73}$$

Combining Eq. (1.72) and Eq. (1.73), the last sum in Eq. (1.73) can be written as:

$$\sum_{F_2\in\mathcal{X}} r(g_{\mathbf{x},b_1}, F_2) = \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L: \\ \mathrm{j}=\max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\} \\ \mathbf{y} \text{ is good}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \,. \tag{1.74}$$

Substituting the definition of r in Eq. (1.74), we get

$$\sum_{F_2\in\mathcal{X}} r(g_{\mathbf{x},b_1}, F_2) = \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \mathrm{j}=\max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}\\ \mathbf{y}\text{ is good}}} n^{\mathrm{j}}\,. \tag{1.75}$$

From here, we only need to count the number of such $\mathbf{y}$. There is exactly one good sequence $\mathbf{y}\in\{1\}\times[n]^L$ such that $\mathbf{x}_{1\to L+1}=\mathbf{y}_{1\to L+1}$, namely $\mathbf{y}=\mathbf{x}$. Therefore,

$$\begin{aligned}
\sum_{F_2\in\mathcal{X}} r(g_{\mathbf{x},b_1}, F_2) &= \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \mathbf{y}\text{ is good}}} n^{\mathrm{j}} && \text{(By Eq. (1.75))}\\
&= n^{L+1} + \sum_{\mathrm{j}=1}^{L} \sum_{\substack{\mathbf{y}\in\{1\}\times[n]^L:\\ \max\{\mathrm{i}:\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \mathbf{y}\text{ is good}}} n^{\mathrm{j}}\\
&\text{(Can pull out } n^{L+1} \text{ since there is only one possibility of } \mathbf{x}_{1\to L+1}=\mathbf{y}_{1\to L+1} \text{ when } \mathbf{x}=\mathbf{y})\\
&= n^{L+1} + \sum_{\mathrm{j}=1}^{L} n^{\mathrm{j}} \left[(n-\mathrm{j}-1)\cdot \prod_{\mathrm{i}=\mathrm{j}+2}^{L+1}(n-\mathrm{i}+1)\right] && \text{(By Lemma 13)}\\
&\geq n^{L+1} + \sum_{\mathrm{j}=1}^{L}(n-L-1)^{L+1} && \text{(Since j} \in [L])\\
&\geq (L+1)\cdot n^{L+1}\cdot\left(1-\frac{L+1}{n}\right)^{L+1}\\
&\geq (L+1)\cdot n^{L+1}\cdot\left(1-\frac{1}{L+1}\right)^{L+1} && \text{(Since } L+1\leq\sqrt{n})\\
&\geq \frac{1}{2\mathrm{e}}\cdot(L+1)\cdot n^{L+1}\,. && \text{(Since } n\geq 4\text{, so } L+1\geq 2)
\end{aligned}$$

Since $F_1 = g_{\mathbf{x},b_1}$, this is the required inequality, which completes the proof. □

1.11.4 Corollaries for expanders

Given the lower bound based on congestion, we can now state an implication for expanders. For this we rely on the next corollary of a result from [19], which shows that d-regular expanders have systems of paths with low vertex congestion when the degree d is constant.

Lemma 15. *Let $G = ([n], E)$ be a d-regular β-expander, where d and β are constant. Consider the collection of all n^2 ordered pairs of vertices $\{(a_1, b_1), \ldots, (a_{n^2}, b_{n^2})\}$, including each vertex with itself. Then there exists a set $\mathcal{P}$ of n^2 paths $\{P_1, \ldots, P_{n^2}\}$, such that each path P_i connects a_i to b_i and the congestion on each vertex of G is $O(n \ln n)$.*

Proof. We invoke Theorem 1.8.1 with parameters $K = n^2$, $\alpha(n) = n \ln n$, and $s = 2n$ to get a set of K paths $\mathcal{P} = \{P_1, \ldots, P_K\}$. Since $\alpha(n) \geq 1/2$, we get that the edge congestion g of $\mathcal{P}$ is at most:

$$g \in O(s + \alpha + \ln \ln n) = O(n \ln n)\,.$$

To convert from edge congestion to vertex congestion, consider the vertex v with the highest vertex congestion with respect to $\mathcal{P}$. The vertex congestion at v is no more than the sum of the edge congestions on each of the edges incident to v. Because G is d-regular, this means the vertex congestion at v is at most $d \cdot g$. Since d is constant, the vertex congestion is $O(n \ln n)$. □

Using this corollary, we get the following result for local search on expanders.

Corollary 1. *Let $G = (V, E)$ be an undirected d-regular β-expander with n vertices, where d and β are constant. Then the randomized query complexity of local search on G is $\Omega\left(\frac{\sqrt{n}}{\log n}\right)$.*

Proof. By Lemma 15, the graph G has an all-pairs set of paths $\mathcal{P}$ with vertex congestion $g \in O(n \ln n)$. By Theorem 1.3.2, the randomized query complexity of local search on G is $\Omega(\sqrt{n}/\ln n)$. □

By alternatively using a prior result from [48], we get a result in terms of the expansion and maximum degree for any graph.

Lemma 16. *Let G be an undirected β-expander with maximum degree Δ. Then G has vertex congestion $g \in O(n \ln^2(n) \cdot \frac{\Delta}{\beta})$.*

Proof. Decompose the clique on n vertices into n partial matchings $M_1, M_2, \ldots, M_n$. Invoke Theorem 1.8.2 on each partial matching M_i to get a set $\mathcal{P}_{u,v}$ of $\lceil \ln n \rceil$ paths from u to v for every pair of vertices $\{u, v\} \in M_i$. Let P_i be an arbitrary path from $\mathcal{P}_{a_i,b_i}$ for every $i \in [n^2]$.

By Theorem 1.8.2, the edge congestion for each partial matching is at most $O(\ln^2(n)/\beta)$. Therefore the edge congestion of the union of the results of invoking Theorem 1.8.2 is at most $O(n \cdot \ln^2(n)/\beta)$. Since $\mathcal{P}$ contains only one path from each $\mathcal{P}_{u,v}$, it also has edge congestion at most $O(n \cdot \ln^2(n)/\beta)$.

To convert from edge congestion to vertex congestion, consider the vertex v with the highest vertex congestion with respect to $\mathcal{P}$. The vertex congestion at v is no more than the sum of the edge congestions on each of the edges incident to v. Because the maximum degree is Δ, this means the vertex congestion at v is at most $O(n \ln^2(n) \cdot \frac{\Delta}{\beta})$, so the vertex congestion of $O(n \ln^2(n) \cdot \frac{\Delta}{\beta})$. ☐

Using this congestion result, we get a corollary for expansion and maximum degree.

Corollary 2. *Let $G = (V, E)$ be an undirected β-expander with n vertices and maximum degree Δ. Then the randomized query complexity of local search on G is $\Omega\left(\frac{\beta\sqrt{n}}{\Delta \log^2 n}\right)$.*

Proof. By Lemma 16, G has an all-pairs set of paths $\mathcal{P}$ with vertex congestion $g \in O\left(n \cdot \ln^2(n) \cdot \Delta/\beta\right)$. By Theorem 1.3.2, the randomized query complexity of local search on G is $\Omega\left(\frac{\beta\sqrt{n}}{\Delta \log^2 n}\right)$. ☐

1.11.5 Corollary for Cayley graphs

We also get a corollary for undirected Cayley graphs thanks to a construction by [16], which has vertex congestion of at most $(d + 1) \cdot n$. This improves the result of [16] for randomized algorithms by a $\ln n$ factor.

Lemma 17. *Let $G = (V, E)$ be an undirected Cayley graph with n vertices and diameter $diam(G)$. Then there exists an all-pairs set of paths $\mathcal{P} = \{P^{u,v}\}_{u,v \in V}$, such that each path $P^{u,v}$ connects u to v and the congestion on each vertex of G is at most $(diam(G) + 1) \cdot n$.*

Proof. Let $1 \in V$ be the group identity of G. For each $v \in V$, fix $P^{1,v}$ to be an arbitrary shortest path from 1 to v. Then for each pair $u, v \in V$ with $u \neq 1$, let $P^{u,v} = u \cdot P^{1,w}$, where $w = u^{-1} \cdot v$. By construction, $P^{u,v}$ starts at $u \cdot 1 = u$ and ends at $u \cdot u^{-1} \cdot v = v$.

For each $w \in V$, let $\mathcal{P}_w = \{P^{u,v} : u^{-1} \cdot v = w\}$. For all $w, x \in V$, exactly $|P^{1,w}|$ paths in $\mathcal{P}_w$ contain x: one has x in the first position, one has x in the second position, etc. Therefore $\mathcal{P}_w$ has the same vertex congestion at every vertex. Then since $\mathcal{P}$ is the disjoint union $\bigcup_{w \in V} \mathcal{P}_w$, we get that $\mathcal{P}$ has the same vertex congestion at every vertex.

Every path in $\mathcal{P}$ is a shortest path, and so has length at most $diam(G)+1$ vertices. Therefore in total there are $(diam(G)+1) \cdot n^2$ vertices in $\mathcal{P}$, so the vertex congestion of $\mathcal{P}$ is $(diam(G)+1) \cdot n$ since every vertex has the same congestion. □

Using this lemma we get the following result for local search on undirected Cayley graphs.

Corollary 3. *Let $G = (V, E)$ be an undirected Cayley graph with n vertices and diameter $diam(G)$. Then the randomized query complexity of local search on G is $\Omega\left(\frac{\sqrt{n}}{diam(G)}\right)$.*

Proof. By Lemma 17, the graph G has an all-pairs set of paths $\mathcal{P}$ that has vertex congestion $g \leq (diam(G)+1) \cdot n$. By Theorem 1.3.2, the randomized query complexity of local search on G is $\Omega\left(\frac{\sqrt{n}}{diam(G)}\right)$. □

1.11.6 Corollary for the hypercube

Finally, we get a corollary for the $\{0,1\}^n$ hypercube.

Corollary 4. *The randomized query complexity of local search on the Boolean hypercube $\{0,1\}^n$ is $\Omega\left(\frac{2^{n/2}}{n}\right)$.*

Proof. We first quantify the vertex congestion g of the Boolean hypercube, and then invoke Theorem 1.3.2 to obtain a lower bound for local search.

The number of vertices in the $\{0,1\}^n$ hypercube is $N = 2^n$. The vertices of the graph can be viewed as bit strings of length n, with bit strings of Hamming distance 1 connected by an edge. Fix an order on the bits. Then for every pair of vertices u, v, the path $P^{u,v}$ is obtained by iterating over the bits, toggling each bit that differs between u and v.

By symmetry of the construction, the system of paths $\mathcal{P}$ has equal congestion at every vertex. Furthermore, the average length of a path in $\mathcal{P}$ is $1 + n/2$. Thus the congestion of $\mathcal{P}$ is $N \cdot (1 + n/2)$. Since every path is a shortest path and the congestion is evenly distributed, this is optimal and the congestion of the graph is $g = N \cdot (1 + n/2)$.

By Theorem 1.3.2, the randomized query complexity of local search on the hypercube is $\Omega\left(\frac{2^{n/2}}{n}\right)$. □

1.12 Lower bound for local search via separation number: proofs

In this section we include the proofs needed to show the lower bound of $\Omega\left((s/\Delta)^{1/4}\right)$.

We start with the basic definitions. Then we state and prove the lower bound. Afterwards, we show the helper lemmas used in the proof of the theorem.

1.12.1 Basic definitions for separation

Notation.

Recall we have a graph $G = ([n], E)$ with separation number s and maximum degree Δ. This means that every subset $H \subseteq [n]$ can be split in two parts, S and $H \setminus S$, each of size at least $|H|/4$, such that at most s vertices in $H \setminus S$ are adjacent to S.

Let $c \in \mathbb{N}$ be a parameter that we set later.

Given a sequence of k indices $\mathbf{x} = (x_1, \ldots, x_k)$, we write $\mathbf{x}_{1 \to \mathrm{j}} = (x_1, \ldots, x_{\mathrm{j}})$ to refer to a prefix of the sequence, for an index $\mathrm{j} \in [k]$.

Given a walk $Q = (v_1, \ldots, v_k)$ in G, let Q_{i} refer to the i-th vertex in the walk (i.e. $Q_{\mathrm{i}} = v_{\mathrm{i}}$). For each vertex $u \in [n]$, let $\mu(Q, u)$ be the number of times that vertex u appears in Q.

Next we introduce the notion of inter-cluster paths from [17].

Definition 1.12.1 (Path Arrangement and Inter/Intra-Cluster Paths)**.** *A* path arrangement *with parameter m for graph $G = ([n], E)$ is a set of connected, disjoint subsets $N_1, \ldots, N_m \subseteq V$ with the following property: for all $\mathrm{i}, \mathrm{j} \in [m]$, there exist m paths $P_1(\mathrm{i}, \mathrm{j}), \ldots, P_m(\mathrm{i}, \mathrm{j})$ such that*

- *For each $k \in [m]$, the first vertex of $P_k(\mathrm{i}, \mathrm{j})$ is in N_{i}, the last vertex of $P_k(\mathrm{i}, \mathrm{j})$ is in N_{j}, and every other vertex of $P_k(\mathrm{i}, \mathrm{j})$ is outside N_{i} and N_{j}.*
- *For every pair $\mathrm{i}, \mathrm{j} \in [m]$, vertices in $V \setminus (N_{\mathrm{i}} \cup N_{\mathrm{j}})$ are visited at most once collectively by the paths $P_1(\mathrm{i}, \mathrm{j}), \ldots, P_m(\mathrm{i}, \mathrm{j})$.*

Observation 1.12.1. *Definition 1.12.1 differs slightly from the original construction of [17] in that Definition 1.12.1 uses inter-cluster paths of the form $P_k(\mathrm{i}, \mathrm{i})$; i.e. from a cluster to itself. However, $P_k(\mathrm{i}, \mathrm{i})$ may always be chosen as a degenerate single-vertex path in N_{i}, so this deviation does not affect the value of m for any graph.*

(76) Let H be the grid of Figure 1.4. There exists a path arrangement for H with parameter $\sqrt{n}$ since one can use the columns as the N_{i} and the shortest path in each row as the inter-cluster paths.

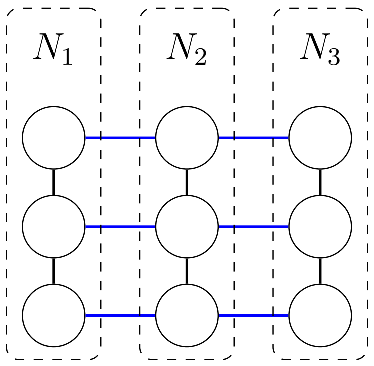


Figure 1.4. Let H be the $\sqrt{n} \times \sqrt{n}$ graph with clusters $N_1, \ldots, N_{\sqrt{n}}$, where $n = 9$. The black edges represent intra-cluster paths while the blue edges represent inter-cluster paths. Then, there exists a path arrangement for this graph with parameter $\sqrt{n} = 3$.

Next we present without proof a lemma from [17], which relates the path arrangement parameter to the separation number and maximum degree of the graph.

Lemma 18 ([17], Theorem 6)**.** *If an undirected graph G has maximum degree Δ and separation number s, then there exist a path arrangement on G with parameter at least* $\max\{\lfloor\sqrt{s/2\Delta}\rfloor, 1\}$.

Path arrangement number m of the graph G.

Let m be the maximum number such that there exists a path arrangement on our graph G with parameter m. Let $N_1, \ldots, N_m$ be the corresponding clusters. An example of a set of inter-cluster paths between N_{i} and N_{j} from [17] is shown in Figure 1.5.

For all $\text{i} \in [m]$ and $u, v \in N_{\text{i}}$, let $E_{\text{i}}(u, v)$ be an arbitrary shortest path from u to v within N_{i}.

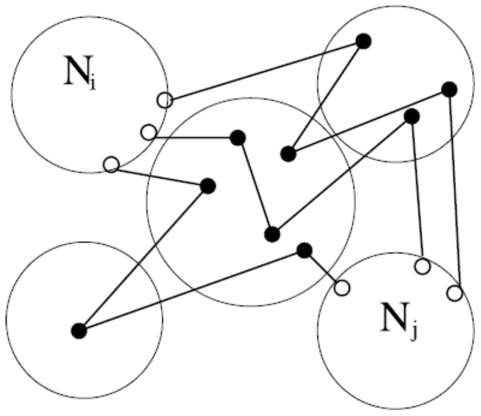


Figure 1.5. A set of inter-cluster paths between N_{i} and N_{j} from [17].

Define $first(P_{\mathrm{i}}(\mathrm{j},k))$ and $last(P_{\mathrm{i}}(\mathrm{j},k))$ as the first and last vertices in $P_{\mathrm{i}}(\mathrm{j},k)$, respectively. We have $first(P_{\mathrm{i}}(\mathrm{j},k)) \in N_{\mathrm{j}}$ and $last(P_{\mathrm{i}}(\mathrm{j},k)) \in N_k$, by definition of $P_{\mathrm{i}}(\mathrm{j},k)$.

We will map a sequence of indices to walks starting from a fixed starting vertex $v_{start} \in N_1$.

Definition 1.12.2 (Staircase; separation version)**.** *For* $\mathbf{x} = (x_1 = 1, x_2, \ldots, x_{2c+1}) \in \{1\} \times [m]^{2c}$*, and* $\mathrm{i} \in [2c]$*, we define*

$$S_{\mathbf{x},\mathrm{i}} = \begin{cases} E_1(v_{start}, first(P_{x_2}(1, x_3)) & \textit{if } \mathrm{i} = 1 \\ P_{x_{\mathrm{i}}}(x_{\mathrm{i}-1}, x_{\mathrm{i}+1}) & \textit{if } \mathrm{i} \textit{ is even} \\ E_{x_{\mathrm{i}}}(last(Q_{x_{\mathrm{i}-1}}(x_{\mathrm{i}-2}, x_{\mathrm{i}})), first(P_{x_{\mathrm{i}+1}}(x_{\mathrm{i}}, x_{\mathrm{i}+2}))) & \textit{if } \mathrm{i} > 1 \textit{ is odd} \end{cases} \tag{1.77}$$

Then for all sequences of indices $\mathbf{x} = (1, x_2, \ldots, x_{2k+1}) \in \{1\} \times [m]^{2k}$*, the full walk* $S_{\mathbf{x}}$ *induced by* $\mathbf{x}$ *is the concatenation* $S_{\mathbf{x}} = S_{\mathbf{x},1} \circ S_{\mathbf{x},2} \circ \ldots \circ S_{\mathbf{x},2k}$*.*

That is, we use the inter-cluster paths dictated by the even indices of $\mathbf{x}$ to travel between clusters dictated by the odd indices of $\mathbf{x}$, stitching the inter-cluster paths together using shortest paths within the clusters. So $S_{\mathbf{x}}$ starts at $v_{start} \in N_1$, travels via $S_{\mathbf{x},1}$ and $S_{\mathbf{x},2}$ to a vertex in N_{x_3}, and so on. Observe $S_{\mathbf{x}}$ is a well-defined walk since $S_{\mathbf{x},\mathrm{i}}$ connects the last vertex of $S_{\mathbf{x},\mathrm{i}-1}$ and first vertex of $S_{\mathbf{x},\mathrm{i}+1}$ for any odd $\mathrm{i} > 1$. An illustrated example is given next.

(78) Consider the graph in Figure 1.6. Suppose $m = 3$. For each $\mathrm{i} \in \{1, 2, 3\}$, let the i^{th}

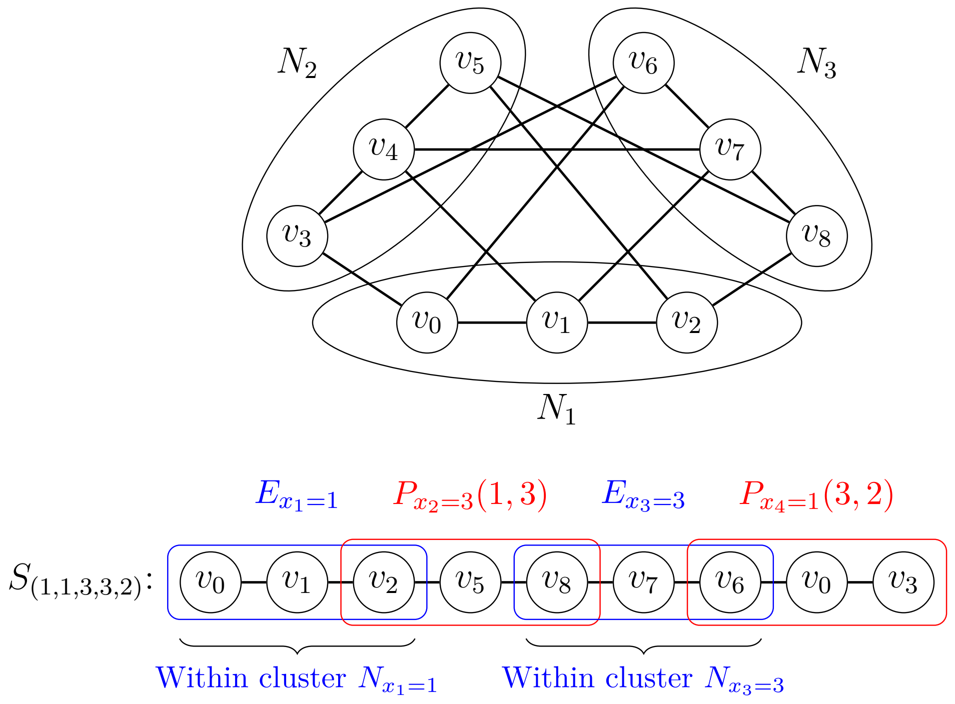

N_2
N_3
N_1
v_0 v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8
$E_{x_1=1}$
$P_{x_2=3}(1,3)$
$E_{x_3=3}$
$P_{x_4=1}(3,2)$
$S_{(1,1,3,3,2)}$:
v_0 v_1 v_2 v_5 v_8 v_7 v_6 v_0 v_3
Within cluster $N_{x_1=1}$
Within cluster $N_{x_3=3}$

Figure 1.6. Graph on nine nodes, with connected cluster partitions $N_1 = \{v_0, v_1, v_2\}$,
$N_2 = \{v_3, v_4, v_5\}$, and $N_3 = \{v_6, v_7, v_8\}$.

path for each pair of clusters consist of vertices $\{v_{\mathrm{i}-1},\ v_{\mathrm{i}-1+m}, v_{\mathrm{i}-1+2m}\}$.

For example, $P_2(2, 3) = (v_4, v_1, v_7)$ is the second path from N_2 to N_3 and $P_2(1, 3) = (v_1, v_4, v_7)$ is the second path from N_1 to N_3.

Fix $v_{start} = v_0$ and let $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5) = (1, 1, 3, 3, 2)$. We have $P_3(1, 3) = (v_2, v_5, v_8)$ and $P_1(3, 2) = (v_6, v_0, v_3)$. Thus

$$\begin{aligned} S_{\mathbf{x}} &= S_{\mathbf{x},1} \circ S_{\mathbf{x},2} \circ S_{\mathbf{x},3} \circ S_{\mathbf{x},4} \\ &= E_{x_1=1}(v_0, v_2) \circ P_{x_2=3}(1, 3) \circ E_{x_3=3}(v_8, v_6) \circ P_{x_4=1}(3, 2) \end{aligned}$$

$$= (v_0, v_1, v_2) \circ (v_2, v_5, v_8) \circ (v_8, v_7, v_6) \circ (v_6, v_0, v_3)$$
$$= (v_0, v_1, v_2, v_5, v_8, v_7, v_6, v_0, v_3) .$$

Definition 1.12.3 (Tail of a staircase; separation version)**.** *Let* $S_{\mathbf{x}} = S_{\mathbf{x},1} \circ \ldots \circ S_{\mathbf{x},2k}$ *be a staircase induced by some sequence* $\mathbf{x} = (1, x_2, \ldots, x_{2k+1}) \in \{1\} \times [m]^{2k}$*. For each odd* $\mathrm{j} \in [2k]$*, let* $T = S_{\mathbf{x},\mathrm{j}} \circ \ldots \circ S_{\mathbf{x},2k}$*. Then* $Tail(\mathrm{j}, S_{\mathbf{x}})$ *is obtained from* T *by removing the first occurrence of the first vertex in* T *(and only the first occurrence). Let* $Tail(2k+1, S_{\mathbf{x}})$ *be the empty sequence.*

Observe $Tail(\mathrm{j}, S_{\mathbf{x}})$ *is only defined for odd* j *since staircase* $S_{\mathbf{x}_{1\to\mathrm{j}}}$ *is only defined for odd* j.

Next, we define the set of functions $\mathcal{X}$ that will be used when invoking Theorem 1.3.1.

Definition 1.12.4 (The functions $f_{\mathbf{x}}$ and $g_{\mathbf{x},b}$; the set $\mathcal{X}$; separation version)**.** *Given an input graph* G*, let* m *be its path arrangement parameter and* $N_1, \ldots, N_m$ *be the clusters with respect to Definition 1.12.1. For each sequence of indices* $\mathbf{x} \in \{1\} \times [m]^{2c}$*, define* $f_{\mathbf{x}} : [n] \to \{-n^2, -n^2+1, \ldots, n\}$ *such that for each* $v \in [n]$*:*

- *If* $v \notin S_{\mathbf{x}}$*, then* $f_{\mathbf{x}}(v) = dist(v, v_{start})$*, where* $S_{\mathbf{x}}$ *is the staircase induced by* $\mathbf{x}$ *and the cluster construction.*
- *If* $v \in S_{\mathbf{x}}$*, then* $f_{\mathbf{x}}(v) = -\mathrm{i}$*, where* i *is the maximum index such that* v *is the* i*-th vertex in* $S_{\mathbf{x}}$*.*

Also, for each $\mathbf{x} \in \{1\} \times [m]^{2c}$ *and* $b \in \{0,1\}$*, let* $g_{\mathbf{x},b} : [n] \to \{-n^2, \ldots, 0, \ldots, n\} \times \{-1, 0, 1\}$ *be such that, for all* $v \in [n]$*:*

$$g_{\mathbf{x},b}(v) = \begin{cases} (f_{\mathbf{x}}(v), b) & \text{if } v \text{ is the last vertex in } S_{\mathbf{x}} \\ (f_{\mathbf{x}}(v), -1) & \text{if } v \text{ is not the last vertex in } S_{\mathbf{x}} . \end{cases}$$

Let $\mathcal{X} = \{g_{\mathbf{x},b} \mid \mathbf{x} \in \{1\} \times [m]^{2c} \text{ and } b \in \{0,1\}\}$.

Definition 1.12.5 (The map $\mathcal{H}$; separation version)**.** *Given an input graph G, let m be its path arrangement parameter and $N_1, \ldots, N_m$ be the clusters with respect to Definition 1.12.1. Let $\mathcal{X}$ be the set of functions $g_{\mathbf{x},b}$ from Definition 1.12.4. Define $\mathcal{H} : \mathcal{X} \to \{0,1\}$ as*

$$\mathcal{H}(g_{\mathbf{x},b}) = b \qquad \forall \mathbf{x} \in \{1\} \times [m]^{2c} \text{ and } b \in \{0,1\}\,.$$

Definition 1.12.6 (Good/bad sequences of indices; Good/bad functions; separation version)**.** *A sequence of k indices $\mathbf{x} = (x_1, \ldots, x_k)$ is* good *if $x_\mathrm{i} \neq x_\mathrm{j}$ for all i, j with $1 \leq \mathrm{i} < \mathrm{j} \leq k$; otherwise, $\mathbf{x}$ is bad. For each $b \in \{0,1\}$, a function $F = g_{\mathbf{x},b} \in \mathcal{X}$ is good if $\mathbf{x}$ is good, and bad otherwise.*

Definition 1.12.7 (The function r; separation version)**.** *Let $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be defined by, for all $X, Y \in \{1\} \times [m]^{2c}$ and $b_1, b_2 \in \{0,1\}$, letting*

$$r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = \begin{cases} 0 & \text{if at least one of the following holds: } b_1 = b_2,\ \mathbf{x} \text{ is bad, } \mathbf{y} \text{ is bad} \\ m^{\mathrm{j}} & \text{otherwise, where } \mathrm{j} \text{ is the maximum odd index for which } \mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}} \end{cases}$$

Note the specification that j be odd in the definition of r. Intuitively, this is because it takes two indices to specify an additional portion of $S_{\mathbf{x}}$: one for the destination cluster and one for the inter-cluster path by which to reach it.

The function r will be used directly to invoke Theorem 1.3.1, but we also define here some related helper functions to use r in conjunction with certain indicator variables.

Definition 1.12.8 (The function r_v; separation version)**.** *For each $v \in [n]$, define $r_v : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ as follows:*

$$r_v(F_1, F_2) = r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} \qquad \forall F_1, F_2 \in \mathcal{X}\,.$$

Definition 1.12.9 (The function $\widetilde{r}_v$; separation version)**.** *For each $v \in [n]$, define $\widetilde{r}_v : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ as follows:*

$$\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot \mathbb{1}_{\{\mu(S_{\mathbf{x}},v) \leq \mu(S_{\mathbf{y}},v)\}} \qquad \forall \mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L \;\; \forall b_1, b_2 \in \{0,1\}\,.$$

1.12.2 Proof of the separation number lower bound

Next we give the proof of Theorem 1.3.3. The proofs of lemmas used in the theorem are included afterwards in Section 1.12.3.

Theorem 1.3.3. *Let $G = (V, E)$ be a connected undirected graph with n vertices, maximum degree Δ, and separation number s. Then the randomized query complexity of local search on G is $\Omega\left(\sqrt[4]{\frac{s}{\Delta}}\right)$.*

Proof. Because we are proving an asymptotic bound in s/Δ, we may assume $s/\Delta \geq 162$. Applying Lemma 18 with the assumption that $s/\Delta \geq 162$ gives $m \geq 9$, so $c \geq 1$. Additionally, we know $\Delta \geq 1$ and $s \leq n$ by definitions of Δ and s. Therefore $n \geq s/\Delta$, so we may also assume $n \geq 9$.

Consider the following setting of parameters:

(a) Let m be its path arrangement parameter and $N_1, \ldots, N_m$ be the clusters with respect to Definition 1.12.1.

(b) Let $c = \lfloor \sqrt{m}/2 - 1/2 \rfloor$ and each staircase has $2c$ quasisegments.

(c) The finite set A is the set of vertices $[n]$.

(d) The finite set B is $\{-n^2, \ldots, n\} \times \{-1, 0, 1\}$.

(e) The set of functions $f_{\mathbf{x}}$, $g_{\mathbf{x},b}$ and the set $\mathcal{X}$ as defined in Definition 1.12.4. Recall $g_{\mathbf{x},b} = (f_{\mathbf{x}}, c)$ for all $v \in [n]$, where $c = -1$ if v is not the last vertex of the induced staircase $S_{\mathbf{x}}$, and $c = b$ if v is (i.e. $c = b$ if and only if v is a local minimum of $f_{\mathbf{x}}$). Also recall $\mathcal{X} = \{g_{\mathbf{x},b} \mid \mathbf{x} \in \{1\} \times [m]^{2c} \text{ and } b \in \{0,1\}\}$.

(f) Map $\mathcal{H} : \mathcal{X} \to \{0,1\}$ as in Definition 1.12.5. Recall $\mathcal{H}(g_{\mathbf{x},b}) = b$ for all $\mathbf{x} \in \{1\} \times [m]^{2c}$ and $b \in \{0,1\}$.

(g) The function r as defined in Definition 1.12.7.

By Lemma 19, each function $f_{\mathbf{x}}$ is valid for all $\mathbf{x} \in \{1\} \times [m]^{2c}$, so Lemma 6 implies that each function $f_{\mathbf{x}}$ has a unique local minimum (at the last vertex of $S_{\mathbf{x}}$). Therefore by Lemma 2 invoked with $f = f_{\mathbf{x}}$ and $h_b = g_{\mathbf{x},b}$, it suffices to show a lower bound for the corresponding decision problem: return the hidden bit $b \in \{0,1\}$ given oracle access to the function $g_{\mathbf{x},b}$.

For each $\mathcal{Z} \subseteq \mathcal{X}$, let

$$M(\mathcal{Z}) = \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2)\,. \tag{1.79}$$

Since by assumption $n \geq 9$, we may invoke Lemma 20 to get a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$. Thus the conditions required by Theorem 1.3.1 are met. By invoking Theorem 1.3.1 with the parameters in (a-g), we get that the randomized query complexity of the decision problem, and thus also of local search on G, is

$$\Omega\left(\min_{\mathcal{Z} \subseteq \mathcal{X} : q(\mathcal{Z}) > 0} \frac{M(\mathcal{Z})}{q(\mathcal{Z})}\right), \text{ where } q(\mathcal{Z}) = \max_{v \in [n]} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}}\,.$$

To get an explicit lower bound in terms of congestion, we will upper bound $q(\mathcal{Z})$ and lower bound $M(\mathcal{Z})$ for subsets $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$.

Fix an arbitrary subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$. Since $r(F_1, F_2) = 0$ when F_1 or F_2 is bad, it suffices to consider subsets $\mathcal{Z} \subseteq \mathcal{X}$ where each function $F \in \mathcal{Z}$ is good.

Upper bounding $q(\mathcal{Z})$.

Let $v \in [n]$ be arbitrary.

Fix an arbitrary good function $F_1 = g_{\mathbf{x},b_1} \in \mathcal{Z}$ for some good $\mathbf{x} \in \{1\} \times [m]^{2c}$ and $b_1 \in \{0,1\}$.

Since $\mathcal{Z} \subseteq \mathcal{X}$ and $\widetilde{r}_v \geq 0$, we have

$$\sum_{F_2 \in \mathcal{Z}} \widetilde{r}_v(F_1, F_2) \leq \sum_{F_2 \in \mathcal{X}} \widetilde{r}_v(F_1, F_2)\,. \tag{1.80}$$

Using the definition of $\mathcal{X} = \{g_{\mathbf{y},b_2} \mid \mathbf{y} \in \{1\} \times [m]^{2c},\ b_2 \in \{0,1\}\}$, the fact that $F_1 = g_{\mathbf{x},b_1}$, and partitioning the space of functions $F_2 \in \mathcal{X}$ by the length of the prefix that the staircase corresponding to F_2 shares with the staircase corresponding to F_1, we can upper bound the right hand of Eq. (1.80):

$$\begin{aligned}\sum_{F_2 \in \mathcal{X}} \widetilde{r}_v(F_1, F_2) &= \sum_{\mathbf{y} \in \{1\} \times [m]^{2c},\ b_2 \in \{0,1\}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \\ &\leq \sum_{\mathrm{j}=1}^{2c+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c},\ b_2 \in \{0,1\} \\ \mathrm{j} = \max\{\mathrm{i}\,:\,\mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2})\,. \end{aligned} \tag{1.81}$$

Combining Eq. (1.80) and Eq. (1.81), we get

$$\sum_{F_2 \in \mathcal{Z}} \widetilde{r}_v(F_1, F_2) \leq \sum_{\mathrm{j}=1}^{2c+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c},\ b_2 \in \{0,1\} \\ \mathrm{j} = \max\{\mathrm{i}\,:\,\mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2})\,. \tag{1.82}$$

For each $\mathrm{j} \in [2c+1]$, let

$$T_{\mathrm{j}} = \left| \left\{ \mathbf{y} \in \{1\} \times [m]^{2c} \mid \max\{\mathrm{i} : \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} = \mathrm{j} \text{ and } v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}}) \right\} \right|.$$

Then for each odd $\mathrm{j} \in [2c+1]$ such that $v \notin N_{x_{\mathrm{j}}}$, we can bound the part of the sum in Eq. (1.82) corresponding to index j via the next chain of inequalities:

$$\sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c},\ b_2 \in \{0,1\}: \\ \mathrm{j} = \max\{\mathrm{i}\,:\,\mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2})$$

$$
\begin{aligned}
&\leq \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}:\\ \mathrm{j}=\max\{\mathrm{i}\,:\,\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}\\ v\in Tail(\mathrm{j},S_{\mathbf{y}})}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) && \text{(By Lemma 22)}\\
&\leq m^{\mathrm{j}}\cdot|T_{\mathrm{j}}| && \text{(Since } r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1})\leq m^{\mathrm{j}} \text{ when } \mathrm{j}=\max\{\mathrm{i} : \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\})\\
&\leq m^{\mathrm{j}}\cdot\Big((2c+1)\cdot m^{2c-\mathrm{j}}\Big) && \text{(By Lemma 23)}\\
&= (2c+1)\cdot m^{2c} && (1.83)
\end{aligned}
$$

Meanwhile, since $\mathbf{x}$ is good, there is at most one odd j such that $v\in N_{x_{\mathrm{j}}}$. For that index j, since $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2})>0$ implies $b_2=1-b_1$ (see observation 1.11.1), we have

$$
\begin{aligned}
\sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c},\ b_2\in\{0,1\}:\\ \mathrm{j}=\max\{\mathrm{i}\,:\,\mathrm{i}\text{ is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) &\leq m^{\mathrm{j}}\cdot\Big|\{\mathbf{y}\in\{1\}\times[m]^{2c} \mid \mathbf{x}_{1\to\mathrm{j}}=\mathbf{y}_{1\to\mathrm{j}}\}\Big|\\
&= m^{\mathrm{j}}\cdot m^{2c+1-\mathrm{j}} = m^{2c+1}\,. \qquad (1.84)
\end{aligned}
$$

Summing Eq. (1.83) to Eq. (1.84), we can now upper bound the right hand side of Eq. (1.82) as follows:

$$
\begin{aligned}
\sum_{F_2\in\mathcal{Z}}\widetilde{r}_v(F_1,F_2) &\leq \sum_{\mathrm{j}=1}^{2c+1}\ \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c},\ b_2\in\{0,1\}\\ \mathrm{j}=\max\{\mathrm{i}\,:\,\mathrm{i}\text{ is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) && \text{(By Eq. (1.82))}\\
&\leq (c+1)\cdot(2c+1)\cdot m^{2c}+m^{2c+1} && \text{(By Eq. (1.83) and Eq. (1.84))}\\
&\leq 3\cdot m^{2c+1}\,. && \text{(Since } c\leq\sqrt{m}-1)
\end{aligned}
$$

Thus, for each good function $F_1\in\mathcal{Z}$, we have

$$
\sum_{F_2\in\mathcal{Z}}\widetilde{r}_v(F_1,F_2)\leq 3\cdot m^{2c+1}\,. \tag{1.85}
$$

Summing Eq. (1.85) over all $F_1 \in \mathcal{Z}$ (each of which is good, since $\mathcal{Z}$ was chosen to have good functions only), and invoking Lemma 9 yields

$$\begin{aligned}
\sum_{F_1,F_2\in\mathcal{Z}} r_v(F_1,F_2) &\le 2\cdot \sum_{F_1,F_2\in\mathcal{Z}} \widetilde{r}_v(F_1,F_2) && \text{(By Lemma 9)}\\
&\le 2\cdot 3\cdot m^{2c+1} && \text{(By Eq. (1.85))}\\
&= |\mathcal{Z}|\cdot 6m^{2c+1}\,. && (1.86)
\end{aligned}$$

Since we had considered an arbitrary vertex $v \in [n]$, taking the maximum over all $v \in [n]$ in Eq. (1.86) yields

$$q(\mathcal{Z}) = \max_{v\in[n]} \sum_{F_1\in\mathcal{Z}} \sum_{F_2\in\mathcal{Z}} r_v(F_1,F_2) \le |\mathcal{Z}|\cdot 6m^{2c+1}\,. \tag{1.87}$$

Lower bounding $M(\mathcal{Z})$.

Each function $F_1 \in \mathcal{Z}$ is good by choice of $\mathcal{Z}$. Additionally, since $c \ge 1$, Lemma 25 yields

$$\sum_{F_2\in\mathcal{X}} r(F_1,F_2) \ge \frac{1}{2\mathrm{e}}\cdot(c+1)\cdot m^{2c+1} \qquad \forall F_1\in\mathcal{Z}\,. \tag{1.88}$$

Using Eq. (1.88) and recalling the definition of $M(\mathcal{Z})$ from Eq. (1.79), we get

$$\begin{aligned}
M(\mathcal{Z}) &= \sum_{F_1\in\mathcal{Z}} \sum_{F_2\in\mathcal{X}} r(F_1,F_2)\\
&\ge \frac{|\mathcal{Z}|}{2\mathrm{e}}\cdot(c+1)\cdot m^{2c+1}\,. && (1.89)
\end{aligned}$$

Combining the bounds.

Combining the bounds from Eq. (1.87) and Eq. (1.89), we can now estimate the bound from Theorem 1.3.1:

$$\min_{\substack{\mathcal{Z}\subseteq\mathcal{X}:\\ q(\mathcal{Z})>0}} \frac{M(\mathcal{Z})}{q(\mathcal{Z})} \ge \frac{\frac{|\mathcal{Z}|}{2\mathrm{e}}\cdot(c+1)m^{2c+1}}{|\mathcal{Z}|\cdot 6m^{2c+1}} \qquad \text{(By Eq. (1.87) and Eq. (1.89))}$$

$$\geq \frac{\sqrt{m}}{24\mathrm{e}} \qquad \text{(Since } c+1 \geq \sqrt{m}/2\text{)}$$
$$\geq \frac{1}{96} \cdot \left(\frac{s}{\Delta}\right)^{1/4} \qquad \text{(By Lemma 18)}$$

Therefore, the randomized query complexity of local search is

$$\Omega\left(\min_{\substack{\mathcal{Z}\subseteq\mathcal{X}:\\ q(\mathcal{Z})>0}} \frac{M(\mathcal{Z})}{q(\mathcal{Z})}\right) \subseteq \Omega\left(\left(\frac{s}{\Delta}\right)^{1/4}\right) .$$

This completes the proof of the theorem. □

1.12.3 Helper lemmas

Here we prove the helper lemmas that are used in the proof of Theorem 1.3.3. All the lemmas assume the setup of the parameters from Theorem 1.3.3.

Lemma 19. *For each $\mathbf{x} \in \{1\} \times [m]^{2c}$, the function $f_{\mathbf{x}}$ is valid for the staircase $S_{\mathbf{x}}$ induced by $\mathbf{x}$ and the cluster paths.*

Proof. Let $S_{\mathbf{x}} = (w_1, \ldots, w_s)$ be the vertices of the staircase $S_{\mathbf{x}}$ induced by $\mathbf{x}$ and $\mathcal{P}$. We show that all the three conditions required by the definition of a valid function (Definition 1.10.1) hold.

To show the first condition of validity, consider two vertices $v_1, v_2 \in S_{\mathbf{x}}$. Let i_1 be the maximum index such that v_1 is the i_1-th vertex in $S_{\mathbf{x}}$. Let i_2 be defined similarly for v_2. By Definition 1.11.3, we have $f_{\mathbf{x}}(v_1) = -\mathrm{i}_1 < 0$ and $f_{\mathbf{x}}(v_1) = -\mathrm{i}_2 < 0$. Furthermore, if $\mathrm{i}_1 < \mathrm{i}_2$, then $f_{\mathbf{x}}(v_1) > f_{\mathbf{x}}(v_2)$. Therefore the first condition of validity is satisfied.

Also by Definition 1.11.3, of the function $f_{\mathbf{x}}$, we have that:

- By definition, $S_{\mathbf{x}}$ starts at $v_{start} \in N_1$
- $f_{\mathbf{x}}(v) = dist(v, v_{start}) > 0$ for all $v \notin S_{\mathbf{x}}$, so the second condition of validity is satisfied.

- $f_{\mathbf{x}}(v) \leq 0$ for all $v \in S_{\mathbf{x}}$, so the third condition of validity is satisfied.

Therefore $f_{\mathbf{x}}$ is valid for the staircase $S_{\mathbf{x}}$ induced by $\mathbf{x}$ and $\mathcal{P}$. □

Lemma 20. *If $n \geq 9$, then the next two properties hold:*

- *Let $F_1, F_2 \in \mathcal{X}$. Then $r(F_1, F_2) = 0$ when $\mathcal{H}(F_1) = \mathcal{H}(F_2)$.*

- *There exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ such that*

$$q(\mathcal{Z}) = \max_{v \in [n]} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v) \neq F_2(v)\}} > 0 \ .$$

Proof. We first show that $r(F_1, F_2) = 0$ when $\mathcal{H}(F_1) = \mathcal{H}(F_2)$. To see this, suppose $\mathcal{H}(F_1) = \mathcal{H}(F_2)$ for some functions $F_1, F_2 \in \mathcal{X}$. Then by definition of the set of functions $\mathcal{X}$, there exist sequences of vertices $\mathbf{x}, \mathbf{y} \in \{1\} \times [m]^{2c}$ and bits $b_1, b_2 \in \{0, 1\}$ such that $F_1 = g_{\mathbf{x},b_1}$ and $F_2 = g_{\mathbf{y},b_2}$. By definition of $\mathcal{H}$, we have $\mathcal{H}(g_{\mathbf{x},b_1}) = b_1$ and $\mathcal{H}(g_{\mathbf{y},b_2}) = b_2$. Since $\mathcal{H}(F_1) = \mathcal{H}(F_2)$, we have $b_1 = b_2$. Then $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = 0$ by definition of r, or equivalently, $r(F_1, F_2) = 0$.

Next we show there is a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$. To see this, consider two disjoint sets of vertices $U_1, U_2 \subset [m]$ such that $U_1 = \{u_2^1, \ldots, u_{2c+1}^1\}$, $U_2 = \{u_2^2, \ldots, u_{2c+1}^2\}$, each vertex $u_{\mathrm{j}}^{\mathrm{i}}$ appears exactly once in U_{i}, and $u_{\mathrm{j}}^{\mathrm{i}} \neq 1$ for all i, j. Such sets U_1, U_2 exist since $m \leq n$ and

$$|U_1| + |U_2| + |\{1\}| = 4c + 1 = 4(\lfloor \sqrt{m} \rfloor - 1) + 1 \leq 4\sqrt{n} - 3 \leq n \qquad \text{for } n \geq 9.$$

Form the sequences of vertices $\mathbf{x} = (1, u_2^1, \ldots, u_{2c+1}^1)$ and $\mathbf{y} = (1, u_2^2, \ldots, u_{2c+1}^2)$. Then both $\mathbf{x}$ and $\mathbf{y}$ are good. Consider now the functions $g_{\mathbf{x},0}$ and $g_{\mathbf{y},1}$. By definition of r, we have $r(g_{\mathbf{x},0}, g_{\mathbf{y},1}) = n$, since the maximum index j for which $\mathbf{x}_{1 \to \mathrm{j}} = \mathbf{y}_{1 \to \mathrm{j}}$ is $\mathrm{j} = 1$. Let v be the last vertex of $S_{\mathbf{x}}$. Then

$$q(\{g_{\mathbf{x},0}, g_{\mathbf{y},1}\}) \geq r(g_{\mathbf{x},0}, g_{\mathbf{y},1}) \cdot \mathbb{1}_{\{g_{\mathbf{x},0}(u_{2c+1}^1) \neq g_{\mathbf{y},1}(u_{2c+1}^1)\}} = n > 0 \ .$$

Thus there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$ as required. □

Lemma 21. *Let* $\mathbf{x}, \mathbf{y} \in \{1\} \times [m]^{2c}$, $b_1, b_2 \in \{0,1\}$, $v \in [n]$. *Let* $\mathrm{j} \in [2c+1]$ *be the maximum odd index for which* $\mathbf{x}_{1\to \mathrm{j}} = \mathbf{y}_{1\to \mathrm{j}}$. *Then if* $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$, *then at least one of the next two properties holds:*

(i) $v \in Tail(\mathrm{j}, S_{\mathbf{y}})$.

(ii) $\mathbf{x} = \mathbf{y}$.

Proof. We start with a few observations.

Recall from Definition 1.12.9 that $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_1}) = r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot \mathbb{1}_{\{\mu(S_{\mathbf{x}},v) \le \mu(S_{\mathbf{y}},v)\}}$. By the lemma statement, we have $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$, and so the next two inequalities hold:

$$r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0 \tag{1.90}$$

$$\mu(S_{\mathbf{x}}, v) \le \mu(S_{\mathbf{y}}, v)\,. \tag{1.91}$$

Also recall by definition of r_v that $r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot \mathbb{1}_{\{g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)\}}$. Then Eq. (1.90) implies that

$$g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)\,. \tag{1.92}$$

To prove that $v \in Tail(\mathrm{j}, S_{\mathbf{y}})$ or $\mathbf{x} = \mathbf{y}$ we consider two cases:

Case 1: $v \in Tail(\mathrm{j}, S_{\mathbf{x}})$**.**

Decomposing the staircase $S_{\mathbf{x}}$ into the initial segment $S_{\mathbf{x}_{1\to \mathrm{j}}}$ and the remainder $Tail(\mathrm{j}, S_{\mathbf{x}})$, and similarly the staircase $S_{\mathbf{y}}$ into initial segment $S_{\mathbf{y}_{1\to \mathrm{j}}}$ and the remainder $Tail(\mathrm{j}, S_{\mathbf{y}})$, we get:

$$\begin{aligned} & \mu(S_{\mathbf{x}}, v) \le \mu(S_{\mathbf{y}}, v) && \text{(By Eq. (1.91))} \\ \iff & \mu(S_{\mathbf{x}_{1\to \mathrm{j}}}, v) + \mu(Tail(\mathrm{j}, S_{\mathbf{x}}), v) \le \mu(S_{\mathbf{y}_{1\to \mathrm{j}}}, v) + \mu(Tail(\mathrm{j}, S_{\mathbf{y}}), v) \end{aligned}$$

$$\iff \mu(Tail(\mathrm{j}, S_{\mathbf{x}}), v) \leq \mu(Tail(\mathrm{j}, S_{\mathbf{y}}), v) \qquad \text{(Since } \mathbf{x}_{1\to \mathrm{j}} = \mathbf{y}_{1\to \mathrm{j}}.\text{)}$$

But since $v \in Tail(\mathrm{j}, S_{\mathbf{x}})$, we have $\mu(Tail(\mathrm{j}, S_{\mathbf{x}}), v) \geq 1$, so

$$1 \leq \mu(Tail(\mathrm{j}, S_{\mathbf{y}}), v)$$

Thus $v \in Tail(\mathrm{j}, S_{\mathbf{y}})$, so property (i) from the lemma statement holds. This completes Case 1.

Case 2: $v \notin Tail(\mathrm{j}, S_{\mathbf{x}})$**.**

If $\mathbf{x} = \mathbf{y}$, then property (ii) from the lemma statement holds.

Now suppose $\mathbf{x} \neq \mathbf{y}$. Then $Tail(\mathrm{j}, S_{\mathbf{y}}) \neq \emptyset$. We claim $v \in S_{\mathbf{x}} \cup S_{\mathbf{y}}$.

Suppose towards a contradiction that $v \notin S_{\mathbf{x}} \cup S_{\mathbf{y}}$. For each $b \in \{0, 1\}$, $u \in [n]$, and sequence $\mathbf{z} = (1, z_2, z_3 \ldots, z_{2c+1}) \in \{1\} \times [m]^{2c}$, we have by Eq. (1.45) (which defines the function $g_{\mathbf{z},b}$) that

$$g_{\mathbf{z},b}(u) = \begin{cases} (f_{\mathbf{z}}(u), b) & u \text{ is the last vertex of } S_{\mathbf{z}} \\ (f_{\mathbf{z}}(u), -1) & \text{otherwise} \end{cases} \tag{1.93}$$

Since $v \notin Tail(\mathrm{j}, S_{\mathbf{x}})$, v is not the last vertex of $S_{\mathbf{x}}$, and so $g_{\mathbf{x},b_1}(v) = (f_{\mathbf{x}}(v), -1)$. Moreover, since $x_1 = y_1 = 1$ and $v \notin S_{\mathbf{x}} \cup S_{\mathbf{y}}$, we have $f_{\mathbf{x}}(v) = dist(v, x_1) = dist(v, y_1) = f_{\mathbf{y}}(v)$. Combining these observations yields $g_{\mathbf{x},b_1}(v) = (f_{\mathbf{x}}(v), -1) = (f_{\mathbf{y}}(v), -1) = g_{\mathbf{y},b_2}(v)$, which contradicts Eq. (1.92). Thus the assumption must have been false and $v \in S_{\mathbf{x}} \cup S_{\mathbf{y}}$.

To summarize, we have $\mathbf{x}_{1\to \mathrm{j}} = \mathbf{y}_{1\to \mathrm{j}}$, $\mathbf{x} \neq \mathbf{y}$, $v \in S_{\mathbf{x}} \cup S_{\mathbf{y}}$, and $v \notin Tail(\mathrm{j}, S_{\mathbf{x}})$. Suppose towards a contradiction that $v \notin Tail(\mathrm{j}, S_{\mathbf{y}})$. Then

$$\begin{aligned} g_{\mathbf{x},b_1}(v) &= (f_{\mathbf{x}}(v), -1) && \text{(Since } v \neq x_{L+1}\text{, as } v \notin Tail(\mathrm{j}, S_{\mathbf{x}}).\text{)} \\ &= (f_{\mathbf{y}}(v), -1) && \text{(Since } v \notin Tail(\mathrm{j}, S_{\mathbf{x}}) \text{ and } v \notin Tail(\mathrm{j}, S_{\mathbf{y}}).\text{)} \end{aligned}$$

$$= g_{\mathbf{y},b_2}(v), \qquad \text{(Since } v \neq y_{L+1}\text{, as } v \notin Tail(\mathrm{j}, S_{\mathbf{y}})\text{.)}$$

which contradicts Eq. (1.92). Thus the assumption must have been false and $v \in Tail(\mathrm{j}, S_{\mathbf{y}})$, so property (i) from the lemma statement holds.

We conclude that at least one of properties (i) and (ii) holds. This completes Case 2, as well as the proof of the lemma. □

Lemma 22. *For each* $\mathbf{x} \in \{1\} \times [m]^{2c}$*,* $b_1 \in \{0,1\}$*, odd* $\mathrm{j} \in [2c+1]$*, and* $v \in [n]$ *such that* $v \notin N_{x_{\mathrm{j}}}$*, we have*

$$\sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c},\ b_2\in\{0,1\}:\\ \max\{\mathrm{i}:\mathrm{i}\ \textit{is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \leq \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}:\\ \max\{\mathrm{i}:\mathrm{i}\ \textit{is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ v\in Tail(\mathrm{j},S_{\mathbf{y}})}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) .$$

Proof. Let $\mathbf{x} \in \{1\} \times [m]^{2c}$, $b_1 \in \{0,1\}$, odd $\mathrm{j} \in [2c+1]$, and $v \in [n]$ such that $v \notin N_{x_{\mathrm{j}}}$. Using the definitions of r, r_v, and $\widetilde{r}_v$, we have the following chain of identities:

$$\sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c},\ b_2\in\{0,1\}:\\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c},\ b_2\in\{0,1\}:\\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2})>0}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2})$$

(Since $\widetilde{r}_v$ is non-negative.)

$$= \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}:\\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \qquad \text{(Since } \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_1}) = 0\text{.)}$$

$$= \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}:\\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \mu(S_{\mathbf{x}},v)\leq\mu(S_{\mathbf{y}},v)\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} r_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \qquad \text{(By definition of } \widetilde{r}_v\text{.)}$$

$$= \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}:\\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd},\mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}\\ \mu(S_{\mathbf{x}},v)\leq\mu(S_{\mathbf{y}},v)\\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \cdot \mathbb{1}_{\{g_{\mathbf{x},b_1}(v)\neq g_{\mathbf{y},1-b_1}(v)\}}$$

(By definition of r_v.)

$$= \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}: \\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd}, \mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j} \\ \mu(S_{\mathbf{x}},v)\leq\mu(S_{\mathbf{y}},v) \\ g_{\mathbf{x},b_1}(v)\neq g_{\mathbf{y},1-b_1}(v) \\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \tag{1.94}$$

Consider an arbitrary $\mathbf{y} \in \{1\} \times [m]^{2c}$ meeting the properties from the last sum of Eq. (1.94):

- $\max\{\mathrm{i} : \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} = \mathrm{j}$
- $\mu(S_{\mathbf{x}}, v) \leq \mu(S_{\mathbf{y}}, v)$
- $g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},1-b_1}(v)$
- $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) > 0$

By Lemma 21, the inequality $\widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) > 0$ implies that at least one of $v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$ or $\mathbf{x} = \mathbf{y}$ holds.

However, $\mathbf{x}$ cannot be equal to such $\mathbf{y}$. To see this, suppose for sake of contradiction that $\mathbf{x} = \mathbf{y}$. Then since $g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},1-b_1}(v)$ we would have that v must be the last vertex of $S_{\mathbf{y}}$. Therefore $v \in N_{y_{2c+1}}$. Also, we would have $\mathrm{j} = 2c + 1$ since $\mathbf{x} = \mathbf{y}$. But then $v \in N_{\mathrm{j}}$, which is a contradiction with the assumption $v \notin N_{\mathrm{j}}$. Therefore $\mathbf{x} \neq \mathbf{y}$, and so $v \in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})$.

We can continue to bound the sum from Eq. (1.94) as follows:

$$\sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}: \\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd}, \mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j} \\ \mu(S_{\mathbf{x}},v)\leq\mu(S_{\mathbf{y}},v) \\ g_{\mathbf{x},b_1}(v)\neq g_{\mathbf{y},1-b_1}(v) \\ \widetilde{r}_v(g_{\mathbf{x},b_1},g_{\mathbf{y},1-b_1})>0}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \leq \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}: \\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd}, \mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j} \\ v\in \mathit{Tail}(\mathrm{j},S_{\mathbf{y}})}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1})\,. \tag{1.95}$$

Combining Eq. (1.94) and Eq. (1.95), we get the inequality required by the lemma statement:

$$\sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c},\ b_2\in\{0,1\}: \\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd}, \mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j}}} \widetilde{r}_v(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \leq \sum_{\substack{\mathbf{y}\in\{1\}\times[m]^{2c}: \\ \max\{\mathrm{i}:\mathrm{i}\text{ is odd}, \mathbf{x}_{1\to\mathrm{i}}=\mathbf{y}_{1\to\mathrm{i}}\}=\mathrm{j} \\ v\in \mathit{Tail}(\mathrm{j},S_{\mathbf{y}})}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1})\,.$$

$\square$

Lemma 23. *Let $\mathbf{x} \in \{1\} \times [m]^{2c}$, odd $\mathrm{j} \in [2c+1]$, and $v \in [n]$ such that $v \notin N_{x_\mathrm{i}}$. Then*

$$\left|\left\{\mathbf{y} \in \{1\} \times [m]^{2c} \mid \max\{\mathrm{i} : \mathrm{i} \textit{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} = \mathrm{j} \textit{ and } v \in \mathit{Tail}(\mathrm{j}, S_\mathbf{y})\right\}\right| \leq (2c+1) \cdot m^{2c-\mathrm{j}} .$$

Proof. We will now do case analysis on different conditions on i and j.

(i) For odd $\mathrm{i} \in [2c]$ with $\mathrm{i} \geq \mathrm{j}$, consider the number of $\mathbf{y} \in \{1\} \times [m]^{2c}$ such that

$$\mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}} \quad \text{and} \quad v \in S_{\mathbf{y},\mathrm{i}} . \tag{1.96}$$

Since i is odd, $S_{\mathbf{y},\mathrm{i}}$ is a path within a cluster, and thus can only contain v if N_{y_i} is the one cluster containing v. If $\mathrm{i} = \mathrm{j}$, then we know by assumption that N_{y_i} does *not* contain v. Otherwise, there is only one choice for y_i while there are m choices for each of the rest of $y_{\mathrm{j}+1}, y_{\mathrm{j}+2}, \ldots, y_{2c+1}$. Thus, there are at most $m^{2c-\mathrm{j}}$ choices of $\mathbf{y}$ for which Eq. (1.96) holds.

(ii) For even $\mathrm{i} \in [2c]$ with $\mathrm{i} \geq \mathrm{j}$, consider the number of $\mathbf{y} \in \{1\} \times [m]^{2c}$ such that

$$\mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}, v \in S_{\mathbf{y},\mathrm{i}}, \text{ and } v \notin S_{\mathbf{y},k} \text{ for all odd } k \geq \mathrm{j} \text{ and } v \text{ is not the last vertex of } S_\mathbf{y} . \tag{1.97}$$

Since i is even, $S_{\mathbf{y},\mathrm{i}}$ is an inter-cluster path. Recall the m inter-cluster paths between any two clusters are disjoint except for their start and endpoints. Meanwhile v cannot be their start or endpoints since otherwise $v \in S_{\mathbf{y},k}$ for an odd $k \geq \mathrm{j}$ or v would be the last vertex of $S_\mathbf{y}$. Therefore, $S_{\mathbf{y},\mathrm{i}}$ can only contain v for at most one value of y_i while there are m choices for each of the rest of $y_{\mathrm{j}+1}, y_{\mathrm{j}+2}, \ldots, y_{2c+1}$. Thus, there are at most $m^{2c-\mathrm{j}}$ choices of $\mathbf{y}$ for which Eq. (1.97) holds.

(iii) Finally, consider the number of $\mathbf{y} \in \{1\} \times [m]^{2c}$ such that

$$\mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}} \quad \text{and} \quad v \text{ is the last vertex of } S_\mathbf{y} . \tag{1.98}$$

This cannot occur unless $v \in N_{y_{2c+1}}$. Therefore, there are at most $m^{2c-\mathrm{j}}$ such $\mathbf{y}$.

For v to be in $Tail(\mathrm{j}, S_{\mathbf{y}})$, we must have $\mathbf{y}$ satisfying one of Eq. (1.96), Eq. (1.97), or Eq. (1.98) for some value of i. Summing over choices of i gives us that

$$\left|\left\{\mathbf{y} \in \{1\} \times [m]^{2c} \mid \mathrm{j} = \max\{\mathrm{i} \ : \ \mathrm{i} \text{ is odd}, \ \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} \text{ and } v \in Tail(\mathrm{j}, S_{\mathbf{y}})\right\}\right| \\ \leq (2c+1) \cdot m^{2c-\mathrm{j}} \,.$$

This completes the proof of the lemma. □

Lemma 24. *Let* $\mathrm{j} \in [2c]$ *and* $\mathbf{x} \in \{1\} \times [m]^{2c}$ *be an arbitrary good sequence of indices. Then*

$$\left|\mathbf{y} \in \{1\} \times [m]^{2c} \mid \mathbf{y} \textit{ is good and } \mathrm{j} = \max\{\mathrm{i} : \mathrm{i} \textit{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}\right| = (m-\mathrm{j}-1) \cdot \prod_{\mathrm{i}=\mathrm{j}+2}^{2c+1} (m-\mathrm{i}+1) \,.$$

Proof. Let $\mathbf{y} = (y_1, \ldots, y_{2c+1}) \in [m]^{2c+1}$ be such that $x_1 = y_1 = 1$, $\mathrm{j} = \max\{\mathrm{i} : \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}$, and $\mathbf{y}$ is good.. Recall that for $\mathbf{y}$ to be good, each index in $(y_1, \ldots, y_{2c+1})$ has to be distinct.

We will count by choosing each index of $\mathbf{y}$ in order from y_1 to y_{2c+1}. There is only one choice for each of $y_1, \ldots, y_{\mathrm{j}}$ since we need $\mathbf{x}_{1\to\mathrm{j}} = \mathbf{y}_{1\to\mathrm{j}}$.

Consider the number of possible indices for $y_{\mathrm{j}+1}$:

- j indices already appeared in $\{y_1, \ldots, y_{\mathrm{j}}\}$ so they cannot be reused, otherwise $\mathbf{y}$ becomes bad.
- $\mathrm{j} = \max\{\mathrm{i} : \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}$ means $x_{\mathrm{j}+1} \neq y_{\mathrm{j}+1}$, so $x_{\mathrm{j}+1}$ cannot be used either.

Thus, there are $m - \mathrm{j} - 1$ choices of indices for $y_{\mathrm{j}+1}$.

For all $\mathrm{i} > \mathrm{j} + 1$, there are $m - \mathrm{i} + 1$ choices of indices for y_{i} since $\mathrm{i} - 1$ options have already been used. Thus the final count is $(m - \mathrm{j} - 1) \cdot \prod_{\mathrm{i}=\mathrm{j}+2}^{2c+1}(m - \mathrm{i} + 1)$, as required. □

Lemma 25. *If $F_1 \in \mathcal{X}$ is good and $c \geq 1$, then* $\sum_{F_2 \in \mathcal{X}} r(F_1, F_2) \geq \frac{1}{2\mathrm{e}} \cdot (c+1) \cdot m^{2c+1}$.

Proof. Since $F_1 \in \mathcal{X}$, there exists a sequence $\mathbf{x} \in \{1\} \times [m]^{2c}$ and a bit $b_1 \in \{0,1\}$ such that $F_1 = g_{\mathbf{x},b_1}$. Then we have the following chain of identities:

$$\begin{aligned}
\sum_{F_2 \in \mathcal{X}} r(F_1, F_2) &= \sum_{F_2 \in \mathcal{X}} r(g_{\mathbf{x},b_1}, F_2) \\
&= \sum_{\mathbf{y} \in \{1\} \times [m]^{2c},\, b_2 \in \{0,1\}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) && \text{(By definition of the set } \mathcal{X}) \\
&= \sum_{\mathbf{y} \in \{1\} \times [m]^{2c}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) && \text{(Since } r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_1}) = 0) \\
&= \sum_{\mathrm{j}=1}^{2c+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c} \\ \mathrm{j} = \max\{\mathrm{i} \,:\, \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \,. && (1.99)
\end{aligned}$$

Since $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) = 0$ if $\mathbf{y}$ is bad, we have

$$\sum_{\mathrm{j}=1}^{2c+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c}: \\ \mathrm{j} = \max\{\mathrm{i} \,:\, \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) = \sum_{\mathrm{j}=1}^{2c+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c}: \\ \mathrm{j} = \max\{\mathrm{i} \,:\, \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} \\ \mathbf{y} \text{ is good}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \,. \tag{1.100}$$

Combining Eq. (1.99) and Eq. (1.100), the last sum in Eq. (1.100) can be written as:

$$\sum_{F_2 \in \mathcal{X}} r(g_{\mathbf{x},b_1}, F_2) = \sum_{\mathrm{j}=1}^{2c+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c}: \\ \mathrm{j} = \max\{\mathrm{i} \,:\, \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} \\ \mathbf{y} \text{ is good}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},1-b_1}) \,. \tag{1.101}$$

Substituting the definition of r in Eq. (1.101), we get

$$\sum_{F_2 \in \mathcal{X}} r(g_{\mathbf{x},b_1}, F_2) = \sum_{\mathrm{j}=1}^{2c+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c}: \\ \mathrm{j} = \max\{\mathrm{i} \,:\, \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to\mathrm{i}} = \mathbf{y}_{1\to\mathrm{i}}\} \\ \mathbf{y} \text{ is good}}} m^{\mathrm{j}} \,. \tag{1.102}$$

From here, we only need to count the number of such $\mathbf{y}$. There is exactly one good sequence $\mathbf{y} \in \{1\} \times [m]^{2c}$ such that $\mathbf{x}_{1\to 2c+1} = \mathbf{y}_{1\to 2c+1}$, namely $\mathbf{y} = \mathbf{x}$. Therefore,

$$\begin{aligned}
\sum_{F_2 \in \mathcal{X}} r(g_{\mathbf{x},b_1}, F_2) &= \sum_{\mathrm{j}=1}^{2c+1} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c}: \\ \max\{\mathrm{i} : \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to \mathrm{i}} = \mathbf{y}_{1\to \mathrm{i}}\} = \mathrm{j} \\ \mathbf{y} \text{ is good}}} m^{\mathrm{j}} && \text{(By Eq. (1.102))} \\
&= m^{2c+1} + \sum_{\mathrm{j}=1}^{2c} \sum_{\substack{\mathbf{y} \in \{1\} \times [m]^{2c}: \\ \max\{\mathrm{i} : \mathrm{i} \text{ is odd}, \mathbf{x}_{1\to \mathrm{i}} = \mathbf{y}_{1\to \mathrm{i}}\} = \mathrm{j} \\ \mathbf{y} \text{ is good}}} m^{\mathrm{j}} \\
&= m^{2c+1} + \sum_{\mathrm{j}=1}^{2c} m^{\mathrm{j}} \cdot \mathbb{1}_{\{\mathrm{j} \text{ is odd}\}} \cdot (m - \mathrm{j} - 1) \cdot \prod_{\mathrm{i}=\mathrm{j}+2}^{2c+1} (m - \mathrm{i} + 1) && \text{(By Lemma 24)} \\
&\geq (c+1) \cdot m^{2c+1} \cdot \left(1 - \frac{2c+1}{m}\right)^{2c+1} \\
&\geq (c+1) \cdot m^{2c+1} \cdot \left(1 - \frac{1}{2c+1}\right)^{2c+1} && \text{(Since } 2c+1 \leq \sqrt{m}.\text{)} \\
&\geq \frac{1}{2\mathrm{e}} \cdot (c+1) \cdot m^{2c+1} && \text{(Since } 2c+1 \geq 3 \text{, which is more than sufficient.)}
\end{aligned}$$

Since $F_1 = g_{\mathbf{x},b_1}$, this is the required inequality, which completes the proof. □

2. SPECTRAL LOWER BOUNDS FOR LOCAL SEARCH

This chapter is based on my paper of the same name, which can be found at https://arxiv.org/abs/2403.06248.

2.1 Introduction

Local search stands as a robust heuristic within optimization and computer science, analyzed through both white box and black box frameworks. In the black box model, we are given a graph $G = (V, E)$ alongside oracle access to a function $f : V \to R$. The objective is to identify a vertex v that represents a local minimum, meaning $f(v) \leq f(u)$ for every edge (u, v), while minimizing the number of vertices queried.

Obtaining lower bounds for the complexity of local search has a rich history of analysis via random walks. The first pioneering work on the subject was [9], which did careful tailored analysis of the hitting time of random walks on the Boolean hypercube to obtain lower bounds for local search. Another breakthrough was obtained by [3], which designed a combinatorial method of obtaining lower bounds for local search inspired by the relational adversary method from quantum computing. This approach enabled obtaining sharper lower bounds for the Boolean hypercube and d-dimensional grid, and was successfully used in many later works.

In this paper we consider the high level question: How does the geometry of the graph affect the complexity of local search? While the query complexity is comprehensively understood for neighbourhood structures such as the d-dimensional grid and the Boolean hypercube, knowledge remains limited for more general neighbourhood structures.

Nevertheless, the spatial structure in optimization settings typically extends to more complex graphs. For instance, in scenarios such as low rank matrix estimation with data compromised by adversarial attacks, the function is defined on a Riemannian manifold rather than a traditional Euclidean space [11]; thus the discretization of an optimization search space may not necessarily always correspond to some d-dimensional grid. For a more extensive survey on stochastic gradient descent on Riemannian manifolds, see, e.g., [12]. This motivates studying local search not only on hypercubes and grids, but also on broader classes of graphs.

Inspired by the observation that many lower bounds for local search are based on various types of random walks, we consider general random walks for the graph at hand and obtain lower bounds as a function of their mixing time. Our analysis uses a variant of the classical relational adversary method from [5] and our main result is generic in two ways: the graph is arbitrary and the random walk evolves according to a Markov chain, which we only require to be lazy, irreducible, and reversible. This allows us to formally connect the query complexity of local search and the mixing time of the fastest mixing Markov chain for the given graph, which is a classical problem analyzed starting with [50], with recent results in [51]. As a corollary, we also get a lower bound in terms of the spectral gap of the transition matrix of the chain.

2.2 Model

Let $G = (V, E)$ be a connected undirected graph and $f : V \to \mathbb{R}$ a function defined on the vertices. A vertex $v \in V$ is a local minimum if $f(v) \leq f(u)$ for all $\{u, v\} \in E$. We will write $V = [n] = \{1, \ldots, n\}$.

Given as input a graph G and oracle access to function f, the local search problem is to find a local minimum of f on G using as few queries as possible. Each query is of the form: "Given a vertex v, what is $f(v)$?".

Query complexity.

The *deterministic query complexity* of a task is the total number of queries necessary and sufficient for a correct deterministic algorithm to find a solution. The *randomized query complexity* is the expected number of queries required to find a solution with probability at least $9/10$ for each input, where the expectation is taken over the coin tosses of the protocol.

Degree and distance.

Let d_{max} and d_{min} be the maximum and minimum degree of any vertex in G respectively. Let $d(v)$ be the degree of v for all $v \in V$. For each $u, v \in V$, let $dist(u, v)$ be the length of the shortest path from u to v.

Markov chain.

We consider a discrete-time Markov chain on G with transition matrix $\mathcal{P}$, meaning that the state space is V and $\mathcal{P}_{u,v} = \mathcal{P}_{v,u} = 0$ whenever $(u, v) \notin E(G) \bigcup_{u \in V} \{\{u, u\}\}$. Suppose the chain has stationary distribution $\boldsymbol{\pi}$. The chain is:

- *lazy*: if $\mathcal{P}_{u,u} \geq 1/2$ for all $u \in V$.
- *irreducible*: if all states can be reached from any starting point. Formally, for any two states $x, y \in V$ there exists an integer t (possibly depending on x and y) such that $(\mathcal{P}^t)_{x,y} > 0$, where $\mathcal{P}^t$ is the t-th power of the matrix $\mathcal{P}$.
- *reversible*: if $\boldsymbol{\pi}(u)\mathcal{P}_{u,v} = \boldsymbol{\pi}(v)\mathcal{P}_{v,u}$ for all $u, v \in V$. [1]

For each $\epsilon > 0$, the *mixing time* $t_{mix}(\epsilon)$ of the Markov chain[2] with transition matrix $\mathcal{P}$ is:

$$t_{mix}(\epsilon) = \min\left\{ t \in \mathbb{N} \;\middle|\; \forall u \in V : \;\; \frac{1}{2} \sum_{v \in V} \left| (\mathcal{P}^t)_{u,v} - \boldsymbol{\pi}(v) \right| \leq \epsilon \right\} . \tag{2.1}$$

2.3 Our results

We get the following result in terms of the mixing time of the Markov chain used.

[1]↑For a formal definition of reversibility, see [52] equation 3.26.

[2]↑For a formal definition of mixing time, see e.g. [52] equation 4.30. The definition in [52] equation 4.30 is based on the TV distance, but is equivalent to the one here by Proposition 4.2 in [52].

Theorem 2.3.1. *Let $G = (V, E)$ be a connected undirected graph on n vertices. Consider a discrete-time, lazy, irreducible, and reversible Markov chain on G with transition matrix $\mathcal{P}$ and stationary distribution π. Then the randomized query complexity of local search on G is*

$$\Omega\left(\frac{\sqrt{n}}{t_{mix}\left(\frac{\sigma}{2n}\right)\cdot\exp(3\sigma)}\right), \qquad \textit{where } \sigma = \max_{u,v\in V}\frac{\pi(v)}{\pi(u)}.$$

The best lower bound given by Theorem 2.3.1 is attained by considering the Markov chain with the fastest possible mixing time for G; see [50] for a classical reference on this problem.

Indeed, for many classes of graphs there can significant gaps between the mixing time of the fastest mixing Markov chain and that of more obvious choices of Markov chains (such as the max-degree walk or the Metropolis-Hastings chain), including the barbell graph, edge-transitive graphs, and distance transitive graphs [53]. For example, when the stationary distribution is set to uniform on the barbell graph[3], the max-degree random walk has mixing time $\Theta(n^3)$ while the fastest mixing walk mixes in only $\Theta(n^2)$ steps (see corollaries 5.2 and 5.3 in [54]).

Remark 1. *Since $\sigma \geq 1$, we always have $t_{mix}(\sigma/(2n)) \leq t_{mix}(1/(2n))$. Thus Theorem 2.3.1 implies the randomized query complexity is*

$$\Omega\left(\frac{\sqrt{n}}{t_{mix}\left(\frac{1}{2n}\right)\exp(3\sigma)}\right).$$

For lazy chains, the second eigenvalue is always non-negative, so Theorem 2.3.1 implies a lower bound based on the spectral gap. This leads to the following corollary:

[3]↑The barbell graph consists of two cliques of $n/2$ vertices each, connected by a single edge.

Corollary 5. *Let $G = (V, E)$ be a connected undirected graph on n vertices. Consider a discrete-time, lazy, irreducible, and reversible Markov chain on G with transition matrix $\mathcal{P}$ and stationary distribution $\boldsymbol{\pi}$. The randomized query complexity of local search on G is*

$$\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log(n)\exp(3\sigma)}\right),$$

where λ_2 is the second eigenvalue of $\mathcal{P}$ and $\sigma = \max_{u,v \in V} \boldsymbol{\pi}(v)/\boldsymbol{\pi}(u)$.

If we constrain the ratio d_{max}/d_{min} and focus on the simple lazy random walk, we can remove the dependency on σ in Corollary 5.

Corollary 6. *Let $G = (V, E)$ be a connected undirected graph on n vertices. If $d_{max}/d_{min} \le C$ for some constant $C > 0$, then the randomized query complexity of local search on G is*

$$\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log n}\right),$$

where λ_2 is the second eigenvalue of the transition matrix of the simple lazy random walk on G.

The lower bound in Corollary 6 improves by a $\log n$ factor the lower bound attainable from [5] for such graphs. For comparison, we state the lower bound from [5] next.

Proposition 2.3.1. *[5] Let $G = (V, E)$ be a connected undirected graph on n vertices. If $d_{max}/d_{min} \le C$ for some constant $C > 0$, then the randomized query complexity of local search on G is*

$$\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log^2(n)}\right),$$

where λ_2 is the second eigenvalue of the transition matrix of the simple lazy random walk on G.

The extra factor of $\log(n)$ in Proposition 2.3.1 stems from the result used to connect expansion to edge congestion (see [27], corollary C.2), which rounds a fractional flow with congestion $n\log(n)$ to an integer flow with congestion $n\log^2(n)$. By avoiding expansion altogether, Corollary 6 also avoids this excess factor of $\log(n)$.

High Level Approach

The high level approach is as follows. We consider a set of value functions $f : V \to \mathbb{R}$ induced by walks from a fixed starting vertex. We define f such that the value at any vertex off the walk is the distance to the starting vertex of the walk, and the value at vertices on the walk is decreasing along the walk. This ensures that there is only one local minimum, namely the end of the walk. This type of construction is classical [3, 9]. We denote the space of such functions $\mathcal{X}$.

We then choose a similarity measure $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ (also called the "relation") between any two functions. Semantically, the relation measures the difficulty of distinguishing two possible input functions from each other; thus it will be useful that functions induced by walks with a long shared prefix are defined as more related. Given r, we invoke the relational adversary variant (Lemma 26 from [5]), which implicitly defines a distribution over inputs based on r, and outputs a lower bound on the randomized query complexity. We choose r carefully such that the distribution over walks is that of an arbitrary Markov chain.

The innovation of our methodology over [5] lies in the construction; where their construction is based on low-congestion paths, ours takes the more natural and general approach of using arbitrary Markov chains, including the usual lazy random walks.

A key step in analyzing the formula given by the relational adversary is the following. When fixing a staircase $\mathbf{x}$ and sampling a second staircase $\mathbf{y}$ conditioned to share an initial prefix of random length with $\mathbf{x}$, the proof has to show that no vertex v is too likely to lie on the "tail" of $\mathbf{y}$ (i.e. the portion after the initial segment shared with $\mathbf{x}$). The difference between our setting and previous random walk based methods lies in this analysis.

We analyze separately the portions of the tail before and after it mixes. After $\mathbf{y}$ mixes, it is close to being distributed according to the stationary distribution $\boldsymbol{\pi}$. Before $\mathbf{y}$ mixes, the

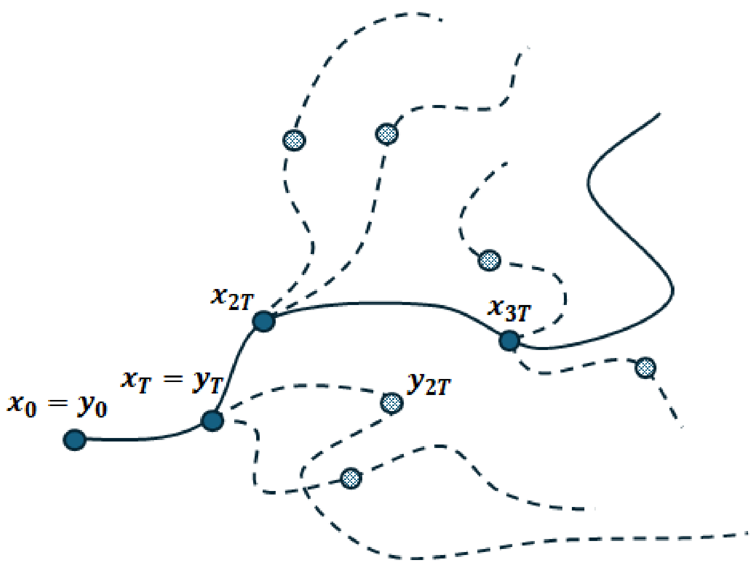


Figure 2.1. Consider a graph G with a lazy, irreducible, and reversible Markov chain $\mathcal{P}$ with stationary distribution $\boldsymbol{\pi}$ and mixing time T. The proof fixes a walk $\mathbf{x} = [x_0, \ldots, x_L]$, where $L = \lfloor \sqrt{n} \rfloor \cdot T$. The walk $\mathbf{x}$ is illustrated as a solid line, where every T-th node is highlighted. Sample a random walk $\mathbf{y} = [y_0, \ldots, y_L]$ according to $\mathcal{P}$, conditioned on $\mathbf{y}$ and $\mathbf{x}$ having a shared prefix of length $\mathrm{j}T$, where $\mathrm{j} \sim U(0, \lfloor \sqrt{n} \rfloor)$. We say that $x_{\mathrm{j}T} = y_{\mathrm{j}T}$ is the "divergence point". In the figure, the shared prefix of $\mathbf{x}$ and $\mathbf{y}$ is $[x_0, \ldots, x_T]$, so $\mathrm{j} = 1$. A critical step of the proof is to show that no vertex v is too likely to lie on $\mathbf{y}$ after its divergence from $\mathbf{x}$. To show this, we divide the walk $\mathbf{y}$ in two regions. Vertices in the region $R_1 = [y_{\mathrm{j}T}, \ldots, y_{\mathrm{j}T+1}]$ are *collectively* close to being distributed according to $\boldsymbol{\pi}$. This is because the divergence point from $\mathbf{x}$ is chosen randomly. In the region $R_2 = [y_{\mathrm{j}T+1}, ..., y_L]$, the walk $\mathbf{y}$ has mixed, so the vertices in R_2 are close to being distributed according to $\boldsymbol{\pi}$. In either case, no vertex v is too likely to lie on $\mathbf{y}$ after diverging from $\mathbf{x}$.

randomness of the point on $\mathbf{x}$ at which $\mathbf{y}$ diverges suffices to keep $\mathbf{y}$ close enough to being distributed according to $\boldsymbol{\pi}$. A visual depiction is shown in Figure 2.1.

This analysis is very generic, parameterized only by the stationary distribution of the walk used. This allows Theorem 2.3.1 to give results for walks from arbitrary Markov chains with no additional analysis other than estimating the mixing time. The natural mixing properties of the lazy random walk on expanders then allow us to derive strong lower bounds for such graphs.

2.4 Lower bound for local search via mixing times

2.4.1 Preliminaries

We fix a discrete-time Markov chain with transition matrix $\mathcal{P}$ that has the properties required by Theorem 2.3.1: lazy, irreducible, and reversible. Let $\boldsymbol{\pi}$ denote the unique stationary distribution of the chain. For each $S \subseteq V$, let $\boldsymbol{\pi}(S) = \sum_{v \in S} \boldsymbol{\pi}(v)$. Moreover, let

$$\sigma = \max_{u,v \in V} \boldsymbol{\pi}(v)/\boldsymbol{\pi}(u)\,. \tag{2.2}$$

For every $k \in \mathbb{N}$ and every walk $\mathbf{x} = (x_0, x_1, \ldots, x_k)$ in G, let $\mathcal{P}[\mathbf{x}]$ be the probability that the random walk started at x_0 with transition matrix $\mathcal{P}$ has trajectory $\mathbf{x}$, that is:

$$\mathcal{P}[\mathbf{x}] = \prod_{\mathrm{i}=0}^{k-1} \mathcal{P}_{x_\mathrm{i}, x_{\mathrm{i}+1}}\,. \tag{2.3}$$

Bottleneck Ratio.

The bottleneck ratio $\Phi_\star$ of the Markov chain with transition matrix $\mathcal{P}$ is [4]:

$$\Phi_\star = \min_{S \subseteq V : \boldsymbol{\pi}(S) \leq \frac{1}{2}} \sum_{u \in S, v \in V \setminus S} \frac{\boldsymbol{\pi}(u)\mathcal{P}_{u,v}}{\boldsymbol{\pi}(S)}.$$

Visiting probability.

For each pair of vertices $u, v \in V$ and integer $\ell \in \mathbb{N}$:

- let $\mathrm{P}_{visit}(u, v, \ell)$ be the probability that a random walk that has transition matrix $\mathcal{P}$, length ℓ, and starts at u visits v.
- let $\mathrm{E}_{visit}(u, v, \ell)$ be the expected number of times that a random walk that has transition matrix $\mathcal{P}$, length ℓ, and starts at u visits v.

[4]↑See, e.g., [52] equation 7.5.

- let $\mathrm{P}_{end}(u, v, \ell)$ be the probability that a random walk that has transition matrix $\mathcal{P}$, length ℓ, and starts at u ends at v.

Edge Expansion.

Let $E(S, V \setminus S) = \{(u, v) \in E \mid u \in S, v \in V \setminus S\}$ be the set of edges with one endpoint in S and the other in $V \setminus S$. The edge expansion of G is

$$\beta = \min_{S \subseteq V : |S| \leq n/2} \frac{|E(S, V \setminus S)|}{|S|} .$$

One of the main ingredients in our proof is a variant of (classical) the relational adversary method from quantum computing given in [5].

Lemma 26 ([5], Theorem 3)**.** *Consider finite sets A and B, a set $\mathcal{X} \subseteq B^A$ of functions, and a map $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ which assigns a label to each function in $\mathcal{X}$. Additionally, we get oracle access to an unknown function $F^* \in \mathcal{X}$. The problem is to compute $\mathcal{H}(F^*)$ using as few queries to F^* as possible.*[5]

Let $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a non-zero symmetric function of our choice with $r(F_1, F_2) = 0$ whenever $\mathcal{H}(F_1) = \mathcal{H}(F_2)$. For each $\mathcal{Z} \subseteq \mathcal{X}$, define

$$M(\mathcal{Z}) = \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{X}} r(F_1, F_2) \qquad \text{and} \qquad q(\mathcal{Z}) = \max_{a \in A} \sum_{F_1 \in \mathcal{Z}} \sum_{F_2 \in \mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(a) \neq F_2(a)\}} . \quad (2.4)$$

If there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$, then the randomized query complexity of the problem is at least

$$\min_{\mathcal{Z} \subseteq \mathcal{X} : q(\mathcal{Z}) > 0} 0.01 \cdot M(\mathcal{Z}) / q(\mathcal{Z}) . \quad (2.5)$$

To get lower bounds for local search, we will analyze the performance of deterministic algorithms when the input distribution is obtained by considering random functions, each of

[5]↑In other words, we have free access to $\mathcal{H}$ and the only queries counted are the ones to F^*, which will be of the form: "What is $F^*(a)$?", for some $a \in A$. The oracle will return $F^*(a)$ in one computational step.

which is defined using a classical "staircase" construction. Each staircase is a random walk with transition matrix $\mathcal{P}$ on G. We first give the setup and then prove the main theorem.

2.4.2 Setup

Definition 2.4.1 (Set of walks $\mathcal{W}$ and parameter T). *Let $L = \lfloor\sqrt{n}\rfloor \cdot T$, where $T = t_{mix}(\frac{\sigma}{2n})$. Let $\mathcal{W}$ be the set of walks $\{\mathbf{w} \mid \mathbf{w} = (w_0, \ldots, w_L)\}$ in G with w_0 equal to the vertex 1 and with $\mathcal{P}_{w_{\mathrm{i}}, w_{\mathrm{i}+1}} > 0$ for all $0 \leq \mathrm{i} < L$.*

Definition 2.4.2 (Milestones). *Given a walk $\mathbf{x} = (x_0, x_1, \ldots, x_L) \in \mathcal{W}$, every T-th vertex of the walk (including the first vertex) is called a "milestone". E.g., the first three milestones of $\mathbf{x}$ are x_0, x_T, and x_{2T}.*

Definition 2.4.3 (Good/bad walk). *A walk $\mathbf{x} \in \mathcal{W}$ is "good" if it does not repeat any milestones and "bad" otherwise. Let $good(\mathbf{x}) = True$ if $\mathbf{x}$ is good and $False$ otherwise.*

Definition 2.4.4 (Heads and Tails.). *For every walk $\mathbf{x} = (x_0, x_1, \ldots, x_L) \in \mathcal{W}$, let*

$$\begin{cases} Head(\mathbf{x}, \mathrm{j}) = (x_0, x_1, \ldots, x_{\mathrm{j}\cdot T}) & \forall \mathrm{j} \in \{0, \ldots, \lfloor\sqrt{n}\rfloor\}. \\ Tail(\mathbf{x}, \mathrm{j}) = (x_{\mathrm{j}\cdot T+1}, x_{\mathrm{j}\cdot T+2}, \ldots, x_L) & \forall \mathrm{j} \in \{0, \ldots, \lfloor\sqrt{n}\rfloor\}. \\ Tail(\mathbf{x}, \mathrm{j}_1, \mathrm{j}_2) = (x_{\mathrm{j}_1\cdot T+1}, x_{\mathrm{j}_1\cdot T+2}, \ldots, x_{\mathrm{j}_2\cdot T}) & \forall \mathrm{j}_1, \mathrm{j}_2 \in \{0, \ldots, \lfloor\sqrt{n}\rfloor\} \text{ with } \mathrm{j}_1 \leq \mathrm{j}_2. \end{cases} \tag{2.6}$$

For all $\mathbf{x}, \mathbf{y} \in \mathcal{W}$, define $J(\mathbf{x}, \mathbf{y})$ as the maximum index j with $Head(\mathbf{x}, \mathrm{j}) = Head(\mathbf{y}, \mathrm{j})$.

Definition 2.4.5 (The functions $f_{\mathbf{x}}$ and $g_{\mathbf{x},b}$; the set $\mathcal{X}$). *For each walk $\mathbf{x} = (x_0, \ldots, x_L) \in \mathcal{W}$, define a function $f_{\mathbf{x}} : [n] \to \{-L, -L+1, \ldots, n\}$ such that for all $v \in [n]$:*

$$f_{\mathbf{x}}(v) = \begin{cases} dist(v, 1) & \text{if } v \notin \mathbf{x} \\ -\max\Big\{\mathrm{i} \in \{0, \ldots, L\} \mid x_{\mathrm{i}} = v\Big\} & \text{if } v \in \mathbf{x}. \end{cases} \tag{2.7}$$

For all $b \in \{0,1\}$, define $g_{\mathbf{x},b} : [n] \to \{-L, -L+1, \dots, n\} \times \{-1,0,1\}$ so that for all $v \in [n]$:

$$g_{\mathbf{x},b}(v) = \begin{cases} (f_{\mathbf{x}}(v), -1) & \text{if } v \neq x_L \\ (f_{\mathbf{x}}(v), b) & \text{if } v = x_L \,. \end{cases} \tag{2.8}$$

Let $\mathcal{X} = \left\{ g_{\mathbf{x},b} \mid \mathbf{x} \in \mathcal{W} \text{ and } b \in \{0,1\} \right\}$.

Definition 2.4.6 (Valid function)**.** *Let $\mathbf{x} = (x_0, \dots, x_\ell)$ be a walk in G. A function $f : V \to \mathbb{R}$ is valid with respect to the walk $\mathbf{x}$ if it satisfies the next conditions:*

1. *For all $u, v \in \mathbf{x}$, if $\max\{\mathrm{i} \in \{0, \dots, \ell\} \mid v = x_\mathrm{i}\} < \max\{\mathrm{i} \in \{0, \dots, \ell\} \mid u = x_\mathrm{i}\}$, then $f(v) > f(u)$. In other words, as one walks along the walk $\mathbf{x}$ starting from x_0 until x_ℓ, if the last time the vertex v appears is before the last time that vertex u appears, then $f(v) > f(u)$.*
2. *For all $v \in V \setminus \mathbf{x}$, we have $f(v) = dist(x_0, v) > 0$.*
3. *$f(x_\mathrm{i}) \leq 0$ for all $\mathrm{i} \in \{0, \dots, \ell\}$.*

We define a similarity measure r between functions from $\mathcal{X}$, commonly referred to as the *relation*.

Definition 2.4.7 (The function r)**.** *Let $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a symmetric function such that for each $\mathbf{x}, \mathbf{y} \in \mathcal{W}$ and $b_1, b_2 \in \{0,1\}$,*

$$r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = \begin{cases} 0 & \text{If } b_1 = b_2 \text{ or } \mathbf{x} = \mathbf{y} \text{ or } \mathbf{x} \text{ is bad or } \mathbf{y} \text{ is bad.} \\ \frac{\mathcal{P}[\mathbf{x}]\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y},\mathrm{j})]} & \text{Otherwise, where } \mathrm{j} = J(\mathbf{x}, \mathbf{y}) \,. \end{cases}$$

Observation 2.4.1. *The function r from Definition 2.4.7 is symmetric, since by definition of j we have $Head(\mathbf{x}, \mathrm{j}) = Head(\mathbf{y}, \mathrm{j})$.*

Having defined the relation r, applying Lemma 26 will give a lower bound by considering a distribution p over $\mathcal{X}$, where each function $F \in \mathcal{X}$ is given as input with probability $p(F) = \frac{M(\{F\})}{M(\mathcal{X})}$, where M is as defined in Eq. (2.4) for the relation r.

What remains to be done is to explain how to invoke Lemma 26 and estimate the lower bound it gives when the input distribution is p.

2.4.3 Proof of the main theorem

Theorem 2.3.1. *Let $G = (V, E)$ be a connected undirected graph on n vertices. Consider a discrete-time, lazy, irreducible, and reversible Markov chain on G with transition matrix $\mathcal{P}$ and stationary distribution π. Then the randomized query complexity of local search on G is*

$$\Omega\left(\frac{\sqrt{n}}{t_{mix}\left(\frac{\sigma}{2n}\right)\cdot \exp(3\sigma)}\right), \qquad \textit{where } \sigma = \max_{u,v\in V} \frac{\pi(v)}{\pi(u)}.$$

Proof. First, we may assume that $n \geq 16\sigma^2$. This is because in the alternative case where $n < 16\sigma^2$, the theorem statement does not give anything useful as $\sqrt{n} < \exp(3\sigma)$ and $t_{mix}(\sigma/(2n)) \geq 1$.

The value functions we will use are of the form $f_{\mathbf{x}}$ as seen in Definition 2.4.5. These functions are parametrized by walks $\mathbf{x}$ of length L from the set $\mathcal{W}$ defined in Definition 2.4.1. For sake of invoking Lemma 26 however, we must turn the local search problem into a decision problem. To do this, we use the technique shown in [3, 16]: associate with each function $f_{\mathbf{x}}$ the function $g_{\mathbf{x},b}$ defined in Definition 2.4.5. This hides a bit at the local minimum vertex (while hiding the value -1 at every other vertex). The decision problem is: given the graph G and oracle access to a function $g_{\mathbf{x},b}$, return the hidden bit b; i.e. we set the function $\mathcal{H}$ for use in Lemma 26 as $\mathcal{H}(g_{\mathbf{x},b}) = b$.

By Lemma 27, the function $f_{\mathbf{x}}$ as defined in Definition 2.4.5 is valid. Therefore by Lemma 35, $f_{\mathbf{x}}$ has a unique local minimum, namely x_L. This means that $g_{\mathbf{x},b}$ as defined in Definition 2.4.5 does indeed hide the bit b only at the local minimum of $f_{\mathbf{x}}$. Therefore measuring the query complexity of this decision problem will give the answer for local search, as the following two problems have query complexity within additive 1:

- *search problem*: given oracle access to a function f_x, find a vertex v that is a local minimum;
- *decision problem*: given oracle access to a function $g_{x,b}$, find b.

We then invoke Lemma 26 with $\mathcal{X}$ as defined in Definition 2.4.5 and with $\mathcal{H}(g_{\mathbf{x},b}) = b$. This tells us that the randomized query complexity of the decision problem is at least $\min_{\mathcal{Z}\subseteq\mathcal{X}:q(\mathcal{Z})>0} 0.01 \cdot M(\mathcal{Z})/q(\mathcal{Z})$, with $M(\mathcal{Z})$ and $q(\mathcal{Z})$ as defined in Lemma 26.

By Lemma 32, there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$. From this point on, we fix an arbitrary subset $\mathcal{Z} \subseteq \mathcal{X}$ of functions with $q(\mathcal{Z}) > 0$ and will then lower bound $M(\mathcal{Z})$ and upper bound $q(\mathcal{Z})$.

Lower bounding $M(\mathcal{Z})$.

Because $q(\mathcal{Z}) > 0$, we know $\mathcal{Z}$ is not empty. Consider an arbitrary function $g_{\mathbf{x},b_1} \in \mathcal{Z}$ with $\mathbf{x}$ good. By definition of M, we have

$$M(\{g_{\mathbf{x},b_1}\}) = \sum_{g_{\mathbf{y},b_2}\in\mathcal{X}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}), \tag{2.9}$$

recalling:

- $\mathcal{P}[\mathbf{y}] = \prod_{\mathrm{i}=0}^{k-1} \mathcal{P}_{y_\mathrm{i},y_{\mathrm{i}+1}}$ for every walk $\mathbf{y} = (y_0, y_1, \ldots, y_k)$ in G;
- Definition 2.4.4 of $Head(\mathbf{y}, \mathrm{j})$ and $J(\mathbf{x}, \mathbf{y})$ and Definition 2.4.7 of the relation r:

$$r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = \begin{cases} 0 & \text{If } b_1 = b_2 \text{ or } \mathbf{x} = \mathbf{y} \text{ or } \mathbf{x} \text{ is bad or } \mathbf{y} \text{ is bad.} \\ \frac{\mathcal{P}[\mathbf{x}]\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y},\mathrm{j})]} & \text{Otherwise, where } \mathrm{j} = J(\mathbf{x}, \mathbf{y})\,. \end{cases}$$

Then we can rewrite $M(g_{\mathbf{x},b_1})$ as

$$M(\{g_{\mathbf{x},b_1}\}) = \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ b_2=1-b_1\\ good(\mathbf{y})\\ \mathbf{x}\neq\mathbf{y}}} \frac{\mathcal{P}[\mathbf{x}]\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y}, J(\mathbf{x},\mathbf{y}))]}\,. \tag{2.10}$$

The condition $\mathbf{x}\neq\mathbf{y}$ in Eq. (2.10) is equivalent to $J(\mathbf{x},\mathbf{y})\neq\lfloor\sqrt{n}\rfloor$, so it must be the case that $J(\mathbf{x},\mathbf{y})\in\{0,\ldots,\lfloor\sqrt{n}\rfloor-1\}$. We decompose the summation in Eq. (2.10) by the value of $J(\mathbf{x},\mathbf{y})$, excluding cases where $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2})=0$, to get the following:

$$M(\{g_{\mathbf{x},b_1}\}) = \mathcal{P}[\mathbf{x}]\sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1}\sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ J(\mathbf{x},\mathbf{y})=\mathrm{j}\\ b_2=1-b_1\\ good(\mathbf{y})}} \frac{\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y},\mathrm{j})]}\,. \tag{2.11}$$

There is a one-to-one correspondence between walks $\mathbf{y}\in\mathcal{W}$ and functions $g_{\mathbf{y},b_2}$ with $b_2 = 1-b_1$. Therefore we may equivalently sum over $\mathbf{y}\in\mathcal{W}$ in Eq. (2.11):

$$M(\{g_{\mathbf{x},b_1}\}) = \mathcal{P}[\mathbf{x}]\sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1}\sum_{\substack{\mathbf{y}\in\mathcal{W}:\\ J(\mathbf{x},\mathbf{y})=\mathrm{j}\\ good(\mathbf{y})}} \frac{\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y},\mathrm{j})]}\,. \tag{2.12}$$

Since $\mathbf{y}$ is the concatenation of $Head(\mathbf{y},\mathrm{j})$ with $Tail(\mathbf{y},\mathrm{j})$, we have:

$$\mathcal{P}[\mathbf{y}] = \mathcal{P}[Head(\mathbf{y},\mathrm{j})]\cdot\mathcal{P}[Tail(\mathbf{y},\mathrm{j})]\cdot\mathcal{P}_{y_{\mathrm{j}\cdot T},\,y_{\mathrm{j}\cdot T+1}}\,. \tag{2.13}$$

Substituting Eq. (2.13) in Eq. (2.12) gives

$$M(\{g_{\mathbf{x},b_1}\}) = \mathcal{P}[\mathbf{x}]\sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1}\sum_{\substack{\mathbf{y}\in\mathcal{W}:\\ J(\mathbf{x},\mathbf{y})=\mathrm{j}\\ good(\mathbf{y})}} \mathcal{P}[Tail(\mathbf{y},\mathrm{j})]\cdot\mathcal{P}_{y_{\mathrm{j}\cdot T},\,y_{\mathrm{j}\cdot T+1}}\,. \tag{2.14}$$

If it weren't for the restrictions that $J(\mathbf{x},\mathbf{y}) = \mathrm{j}$ (instead of $J(\mathbf{x},\mathbf{y}) \geq \mathrm{j}$) and *good*$(y)$, then the inner sum would be over all possible walks from $x_{\mathrm{j}T}$, and would thus be 1. Because of those restrictions, we instead invoke Lemma 30 to continue from Eq. (2.14) as follows:

$$M(\{g_{\mathbf{x},b_1}\}) \geq \mathcal{P}[\mathbf{x}] \cdot 2^{-4\sigma} \cdot \lfloor \sqrt{n} \rfloor \,. \tag{2.15}$$

If $\mathbf{x}$ is bad, then $M(\{g_{\mathbf{x},b_1}\}) = 0$, and so the function $g_{\mathbf{x},b_1}$ does not contribute to $M(\mathcal{Z})$. Therefore

$$M(\mathcal{Z}) \geq 2^{-4\sigma} \lfloor \sqrt{n} \rfloor \sum_{\substack{g_{\mathbf{x},b_1} \in \mathcal{Z} \\ good(\mathbf{x})}} \mathcal{P}[\mathbf{x}] \,. \tag{2.16}$$

Upper bounding $q(\mathcal{Z})$.

Define

$$\widetilde{q}(\mathcal{Z}, v) = \sum_{g_{\mathbf{x},b_1} \in \mathcal{Z}} \sum_{g_{\mathbf{y},b_2} \in \mathcal{Z}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \mathbb{1}_{\{g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)\}} \,. \tag{2.17}$$

By definition of $q(\mathcal{Z})$, we have

$$q(\mathcal{Z}) = \max_{v \in V} \sum_{g_{\mathbf{x},b_1} \in \mathcal{Z}} \sum_{g_{\mathbf{y},b_2} \in \mathcal{Z}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \mathbb{1}_{\{g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)\}} = \max_{v \in V} \widetilde{q}(\mathcal{Z}, v) \,. \tag{2.18}$$

We will bound $q(\mathcal{Z})$ by bounding $\widetilde{q}(\mathcal{Z}, v)$ for an arbitrary v. Fix an arbitrary vertex v. We first partition the inner sum according to $J(\mathbf{x},\mathbf{y})$:

$$\begin{aligned} \widetilde{q}(\mathcal{Z}, v) &= \sum_{g_{\mathbf{x},b_1} \in \mathcal{Z}} \sum_{g_{\mathbf{y},b_2} \in \mathcal{Z}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \mathbb{1}_{\{g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)\}} && \text{(By definition of } \widetilde{q}(\mathcal{Z}, v).) \\ &= \sum_{g_{\mathbf{x},b_1} \in \mathcal{Z}} \sum_{\mathrm{j}=0}^{\lfloor \sqrt{n} \rfloor - 1} \sum_{\substack{g_{\mathbf{y},b_2} \in \mathcal{Z}: \\ J(\mathbf{x},\mathbf{y}) = \mathrm{j}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \mathbb{1}_{\{g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)\}} \,. && (2.19) \end{aligned}$$

By Lemma 31, we may continue from Eq. (2.19) and expand to get

$$\tilde{q}(\mathcal{Z},v) \leq 2 \sum_{g_{\mathbf{x},b_1}\in\mathcal{Z}} \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{Z}: \\ v\in Tail(\mathbf{y},\mathrm{j}) \\ J(\mathbf{x},\mathbf{y})=\mathrm{j}}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \tag{2.20}$$

$$= 2 \sum_{\substack{g_{\mathbf{x},b_1}\in\mathcal{Z} \\ good(\mathbf{x})}} \mathcal{P}[\mathbf{x}] \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{Z}: \\ v\in Tail(\mathbf{y},\mathrm{j}) \\ J(\mathbf{x},\mathbf{y})=\mathrm{j} \\ b_2=1-b_1 \\ good(\mathbf{y})}} \frac{\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y},\mathrm{j})]} \qquad \text{(Using the definition of } r.\text{)}$$

$$\leq 2 \sum_{\substack{g_{\mathbf{x},b_1}\in\mathcal{Z} \\ good(\mathbf{x})}} \mathcal{P}[\mathbf{x}] \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}: \\ v\in Tail(\mathbf{y},\mathrm{j}) \\ J(\mathbf{x},\mathbf{y})=\mathrm{j} \\ b_2=1-b_1 \\ good(\mathbf{y})}} \frac{\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y},\mathrm{j})]}\,. \qquad \text{(Since } \mathcal{Z}\subseteq\mathcal{X}.\text{)} \tag{2.21}$$

Again, there is a one-to-one correspondence between walks $\mathbf{y} \in \mathcal{W}$ and functions $g_{\mathbf{y},b_2} \in \mathcal{X}$ with $b_2 = 1 - b_1$, so we may equivalently sum over $\mathbf{y} \in \mathcal{W}$ in Eq. (2.21). Additionally, we expand the scope from $J(\mathbf{x},\mathbf{y}) = \mathrm{j}$ to $J(\mathbf{x},\mathbf{y}) \geq \mathrm{j}$, which is equivalent to $Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{y},\mathrm{j})$. Finally, we drop the requirement that $good(\mathcal{Y})$. Continuing from Eq. (2.21),

$$\tilde{q}(\mathcal{Z},v) \leq 2 \sum_{\substack{g_{\mathbf{x},b_1}\in\mathcal{Z} \\ good(\mathbf{x})}} \mathcal{P}[\mathbf{x}] \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{\substack{\mathbf{y}\in\mathcal{W}: \\ v\in Tail(\mathbf{y},\mathrm{j}) \\ J(\mathbf{x},\mathbf{y})\geq\mathrm{j}}} \frac{\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y},\mathrm{j})]}\,. \tag{2.22}$$

From here we partition based on where in $Tail(\mathbf{y}, \mathrm{j})$ the vertex v lies: the first short part of the tail, i.e. $Tail(\mathbf{y}, \mathrm{j}, \mathrm{j}+1)$, or the rest of the tail, i.e. $Tail(\mathbf{y}, \mathrm{j}+1)$. Continuing from Eq. (2.22),

$$\widetilde{q}(\mathcal{Z}, v) \leq 2 \sum_{\substack{g_{\mathbf{x},b_1} \in \mathcal{Z} \\ good(\mathbf{x})}} \mathcal{P}[\mathbf{x}] \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor - 1} \left(\sum_{\substack{\mathbf{y} \in \mathcal{W}: \\ v \in Tail(\mathbf{y},\mathrm{j},\mathrm{j}+1) \\ J(\mathbf{x},\mathbf{y}) \geq \mathrm{j}}} \frac{\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y}, \mathrm{j})]} + \sum_{\substack{\mathbf{y} \in \mathcal{W}: \\ v \in Tail(\mathbf{y},\mathrm{j}+1) \\ J(\mathbf{x},\mathbf{y}) \geq \mathrm{j}}} \frac{\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y}, \mathrm{j})]} \right) \tag{2.23}$$

Since $\mathbf{y}$ is the concatenation of $Head(\mathbf{y}, \mathrm{j})$ with $Tail(\mathbf{y}, \mathrm{j})$, we have

$$\mathcal{P}[\mathbf{y}] = \mathcal{P}[Head(\mathbf{y}, \mathrm{j})] \cdot \mathcal{P}[Tail(\mathbf{y}, \mathrm{j})] \cdot \mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} \,. \tag{2.24}$$

Using Eq. (2.24) in Eq. (2.23), we obtain

$$\begin{aligned} \widetilde{q}(\mathcal{Z}, v) \leq 2 \sum_{\substack{g_{\mathbf{x},b_1} \in \mathcal{Z} \\ good(\mathbf{x})}} \mathcal{P}[\mathbf{x}] \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor - 1} \Bigg(& \sum_{\substack{\mathbf{y} \in \mathcal{W}: \\ v \in Tail(\mathbf{y},\mathrm{j},\mathrm{j}+1) \\ J(\mathbf{x},\mathbf{y}) \geq \mathrm{j}}} \mathcal{P}[Tail(\mathbf{y}, \mathrm{j})] \cdot \mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} \\ & + \sum_{\substack{\mathbf{y} \in \mathcal{W}: \\ v \in Tail(\mathbf{y},\mathrm{j}+1) \\ J(\mathbf{x},\mathbf{y}) \geq \mathrm{j}}} \mathcal{P}[Tail(\mathbf{y}, \mathrm{j})] \cdot \mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} \Bigg) \end{aligned} \tag{2.25}$$

To bound the first part, we will use Lemma 29. Since $\mathbf{x}$ is good, we have

$$\begin{aligned} \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor - 1} \sum_{\substack{\mathbf{y} \in \mathcal{W}: \\ v \in Tail(\mathbf{y},\mathrm{j},\mathrm{j}+1) \\ J(\mathbf{x},\mathbf{y}) \geq \mathrm{j}}} \mathcal{P}[Tail(\mathbf{y}, \mathrm{j})] \cdot \mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} &= \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor - 1} \mathrm{P}_{visit}(x_{\mathrm{j}T}, v, T) \\ & \qquad \text{(By definition of } \mathrm{P}_{visit}.\text{)} \\ &\leq T\sigma \,. \qquad \text{(By Lemma 29 since } \mathbf{x} \text{ is good.)} \end{aligned} \tag{2.26}$$

To bound the second part, we first partition according to the possible values of $\mathbf{y}_{(\mathrm{j}+1)T}$. We have

$$\sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{\substack{\mathbf{y}\in\mathcal{W}: \\ v\in Tail(\mathbf{y},\mathrm{j}+1) \\ J(\mathbf{x},\mathbf{y})\geq \mathrm{j}}} \mathcal{P}[Tail(\mathbf{y},\mathrm{j})]\cdot\mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} = \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{u\in V} \mathrm{P}_{end}(x_{\mathrm{j}T},u,T)\mathrm{P}_{visit}(u,v,L-(\mathrm{j}+1)T) \tag{2.27}$$

The visiting probability $\mathrm{P}_{visit}(u,v,\ell)$ is increasing in ℓ, so continuing from Eq. (2.27) we get

$$\begin{aligned} &\sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{\substack{\mathbf{y}\in\mathcal{W}: \\ v\in Tail(\mathbf{y},\mathrm{j}+1) \\ J(\mathbf{x},\mathbf{y})\geq \mathrm{j}}} \mathcal{P}[Tail(\mathbf{y},\mathrm{j})]\cdot\mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} \\ &\leq \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{u\in V} \mathrm{P}_{end}(x_{\mathrm{j}T},u,T)\mathrm{P}_{visit}(u,v,L) \qquad (2.28) \\ &\leq \sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{u\in V} \Big(\boldsymbol{\pi}(u) + |\mathrm{P}_{end}(x_{\mathrm{j}T},u,T)-\boldsymbol{\pi}(u)|\Big)\mathrm{P}_{visit}(u,v,L), \qquad (2.29) \end{aligned}$$

where in Eq. (2.29) we used the inequality $a\leq b+|a-b|$ for $a,b\geq 0$.

A random walk with starting vertex drawn from $\boldsymbol{\pi}$ has probability $\boldsymbol{\pi}(v)$ of being at v at each step. Formally, for all $\ell\in\mathbb{N}$ we have

$$\sum_{u\in V}\boldsymbol{\pi}(u)\mathrm{P}_{end}(u,v,\ell) = \boldsymbol{\pi}(v)\,. \tag{2.30}$$

Additionally, by union bound we have

$$\mathrm{P}_{visit}(u,v,L) \leq \sum_{\ell=1}^{L}\mathrm{P}_{end}(u,v,\ell)\,. \tag{2.31}$$

We have

$$\sum_{\mathrm{j}=0}^{\lfloor\sqrt{n}\rfloor-1} \sum_{u\in V} \left(\boldsymbol{\pi}(u) + |\mathrm{P}_{end}(x_{\mathrm{j}T},u,T)-\boldsymbol{\pi}(u)|\right)\mathrm{P}_{visit}(u,v,L) \tag{2.32}$$

$$\leq \sum_{j=0}^{\lfloor\sqrt{n}\rfloor-1}\left(\sum_{\ell=1}^{L}\sum_{u\in V}\pi(u)\mathrm{P}_{end}(u,v,\ell)+\sum_{u\in V}|\mathrm{P}_{end}(x_{jT},u,T)-\pi(u)|\cdot\mathrm{P}_{visit}(u,v,L)\right) \qquad \text{(By Eq. (2.31).)}$$

$$=\sum_{j=0}^{\lfloor\sqrt{n}\rfloor-1}\left(L\pi(v)+\sum_{u\in V}|\mathrm{P}_{end}(x_{jT},u,T)-\pi(u)|\cdot\mathrm{P}_{visit}(u,v,L)\right). \qquad \text{(By Eq. (2.30))} \tag{2.33}$$

Since $T = t_{mix}(\sigma/(2n))$, we have

$$\begin{aligned}\sum_{u\in V}|\mathrm{P}_{end}(x_{jT},u,T)-\pi(u)|\cdot\mathrm{P}_{visit}(u,v,L) &\leq \max_{u\in V}\mathrm{P}_{visit}(u,v,L)\sum_{u\in V}|\mathrm{P}_{end}(x_{jT},u,T)-\pi(u)|\\ &=\max_{u\in V}\mathrm{P}_{visit}(u,v,L)\sum_{u\in V}\left|(\mathcal{P}^T)_{x_{jT},u}-\pi(u)\right|\\ &\leq \max_{u\in V}\mathrm{P}_{visit}(u,v,L)\cdot\frac{\sigma}{n}.\end{aligned} \tag{2.34}$$

Combining Eq. (2.33) and Eq. (2.34) yields

$$\sum_{j=0}^{\lfloor\sqrt{n}\rfloor-1}\sum_{u\in V}\left(\pi(u)+|\mathrm{P}_{end}(x_{jT},u,T)-\pi(u)|\right)\mathrm{P}_{visit}(u,v,L)\leq\sum_{j=0}^{\lfloor\sqrt{n}\rfloor-1}\left(L\pi(v)+\frac{\sigma}{n}\max_{u\in V}\mathrm{P}_{visit}(u,v,L)\right) \tag{2.35}$$

$$\begin{aligned}&\leq \lfloor\sqrt{n}\rfloor(L+1)\frac{\sigma}{n} && \text{(Since } \mathrm{P}_{visit}(u,v,L)\leq 1 \text{ and } \pi(v)\leq\tfrac{\sigma}{n}.\text{)}\\ &\leq 2T\sigma\,. && \text{(Since } L=T\lfloor\sqrt{n}\rfloor \text{ and } 1\leq L.\text{)}\end{aligned} \tag{2.36}$$

Combining Eq. (2.29) and Eq. (2.36), we obtain

$$\sum_{j=0}^{\lfloor\sqrt{n}\rfloor-1}\sum_{\substack{\mathbf{y}\in\mathcal{W}:\\ v\in Tail(\mathbf{y},j+1)\\ J(\mathbf{x},\mathbf{y})\geq j}}\mathcal{P}[Tail(\mathbf{y},j)]\cdot\mathcal{P}_{y_{j\cdot T},\,y_{j\cdot T+1}}\leq 2T\sigma\,. \tag{2.37}$$

Combining Eq. (2.25), Eq. (2.26), and Eq. (2.37),

$$\tilde{q}(\mathcal{Z}, v) \leq 2 \sum_{\substack{g_{\mathbf{x},b_1} \in \mathcal{Z} \\ \mathit{good}(\mathbf{x})}} \mathcal{P}[\mathbf{x}] 3T\sigma = 6T\sigma \sum_{\substack{g_{\mathbf{x},b_1} \in \mathcal{Z} \\ \mathit{good}(\mathbf{x})}} \mathcal{P}[\mathbf{x}]\,. \tag{2.38}$$

Since $q(\mathcal{Z}) = \min_{v \in V} \tilde{q}(\mathcal{Z}, v)$ and Eq. (2.38) holds for an arbitrary choice of v, we get

$$q(\mathcal{Z}) \leq 6T\sigma \sum_{\substack{g_{\mathbf{x},b_1} \in \mathcal{Z} \\ \mathit{good}(\mathbf{x})}} \mathcal{P}[\mathbf{x}]\,. \tag{2.39}$$

Bounding $M(\mathcal{Z})/q(\mathcal{Z})$.

Combining Eq. (2.16) and Eq. (2.39), we obtain

$$\frac{M(\mathcal{Z})}{q(\mathcal{Z})} \geq \frac{2^{-4\sigma} \lfloor \sqrt{n} \rfloor \sum_{\substack{g_{\mathbf{x},b_1} \in \mathcal{Z} \\ \mathit{good}(\mathbf{x})}} \mathcal{P}[\mathbf{x}]}{6T\sigma \sum_{\substack{g_{\mathbf{x},b_1} \in \mathcal{Z} \\ \mathit{good}(\mathbf{x})}} \mathcal{P}[\mathbf{x}]} = \frac{2^{-4\sigma}}{6\sigma} \cdot \frac{\lfloor \sqrt{n} \rfloor}{T}\,. \tag{2.40}$$

We have $2^{-4\sigma}/(6\sigma) \in \Omega(1/exp(3\sigma))$. Thus applying Lemma 26 to the expression in Eq. (2.40), we obtain the required lower bound for local search on G, namely

$$\Omega\left(\frac{\sqrt{n}}{t_{mix}\left(\frac{\sigma}{2n}\right) \cdot \exp(3\sigma)}\right).$$

This completes the proof. $\square$

2.5 Helper lemmas

Lemma 27. *For each walk $\mathbf{x} = (x_0, \ldots, x_L) \in \mathcal{W}$, the function $f_{\mathbf{x}}$ given by Definition 2.4.5 is valid for the walk $\mathbf{x}$.*

Proof. We show the conditions required by the definition of a valid function (Definition 1.10.1) hold.

Condition 1. Consider two arbitrary vertices $v_1, v_2 \in \mathbf{x}$. Define

$$\mathrm{i}_1 = \max\Big\{k \in \{0,\ldots,L\} \mid v_1 = x_k\Big\} \qquad \text{and} \qquad \mathrm{i}_2 = \max\Big\{k \in \{0,\ldots,L\} \mid v_2 = x_k\Big\}.$$

Without loss of generality, we have $\mathrm{i}_1 \leq \mathrm{i}_2$. Condition 1 requires that if $\mathrm{i}_1 < \mathrm{i}_2$, then $f_{\mathbf{x}}(v_1) > f_{\mathbf{x}}(v_2)$. Since $v_1, v_2 \in \mathbf{x}$, Definition 2.4.5 of the function $f_{\mathbf{x}}$ states that $f_{\mathbf{x}}(v_1) = -\mathrm{i}_1$ and $f_{\mathbf{x}}(v_2) = -\mathrm{i}_2$. Thus if $\mathrm{i}_1 < \mathrm{i}_2$, then $f_{\mathbf{x}}(v_1) > f_{\mathbf{x}}(v_2)$, as required.

Condition 2. The second condition requires that $f_{\mathbf{x}}(v) = \mathit{dist}(x_0, v) > 0$ for all $v \in V \setminus \mathbf{x}$. Since $\mathbf{x} \in \mathcal{W}$, we have $x_0 = 1$. Using Definition 2.4.5 we have $f_{\mathbf{x}}(v) = \mathit{dist}(v, 1) = \mathit{dist}(v, x_0) > 0$ for all $v \notin \mathbf{x}$.

Condition 3. The third condition requires that $f_{\mathbf{x}}(x_{\mathrm{i}}) \leq 0$ for all $\mathrm{i} \in \{0,\ldots,L\}$. This condition is clearly met since $f_{\mathbf{x}}(v) \in \{0, -1, \ldots, -L\}$ for all $v \in \mathbf{x}$.

Therefore $f_{\mathbf{x}}$ is valid for the walk $\mathbf{x}$. □

Lemma 28. *Let $(w_0, w_1, \ldots)$ be a Markov chain generated by transition matrix $\mathcal{P}$ with arbitrary starting distribution. Then for all $v \in V$ we have*

$$\Pr\left[w_{t_{mix}(\sigma/(2n))} = v\right] \leq 2\sigma/n\,.$$

Proof. By definition of $t_{mix}(\sigma/(2n))$ we have

$$\sum_{v \in V} \Big|\Pr\left[w_{t_{mix}(\sigma/(2n))} = v\right] - \pi(v)\Big| \leq \sigma/n\,. \tag{2.41}$$

Since each term of the sum in Eq. (2.41) is non-negative, we have

$$\forall v \in V \ \Big|\Pr\left[w_{t_{mix}(\sigma/(2n))} = v\right] - \pi(v)\Big| \leq \sigma/n\,. \tag{2.42}$$

Removing the absolute value from Eq. (2.42) and rearranging, we get

$$\begin{aligned}\forall v \in V \;\; \Pr\left[w_{t_{mix}(\sigma/(2n))} = v\right] &\leq \sigma/n + \pi(v) \\ &\leq 2\sigma/n\,. && \text{(Since } \sigma \geq \pi(v)\cdot n.\text{)}\end{aligned}$$

This concludes the proof of the lemma. □

Lemma 29. *Let $S \subseteq V$ be a subset of vertices. Let $v \in V$ be a vertex and $\ell \in \mathbb{N}$. Then*

$$\sum_{u\in S} \mathrm{P}_{visit}(u, v, \ell) \leq \ell\sigma\,.$$

Proof. Let T_ℓ be the random variable representing the number of times a random walk of length ℓ starting at v visits a vertex in S. Decomposing by vertices in S, we have

$$\begin{aligned}\sum_{u\in S} \mathrm{P}_{visit}(u, v, \ell) &\leq \sum_{u\in S} \mathrm{E}_{visit}(u, v, \ell) && (2.43)\\ &= \sum_{u\in S} \mathrm{E}_{visit}(v, u, \ell)\frac{\pi(v)}{\pi(u)}\,. && \text{(By Lemma 34)}\end{aligned} \tag{2.44}$$

Using the definition of $\sigma = \max_{u,v\in V} \pi(v)/\pi(u)$, we have

$$\begin{aligned}\sum_{u\in S} \mathrm{E}_{visit}(v, u, \ell)\frac{\pi(v)}{\pi(u)} &\leq \sigma \sum_{u\in S} \mathrm{E}_{visit}(v, u, \ell) \\ &= \sigma \cdot \mathbb{E}\left[T_\ell\right] && \text{(By definition of } T_\ell.\text{)} \\ &\leq \ell\sigma\,. && \text{(Since } T_\ell \leq \ell.\text{)}\end{aligned} \tag{2.45}$$

Combining Eq. (2.44) and Eq. (2.45), we get $\sum_{u\in S} \mathrm{P}_{visit}(u, v, \ell) \leq \ell\sigma$, as required. □

Lemma 30. *Let $n \geq 16\sigma^2$. Fix a good walk $\mathbf{x} = (x_0, \ldots, x_L)$ with $x_0 = 1$ and $\mathcal{P}_{x_\mathrm{i}, x_{\mathrm{i}+1}} > 0$ for all $0 \leq \mathrm{i} < L$. Then for each $0 \leq \mathrm{j} < \lfloor\sqrt{n}\rfloor$,*

$$\sum_{\substack{\mathbf{y} \in \mathcal{W}: \\ J(\mathbf{x},\mathbf{y})=\mathrm{j} \\ good(\mathbf{y})}} \mathcal{P}[Tail(\mathbf{y}, \mathrm{j})] \cdot \mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} \geq 2^{-4\sigma} . \tag{2.46}$$

Proof. Let $P_{\mathcal{W}}$ be the distribution over the set of walks $\mathcal{W}$ generated by sampling a walk according to $\mathcal{P}$ starting at the vertex 1 with $L = \lfloor\sqrt{n}\rfloor \cdot T$ edges. Let $\mathbf{z}$ be a random walk drawn from $P_{\mathcal{W}}$. Recall that every T-th vertex of a walk is called a milestone, and a walk is good if it does not repeat milestones. We have

$$\sum_{\substack{\mathbf{y} \in \mathcal{W}: \\ J(\mathbf{x},\mathbf{y})=\mathrm{j} \\ good(\mathbf{y})}} \mathcal{P}[Tail(\mathbf{y}, \mathrm{j})] \cdot \mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} = \sum_{\substack{\mathbf{y} \in \mathcal{W}: \\ J(\mathbf{x},\mathbf{y})=\mathrm{j} \\ good(\mathbf{y})}} \mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} \cdot \mathcal{P}_{y_{\mathrm{j}\cdot T+1}, y_{\mathrm{j}\cdot T+2}} \cdot \ldots \cdot \mathcal{P}_{y_{L-1}, y_L} \tag{2.47}$$

$$= \Pr\Big[good(\mathbf{z}) \wedge J(\mathbf{x}, \mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x}, \mathrm{j}) = Head(\mathbf{z}, \mathrm{j})\Big] . \tag{2.48}$$

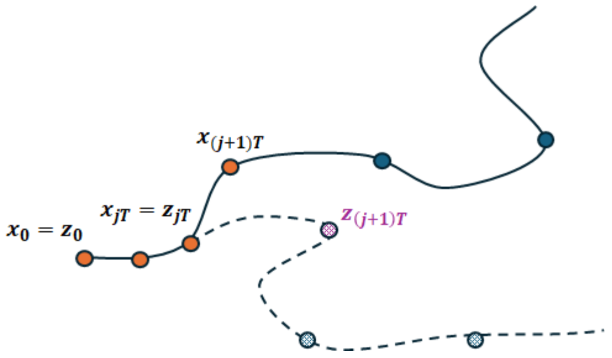


Figure 2.2. The milestone $z_{(\mathrm{j}+1)T}$, shown in purple, may not match any of the orange milestones: x_0 through $x_{\mathrm{j}T}$ because it would make $\mathbf{z}$ bad, and $x_{(\mathrm{j}+1)T}$ because it would make $J(\mathbf{x}, \mathbf{z}) > \mathrm{j}$.

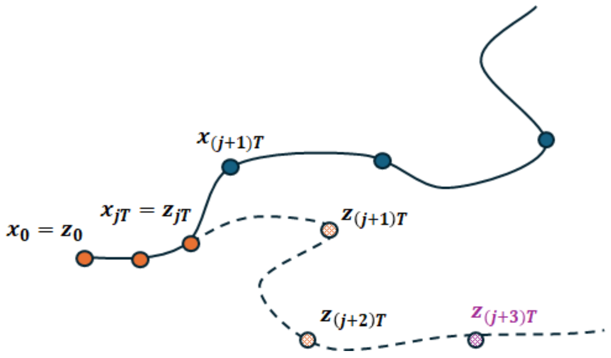


Figure 2.3. The milestone $z_{(\mathrm{j}+3)T}$, shown in purple, may not match any of the orange milestones because it would make $\mathbf{z}$ bad.

Equivalently, we can sample $\mathbf{z} \sim P_{\mathcal{W}}$ one segment at a time with the constraint that the initial $\mathrm{j}T+1$ vertices of $\mathbf{z}$ must match those of $\mathbf{x}$. That is, set $Head(\mathbf{z}, \mathrm{j}) = Head(\mathbf{x}, \mathrm{j})$. Then for

$\mathrm{i} = \mathrm{j}, \dots, \lfloor\sqrt{n}\rfloor$, sample the segment $Tail(\mathbf{z}, \mathrm{i}-1, \mathrm{i})$ conditioned on having set $Head(\mathbf{z}, \mathrm{i}-1)$.

For every $0 \le k \le \lfloor\sqrt{n}\rfloor$, the set of vertices given by the first $k+1$ milestones of $\mathbf{z}$ is

$$S_k = \{z_0, z_T, \dots, z_{kT}\}\,. \tag{2.49}$$

Given that $Head(\mathbf{x}, \mathrm{j}) = Head(\mathbf{z}, \mathrm{j})$, the condition $J(\mathbf{x}, \mathbf{z}) = \mathrm{j}$ is equivalent to

$$z_{(\mathrm{j}+1)T} \neq x_{(\mathrm{j}+1)T}\,. \tag{2.50}$$

Similarly, given that $Head(\mathbf{x}, \mathrm{j}) = Head(\mathbf{z}, \mathrm{j})$, the condition that $\mathbf{z}$ is good is equivalent to

$$z_{kT} \notin S_{k-1} \qquad \forall k \in \{\mathrm{j}+1, \mathrm{j}+2, \dots, \lfloor\sqrt{n}\rfloor\}\,. \tag{2.51}$$

For each $\mathrm{i} \in \{\mathrm{j}+1, \dots, \lfloor\sqrt{n}\rfloor$, define

$$Q_{\mathrm{i}} = \begin{cases} \{x_{(\mathrm{j}+1)T}\} \cup S_{\mathrm{j}} & \text{if } \ \mathrm{i} = \mathrm{j}+1 \\ S_{\mathrm{i}-1} & \text{if } \ \mathrm{j}+1 < \mathrm{i} \le \lfloor\sqrt{n}\rfloor\,. \end{cases} \tag{2.52}$$

Combining (2.50) and (2.51), we get

$$\begin{aligned} &\Pr\Big[good(\mathbf{z}) \wedge J(\mathbf{x}, \mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x}, \mathrm{j}) = Head(\mathbf{z}, \mathrm{j})\Big] \\ &\quad = \Pr\Big[\Big(z_{kT} \notin S_{k-1}\ \forall k \in \{\mathrm{j}+1, \dots, \lfloor\sqrt{n}\rfloor\}\Big) \wedge \Big(z_{(\mathrm{j}+1)T} \neq x_{(\mathrm{j}+1)T}\Big) \mid Head(\mathbf{x}, \mathrm{j}) = Head(\mathbf{z}, \mathrm{j})\Big] \end{aligned} \tag{2.53}$$

$$= \Pr\left[\bigwedge_{\mathrm{i}=\mathrm{j}+1}^{\lfloor\sqrt{n}\rfloor} z_{\mathrm{i}T} \notin Q_{\mathrm{i}} \mid Head(\mathbf{x}, \mathrm{j}) = Head(\mathbf{z}, \mathrm{j})\right] \qquad \text{(By definition of } Q_{\mathrm{i}} \text{ in (2.52))}$$

$$= \prod_{\mathrm{i}=\mathrm{j}+1}^{\lfloor\sqrt{n}\rfloor} \Pr\left[z_{\mathrm{i}T} \notin Q_{\mathrm{i}} \mid (Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{z},\mathrm{j})) \wedge \Big(\bigwedge_{k=\mathrm{j}+1}^{\mathrm{i}-1} z_{kT} \notin Q_k\Big)\right] \tag{2.54}$$

For all $\mathrm{i} \leq \lfloor\sqrt{n}\rfloor$, let $\mathcal{W}_{\mathrm{i}}$ be the space of walks of length $T\mathrm{i}$ that can occur with positive probability under transition matrix $\mathcal{P}$, formally defined as:

$$\mathcal{W}_{\mathrm{i}} = \left\{\mathbf{w} \mid \mathbf{w} = (w_0, \ldots, w_{T\cdot\mathrm{i}}) \text{ where } w_0 = 1 \text{ and } \mathcal{P}_{w_k,w_{k+1}} > 0 \text{ for all } 0 \leq k < T\cdot\mathrm{i}\right\}. \tag{2.55}$$

Then, using Eq. (2.53) gives

$$\Pr\Big[good(\mathbf{z}) \wedge J(\mathbf{x},\mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{z},\mathrm{j})\Big]$$
$$\geq \prod_{\mathrm{i}=\mathrm{j}+1}^{\lfloor\sqrt{n}\rfloor} \min_{\xi\in\mathcal{W}_{\mathrm{i}-1}} \Pr\Big[z_{\mathrm{i}T} \notin Q_{\mathrm{i}} \mid Head(\mathbf{z},\mathrm{i}-1) = \xi\Big]. \tag{2.56}$$

Since $T = t_{mix}(\sigma/(2n))$, Lemma 28 tells us that for all $v \in V$ and $1 \leq \mathrm{i} \leq \lfloor\sqrt{n}\rfloor$

$$\min_{\xi\in\mathcal{W}_{\mathrm{i}-1}} \Pr\Big[z_{\mathrm{i}T} = v \mid Head(\mathbf{z},\mathrm{i}-1) = \xi\Big] \leq \frac{2\sigma}{n}. \tag{2.57}$$

By the union bound applied to Eq. (2.57), for each set $R \subseteq V$ and $1 \leq \mathrm{i} \leq \lfloor\sqrt{n}\rfloor$, we have

$$\min_{\xi\in\mathcal{W}_{\mathrm{i}-1}} \Pr\Big[z_{\mathrm{i}T} \notin R \mid Head(\mathbf{z},\mathrm{i}-1) = \xi\Big] \geq 1 - \frac{2\sigma|R|}{n}. \tag{2.58}$$

We consider two cases based on j:

Case 1: $\mathrm{j} < \lfloor\sqrt{n}\rfloor - 1$**.**

Since $\mathrm{j} < \lfloor\sqrt{n}\rfloor - 1$, we have

$$\Big|\{x_{(\mathrm{j}+1)T}\} \cup S_{\mathrm{j}}\Big| \leq \lfloor\sqrt{n}\rfloor. \tag{2.59}$$

Furthermore, for all $\mathrm{j} < k < \lfloor\sqrt{n}\rfloor$ we have

$$|S_k| \leq k + 1 \leq \lfloor\sqrt{n}\rfloor\,. \tag{2.60}$$

Combining Eq. (2.59) and Eq. (2.60) gives

$$|Q_{\mathrm{i}}| \leq \lfloor\sqrt{n}\rfloor \qquad \forall \mathrm{j}+1 \leq \mathrm{i} \leq \lfloor\sqrt{n}\rfloor\,. \tag{2.61}$$

We obtain:

$$\begin{aligned}
&\Pr\Big[good(\mathbf{z}) \wedge J(\mathbf{x},\mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{z},\mathrm{j})\Big] \\
&\quad \geq \prod_{\mathrm{i}=\mathrm{j}+1}^{\lfloor\sqrt{n}\rfloor} \min_{\xi\in\mathcal{W}_{\mathrm{i}-1}} \Pr\Big[z_{\mathrm{i}T} \notin Q_{\mathrm{i}} \mid Head(\mathbf{z},\mathrm{i}-1) = \xi\Big] && \text{(By Eq. (2.56))} \\
&\quad \geq \left(1 - \frac{2\sigma\lfloor\sqrt{n}\rfloor}{n}\right)^{\lfloor\sqrt{n}\rfloor - \mathrm{j}} \\
&\qquad\qquad \text{(By Eq. (2.58) with } R = Q_{\mathrm{i}} \text{ and since } |Q_{\mathrm{i}}| \leq \sqrt{n} \text{ by Eq. (2.61).)} \\
&\quad \geq \left(1 - \frac{2\sigma}{\sqrt{n}}\right)^{\sqrt{n}}\,. && (2.62)
\end{aligned}$$

By Lemma 33, we have that $\left(1 - \frac{2\sigma}{\sqrt{n}}\right)^{\sqrt{n}}$ is an increasing function of $\sqrt{n}$ since $n \geq 16\sigma^2$. Therefore it is minimized at $\sqrt{n} = 4\sigma$. Substituting in Eq. (2.62), we get

$$\Pr\Big[good(\mathbf{z}) \wedge J(\mathbf{x},\mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{z},\mathrm{j})\Big] \geq \left(1 - \frac{2\sigma}{4\sigma}\right)^{4\sigma} = 2^{-4\sigma}\,. \tag{2.63}$$

Case 2: $\mathrm{j} = \lfloor\sqrt{n}\rfloor - 1$.

We again invoke Eq. (2.58). In this case,

$$\left|Q_{\lfloor\sqrt{n}\rfloor}\right| \leq \left|S_{\lfloor\sqrt{n}\rfloor-1}\right| + 1 \leq \sqrt{n} + 1\,. \tag{2.64}$$

Using Eq. (2.56), we obtain:

$$\Pr\left[good(\mathbf{z}) \wedge J(\mathbf{x},\mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{z},\mathrm{j})\right] \\ \geq \min_{\xi \in \mathcal{W}_{\lfloor\sqrt{n}\rfloor-1}} \Pr\left[z_{\lfloor\sqrt{n}\rfloor T} \notin Q_{\lfloor\sqrt{n}\rfloor} \mid Head(\mathbf{z}, \lfloor\sqrt{n}\rfloor - 1) = \xi\right]. \tag{2.65}$$

Using Eq. (2.58) with $R = Q_{\lfloor\sqrt{n}\rfloor}$ and since $\left|Q_{\lfloor\sqrt{n}\rfloor}\right| \leq \sqrt{n} + 1$ by Eq. (2.64), we can lower bound the right hand side of Eq. (2.65) as follows:

$$\min_{\xi \in \mathcal{W}_{\lfloor\sqrt{n}\rfloor-1}} \Pr\left[z_{\lfloor\sqrt{n}\rfloor T} \notin Q_{\lfloor\sqrt{n}\rfloor} \mid Head(\mathbf{z}, \lfloor\sqrt{n}\rfloor - 1) = \xi\right] \geq 1 - \frac{2\sigma(\lfloor\sqrt{n}\rfloor + 1)}{n}. \tag{2.66}$$

Combining Eq. (2.65) and Eq. (2.66), we get

$$\Pr\left[good(\mathbf{z}) \wedge J(\mathbf{x},\mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{z},\mathrm{j})\right] \geq 1 - \frac{2\sigma(\lfloor\sqrt{n}\rfloor + 1)}{n}. \tag{2.67}$$

We can further lower bound the right hand side of Eq. (2.67) as follows:

$$\begin{aligned} 1 - \frac{2\sigma(\lfloor\sqrt{n}\rfloor + 1)}{n} &\geq 1 - \frac{\sqrt{n}(\sqrt{n}+1)}{2n} && (\text{Since } \sqrt{n} \geq 4\sigma.) \\ &\geq \frac{1}{4} && (\text{Since } n \geq 16\sigma^2 \geq 16.) \\ &\geq 2^{-4\sigma}. && (\text{Since } \sigma \geq 1.) \end{aligned} \tag{2.68}$$

Combining Eq. (2.67) and Eq. (2.69), we get

$$\Pr\left[good(\mathbf{z}) \wedge J(\mathbf{x},\mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{z},\mathrm{j})\right] \geq 2^{-4\sigma}. \tag{2.69}$$

In both cases, we obtained (by Eq. (2.63) and Eq. (2.69))

$$\Pr\left[good(\mathbf{z}) \wedge J(\mathbf{x},\mathbf{z}) = \mathrm{j} \mid Head(\mathbf{x},\mathrm{j}) = Head(\mathbf{z},\mathrm{j})\right] \geq 2^{-4\sigma}. \tag{2.70}$$

Combining Eq. (2.47) with Eq. (2.70) yields

$$\sum_{\substack{\mathbf{y}\in\mathcal{W}:\\ J(\mathbf{x},\mathbf{y})=\mathrm{j}\\ good(\mathbf{y})}} \mathcal{P}[Tail(\mathbf{y},\mathrm{j})]\cdot\mathcal{P}_{y_{\mathrm{j}\cdot T},\, y_{\mathrm{j}\cdot T+1}} \geq 2^{-4\sigma}\,.$$

This concludes the proof of the lemma. □

The next lemma is inspired by Lemma 9 from [5]. However, it is slightly different, so we include the proof here.

Lemma 31. *Let $v \in V$ and $\mathcal{Z} \subseteq \mathcal{X}$. Then we have*

$$\sum_{g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}\in\mathcal{Z}} r(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2})\mathbb{1}_{\{g_{\mathbf{x},b_1}(v)\neq g_{\mathbf{y},b_2}(v)\}} \leq 2 \sum_{\substack{g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}\in\mathcal{Z}:\\ v\in Tail(\mathbf{y},J(\mathbf{x},\mathbf{y}))}} r(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2})\,.$$

Proof. If $g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)$, then either:

- $v \in Tail(\mathbf{x}, J(\mathbf{x},\mathbf{y})) \cup Tail(\mathbf{y}, J(\mathbf{x},\mathbf{y}))$
- or $\mathbf{x} = \mathbf{y}$, in which case $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = 0$.

Therefore

$$\begin{aligned}\sum_{g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}\in\mathcal{Z}} r(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2})\mathbb{1}_{\{g_{\mathbf{x},b_1}(v)\neq g_{\mathbf{y},b_2}(v)\}} &\leq \sum_{\substack{g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}\in\mathcal{Z}:\\ v\in Tail(\mathbf{x},J(\mathbf{x},\mathbf{y}))\cup Tail(\mathbf{y},J(\mathbf{x},\mathbf{y}))}} r(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2})\\ &\leq \sum_{\substack{g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}\in\mathcal{Z}:\\ v\in Tail(\mathbf{x},J(\mathbf{x},\mathbf{y}))}} r(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}) + \sum_{\substack{g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}\in\mathcal{Z}:\\ v\in Tail(\mathbf{y},J(\mathbf{x},\mathbf{y}))}} r(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2})\\ &= 2\sum_{\substack{g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}\in\mathcal{Z}:\\ v\in Tail(\mathbf{y},J(\mathbf{x},\mathbf{y}))}} r(g_{\mathbf{x},b_1},g_{\mathbf{y},b_2}) \quad \text{(By symmetry of } r.)\end{aligned}$$

This completes the proof of the lemma. □

Lemma 32. *In the setting of Theorem 2.3.1, if $n \geq 16\sigma^2$ then there exists a subset $\mathcal{Z} \subseteq \mathcal{X}$ with $q(\mathcal{Z}) > 0$.*

Proof. By definition of σ and because $\max_{v\in V}\boldsymbol{\pi}(v) \geq 1/n$, we have

$$\min_v \boldsymbol{\pi}(v) \geq 4/n^{1.5}\,. \tag{2.71}$$

Fix an arbitrary vertex u. Let S be the set of vertices unreachable from u via a random walk that evolves according to $\mathcal{P}$ and has at most $T = t_{mix}(\frac{\sigma}{2n})$ steps. Denote $s = |S|$. Then

$$(\mathcal{P}^T)_{u,v} = 0 \;\; \forall v \in S\,. \tag{2.72}$$

By definition of t_{mix}, we have

$$\begin{aligned} \frac{\sigma}{2n} &\geq \frac{1}{2}\sum_{v\in V}\left|(\mathcal{P}^T)_{u,v} - \boldsymbol{\pi}(v)\right| \\ &\geq \frac{1}{2}\sum_{v\in S}|\boldsymbol{\pi}(v)| && \text{(By Eq. (2.72))} \\ &\geq \frac{s}{2}\min_{v\in V}\boldsymbol{\pi}(v) && \text{(Since } s = |S|.\text{)} \\ &\geq \frac{2s}{n^{1.5}}\,. && \text{(By Eq. (2.71).)} \end{aligned} \tag{2.73}$$

Meanwhile, since $n \geq 16\sigma^2$, we have

$$\frac{\sigma}{2n} \leq \frac{1}{8\sqrt{n}}\,. \tag{2.74}$$

Combining Eq. (2.73) and Eq. (2.74), we get $s \leq n/16$. Thus the number of vertices reachable from u in T steps is $n - s \geq 15n/16$. For $n \geq 5$, we have $15n/16 \geq \lfloor\sqrt{n}\rfloor + 2$, which means that at least $\lfloor\sqrt{n}\rfloor + 2$ vertices are reachable via $\mathcal{P}$ from any vertex u within T steps.

We can then construct two walks, $\mathbf{x}$ and $\mathbf{y}$, in the following manner. For $\mathrm{i} = 1, \ldots, \lfloor\sqrt{n}\rfloor$, take $x_{\mathrm{i}T}$ to be an arbitrary vertex reachable from $x_{(\mathrm{i}-1)T}$ other than $x_0, \ldots, x_{(\mathrm{i}-1)T}$; this is possible since at least $\lfloor\sqrt{n}\rfloor + 2$ vertices are reachable. Connect the milestones using an arbitrary path such that every edge (u, w) in the path has $\mathcal{P}_{u,w} > 0$. Construct $\mathbf{y}$ in the same manner but requiring that $Head(\mathbf{y}, \lfloor\sqrt{n}\rfloor - 1) = Head(\mathbf{x}, \lfloor\sqrt{n}\rfloor - 1)$ and $y_L \notin \left\{x_0, x_T, x_{2T}, \ldots, x_{\lfloor\sqrt{n}\rfloor T}\right\}$.

Define $\mathcal{Z} = \{g_{\mathbf{x},0}, g_{\mathbf{y},1}\}$. We will show that $q(\mathcal{Z}) > 0$. First observe that $\mathbf{x} \neq \mathbf{y}$, both $\mathbf{x}$ and $\mathbf{y}$ are *good* since they do not repeat milestones, and the bit hidden by $g_{\mathbf{x},0}$ is different from the bit hidden by $g_{\mathbf{y},1}$. The length of the prefix shared by $\mathbf{x}$ and $\mathbf{y}$ is $J(\mathbf{x}, \mathbf{y}) = \lfloor\sqrt{n}\rfloor - 1$. Then

$$r(g_{\mathbf{x},0}, g_{\mathbf{y},1}) = \frac{\mathcal{P}[\mathbf{x}]\mathcal{P}[\mathbf{y}]}{\mathcal{P}[Head(\mathbf{y}, \lfloor\sqrt{n}\rfloor - 1)]} > 0\,. \tag{2.75}$$

We have

$$\begin{aligned} q(\mathcal{Z}) &= \max_{v\in V} \sum_{F_1\in\mathcal{Z}} \sum_{F_2\in\mathcal{Z}} r(F_1, F_2) \cdot \mathbb{1}_{\{F_1(v)\neq F_2(v)\}} \\ &= 2\max_{v\in V} r(g_{\mathbf{x},0}, g_{\mathbf{y},1}) \cdot \mathbb{1}_{\{g_{\mathbf{x},0}(v)\neq g_{\mathbf{y},1}(v)\}} \\ &\qquad\qquad (\text{Since } \mathcal{Z} = \{g_{\mathbf{x},0}, g_{\mathbf{y},1}\} \text{ and } r(F,F) = 0\ \forall F \in \mathcal{Z}.) \\ &> 0\,. \qquad (\text{Using Eq. (2.75) and } g_{\mathbf{x},0}(x_L) \neq g_{\mathbf{y},1}(y_L).) \end{aligned}$$

This completes the proof. ☐

Lemma 33. *For all $x \geq 2y \geq 1$ we have*

$$\frac{\partial}{\partial x}\left(1 - \frac{y}{x}\right)^x \geq 0\,. \tag{2.76}$$

Proof. Define $z = x/y$. Then

$$\begin{aligned} \frac{\partial}{\partial x}\left(1 - \frac{y}{x}\right)^x &= \frac{\partial z}{\partial x}\frac{\partial}{\partial z}\left(\left(1 - \frac{1}{z}\right)^z\right)^y && (\text{By chain rule.}) \\ &= \frac{y}{y}\left(\left(1 - \frac{1}{z}\right)^z\right)^{y-1}\frac{\partial}{\partial z}\left(1 - \frac{1}{z}\right)^z && (\text{By chain rule.}) \\ &\geq 0\,. && (\text{Since } z > 1) \end{aligned}$$

This concludes the proof of the lemma. ☐

Lemma 34 (Folklore)**.** *Consider a reversible Markov chain on G with transition matrix $\mathcal{P}$. For all $u, v \in V$ and $\ell \in \mathbb{N}$, we have*

$$\mathrm{E}_{visit}(u, v, \ell)\pi(u) = \mathrm{E}_{visit}(v, u, \ell)\pi(v)\,.$$

Proof. We have

$$\begin{aligned}
\mathrm{E}_{visit}(u, v, \ell)\pi(u) &= \sum_{\mathrm{i}=1}^{\ell} \pi(u)\mathcal{P}^{\mathrm{i}}_{u,v} && \text{(By definition of } \mathrm{E}_{visit}.) \\
&= \sum_{\mathrm{i}=1}^{\ell} \pi(v)\mathcal{P}^{\mathrm{i}}_{v,u} && \text{(By [52] equation 1.30.)} \\
&= \mathrm{E}_{visit}(v, u, \ell)\pi(v)\,. && \text{(By definition of } \mathrm{E}_{visit}.)
\end{aligned}$$

This concludes the proof of the lemma. □

We also use the following lemma from [5].

Lemma 35 ([5], Lemma 6)**.** *Suppose $\mathbf{x} = (x_0, x_1, \ldots, x_\ell)$ is a walk on G and $f : V \to \mathbb{R}$ is a valid function for the walk $\mathbf{x}$. Then f has a unique local minimum at x_ℓ, the last vertex on the walk.*

All of the lemmas in this section are heavily based on lemmas from prior work. However they are slightly different, so we include their proofs here for completion.

2.6 Corollaries of the main theorem

We can connect Theorem 2.3.1 to the spectral gap of the transition matrix of the Markov chain used via the following inequality (see, e.g., [52], Theorem 12.4): If $\mathcal{P}$ is lazy, irreducible, and time-reversible, then

$$t_{mix}(\epsilon) \leq \left(\frac{1}{1-\lambda_2}\right) \log\left(\frac{1}{\epsilon \min_{x\in V} \pi(x)}\right)\,. \tag{2.77}$$

We obtain the following corrollary, which lower bounds the randomized complexity of local search as a function of the spectral gap of $\mathcal{P}$.

Corollary 5. *Let $G = (V, E)$ be a connected undirected graph on n vertices. Consider a discrete-time, lazy, irreducible, and reversible Markov chain on G with transition matrix $\mathcal{P}$ and stationary distribution $\boldsymbol{\pi}$. The randomized query complexity of local search on G is*

$$\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log(n)\exp(3\sigma)}\right),$$

where λ_2 is the second eigenvalue of $\mathcal{P}$ and $\sigma = \max_{u,v \in V} \boldsymbol{\pi}(v)/\boldsymbol{\pi}(u)$.

Proof. We proceed by substituting Eq. (2.77) into Theorem 2.3.1. This directly yields that the randomized query complexity of local search on G is

$$\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log\left(2n/(\sigma \min_{x\in V} \boldsymbol{\pi}(x))\right)\exp(3\sigma)}\right). \tag{2.78}$$

By definition of σ we have

$$\sigma \min_{x\in V} \boldsymbol{\pi}(x) = \max_{x\in V} \boldsymbol{\pi}(x) \geq 1/n\,. \tag{2.79}$$

Combining Eq. (2.78) with Eq. (2.79) yields that the randomized query complexity of local search on G is

$$\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log(n)\exp(3\sigma)}\right). \tag{2.80}$$

This completes the proof of the corollary. □

The prior work in [5] implies a result similar to but weaker than Corollary 5. To get a result in terms of spectral gap via [5], we need to connect the edge expansion β to the spectral gap $1-\lambda_2$ of a particular Markov chain. We will do this via the bottleneck ratio $\Phi_\star$.

Lemma 36 ([52], Theorem 13.3)**.** *If the Markov chain is lazy, then*

$$\frac{\Phi_\star^2}{2} \leq 1 - \lambda_2 \leq 2\Phi_\star \,. \tag{2.81}$$

We can use this to get a bound on local search in terms of spectral gap via the following lemma from [5].

Lemma 37 ([5], corollary 2)**.** *The randomized query complexity of local search is in*

$$\Omega\left(\frac{\beta\sqrt{n}}{d_{max}\log^2(n)}\right) \tag{2.82}$$

As a special case, consider the simple lazy random walk on a graph with $d_{max}/d_{min} \leq C$ for constant C. Applying Lemma 37 to this walk yields the following result:

Proposition 2.3.1. *[5] Let $G = (V, E)$ be a connected undirected graph on n vertices. If $d_{max}/d_{min} \leq C$ for some constant $C > 0$, then the randomized query complexity of local search on G is*

$$\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log^2(n)}\right),$$

where λ_2 is the second eigenvalue of the transition matrix of the simple lazy random walk on G.

Proof. For the simple lazy random walk we have $\pi(u) = d(u)/(2|E|)$ for all $u \in V$. Let $\mathcal{P}$ be the transition matrix of the simple lazy random walk. Then

$$\begin{aligned}
\Phi_\star &= \min_{S\subseteq V \,\mid\, \pi(S)\leq 1/2} \frac{\sum_{(u,v)\in E(S,S^c)} \pi(u)\mathcal{P}_{u,v}}{\pi(S)} && \text{(By definition of } \Phi_\star.\text{)}\\
&= \min_{S\subseteq V \,\mid\, \pi(S)\leq 1/2} \frac{\sum_{(u,v)\in E(S,S^c)} (d(u)/(2|E|))(1/d(u))}{\sum_{u\in S}(d(u)/(2|E|))} && \text{(By definition of } \mathcal{P}.\text{)}\\
&= \min_{S\subseteq V \,\mid\, \pi(S)\leq 1/2} \frac{|E(S,S^c)|}{\sum_{u\in S} d(u)} \,. && (2.83)
\end{aligned}$$

To get a term of β, we need to change the scope of S from

$$\{S \subseteq V \mid \pi(S) \leq 1/2\}$$

to

$$\{S \subseteq V \mid |S| \leq n/2\}\,.$$

Let S^* be a minimizing choice of S from $\{S \subseteq V \mid |S| \leq n/2\}$ for $|E(S, S^c)|/\sum_{u \in S} d(u)$. Then either S^* or S^{*c} (or both) will be in $\{S \subseteq V \mid \pi(S) \leq 1/2\}$. We analyze these two cases separately.

Case 1: $\pi(S^*) \leq 1/2$.

Continuing from Eq. (2.83), this gives us

$$\begin{aligned}\Phi_\star &\leq \frac{|E(S^*, S^{*c})|}{\sum_{u \in S^*} d(u)} && \text{(Since } \pi(S^*) \leq 1/2.\text{)}\\ &= \min_{S \subseteq V \mid |S| \leq n/2} \frac{|E(S, S^c)|}{\sum_{u \in S} d(u)}\,. && \text{(By definition of } S^*.\text{)}\end{aligned} \tag{2.84}$$

Case 2: $\pi(S^{*c}) \leq 1/2$.

Continuing from Eq. (2.83), this gives us

$$\begin{aligned}\Phi_\star &\leq \frac{|E(S^*, S^{*c})|}{\sum_{u \in S^{*c}} d(u)} && \text{(Since } \pi(S^{*c}) \leq 1/2.\text{)}\\ &\leq \frac{|E(S^*, S^{*c})|}{\sum_{u \in S^*} d(u)} \cdot C && \text{(Since } |S^*| \leq n/2 \leq |S^{*c}| \text{ and by definition of } C.\text{)}\\ &= \min_{S \subseteq V \mid |S| \leq n/2} \frac{|E(S, S^c)|}{\sum_{u \in S} d(u)} \cdot C\,. && \text{(By definition of } S^*.\text{)}\end{aligned} \tag{2.85}$$

In both cases, we have

$$\begin{aligned}
\Phi_\star &\leq \min_{S \subseteq V \mid |S| \leq n/2} \frac{|E(S, S^c)|}{\sum_{u \in S} d(u)} \cdot C && \text{(By Eq. (2.84) and Eq. (2.85))} \\
&\leq \min_{S \subseteq V \mid |S| \leq n/2} \frac{|E(S, S^c)|}{|S| \cdot d_{min}} \cdot C && \text{(By definition of } d_{min}.\text{)} \\
&= \frac{\beta \cdot C}{d_{min}} && \text{(By definition of } \beta.\text{)} \\
&\leq \frac{\beta \cdot C^2}{d_{max}} . && \text{(Since } d_{max}/d_{min} \leq C.\text{)}
\end{aligned} \tag{2.86}$$

Substituting Eq. (2.86) into Lemma 36, we get

$$1 - \lambda_2 \leq \frac{2\beta \cdot C^2}{d_{max}} . \tag{2.87}$$

Substituting Eq. (2.87) into Lemma 37, we get a lower bound of $\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log^2(n)}\right)$ on the randomized query complexity of local search on G. This completes the proof of the corollary. □

Compare to the following corollary, which is just Corollary 5 applied to the simple lazy random walk when d_{max}/d_{min} is bounded by a constant C.

Corollary 6. *Let $G = (V, E)$ be a connected undirected graph on n vertices. If $d_{max}/d_{min} \leq C$ for some constant $C > 0$, then the randomized query complexity of local search on G is*

$$\Omega\left(\frac{(1 - \lambda_2)\sqrt{n}}{\log n}\right),$$

where λ_2 is the second eigenvalue of the transition matrix of the simple lazy random walk on G.

Proof. Since $d_{max}/d_{min} \leq C$, we have that $\exp(3\sigma) \leq \exp(3C)$, which is a constant. Therefore Corollary 5 directly gives that the randomized query complexity of local search is

$$\Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log(n)\exp(3\sigma)}\right) = \Omega\left(\frac{(1-\lambda_2)\sqrt{n}}{\log(n)}\right). \tag{2.88}$$

□

Corollary 6 is stronger than Proposition 2.3.1 by a factor of $\log(n)$. This represents an improvement of this paper in bounding the difficulty of local search in terms of spectral gap.

3. QUANTUM LOWER BOUNDS FOR LOCAL SEARCH

3.1 Introduction

A powerful method for giving lower bounds in the quantum setting is the relational adversary method [1]. Several variants of the method exist, such as the strong weighted adversary [2]. [55] showed that multiple quantum relational adversary methods are in fact equivalent.

We will show a lower bound on local search in the quantum setting using the strong weighted adversary method applied to a construction from [56].

3.2 Model

Let $G = (V, E)$ be a connected undirected graph and $f : V \to \mathbb{R}$ a function defined on the vertices. A vertex $v \in V$ is a local minimum if $f(v) \leq f(u)$ for all $\{u, v\} \in E$. We will write $V = [n] = \{1, \ldots, n\}$.

Given as input a graph G and oracle access to function f, the local search problem is to find a local minimum of f on G using as few queries as possible. Each query is of the form: "Given a vertex v, what is $f(v)$?".

Query complexity.

Consider a problem where the input F has n queryable locations, $F(1)$, $F(2)$, …, $F(n)$, each of which has k possible values. In the quantum query model, the quantum bits are partitioned into four regions: the input F, the $\log n$ index bits i, the $\log k$ return bits a, and the rest of the workspace bits z. Collectively, i, a, z constitute the algorithm's workspace; it may freely manipulate these bits via arbitrary unitary transformations at no cost, so long as the transformations do not depend on the input F. To access F, the algorithm is provided the oracle transformation O, which is defined such that, for all values of F, i, a, z (such that $\mathrm{i} \leq N$):

$$O\left(|F, \mathrm{i}, a, z\rangle\right) = |F, \mathrm{i}, a \oplus F(\mathrm{i}), z\rangle\,.$$

Of course, not all quantum states are in the form $|F, \mathrm{i}, a, z\rangle$; you may have a superposition over such states. A superposition is just a linear combination though, so this is sufficient to define the behavior of O on the space spanned by such kets. I.e., given a state $|\psi\rangle = \sum_{F,\mathrm{i},a,z} \alpha_{F,\mathrm{i},a,z} |F, \mathrm{i}, a, z\rangle$ for some scalars $\alpha_{F,\mathrm{i},a,z}$, we have

$$O\left(|\psi\rangle\right) = \sum_{F,\mathrm{i},a,z} \alpha_{F,\mathrm{i},a,z} O\left(|F, \mathrm{i}, a, z\rangle\right) = \sum_{F,\mathrm{i},a,z} \alpha_{F,\mathrm{i},a,z} |F, \mathrm{i}, a \oplus F(\mathrm{i}), z\rangle\,.$$

Between queries, the algorithm is free to manipulate its own bits (i.e. i, a, z but not F) freely. More formally, it may apply an arbitrary unitary transformation U subject to the constraint $U(|F\rangle \otimes |\psi\rangle) = |F\rangle \otimes |\psi'\rangle$ for all ψ and the algorithm's choice of ψ'. Let U_j be the transformation after the jth query.

Then, the algorithm as a whole applies, in order, $U_0, O, U_1, O \ldots, U_{\mathrm{j}-1}, O, U_\mathrm{j}$, and finally measures the final state. The measured final state of a particular bit of the algorithm's workspace is interpreted as the returned answer. The query complexity of such an algorithm is j. The quantum query complexity of a problem is the minimum query complexity among all quantum algorithms that give a correct answer with probability at least 2/3.

Congestion.

Let $\mathcal{P} = \{P^{u,v}\}_{u,v \in V}$ be an all-pairs set of paths in G, where $P^{u,v}$ is a path from u to v. For convenience, we assume $P^{u,u} = (u)$ for all $u \in V$; our results will hold even if $P^{u,u} = \emptyset$.

For a path $Q = (v_1, \ldots, v_s)$ in G, let c_v^Q be the number of times a vertex $v \in V$ appears in Q and c_e^Q the number of times an edge $\mathrm{e} \in E$ appears in Q. The *vertex congestion* of the set of paths $\mathcal{P}$ is $\max_{v \in V} \sum_{Q \in \mathcal{P}} c_v^Q$, while the *edge congestion* of $\mathcal{P}$ is $\max_{\mathrm{e} \in E} \sum_{Q \in \mathcal{P}} c_\mathrm{e}^Q$.

The *vertex congestion of* G is the smallest integer g for which the graph has an all-pairs set of paths $\mathcal{P}$ with vertex congestion g. Clearly, $g \geq n$ since each vertex belongs to at least n paths in $\mathcal{P}$ and $g \leq n^2$ since each vertex appears at most once on each path and there are n^2 paths in $\mathcal{P}$. The *edge congestion* g_{e} is similarly defined, but with respect to the edge congestion of a set of paths $\mathcal{P}$.

d-regular expanders.

For each set of vertices $S \subseteq V$, the edges with one endpoint in S and another in $V \setminus S$ are called *cut edges* and denoted $E(S, V \setminus S) = \{(u,v) \in E \mid u \in S, v \notin S\}$. The graph is a β-expander if $|E(S, V \setminus S)| \geq \beta \cdot |S|$, for all $S \subseteq V$ with $0 < |S| \leq n/2$ (see, e.g. [15]). The graph is *d-regular* if each vertex has degree d.

Distance.

For each $u, v \in V$, let $dist(u,v)$ be the length of the shortest path from u to v.

3.3 Our results

Our main result is the following theorem, which provides a lower bound on the quantum query complexity of local search in terms of congestion.

Theorem 3.3.1. *The quantum query complexity of local search on an undirected graph $G = (V, E)$ with n vertices and vertex congestion g is $\Omega\left(\frac{n^{0.75}}{\sqrt{g}}\right)$.*

This bound is the square root of the similar bound of $\Omega\left(n^{1.5}/g\right)$ on the *classical* randomized query complexity of local search found by [56] and uses a similar construction. This is a natural relationship for quantum and classical bounds to have with each other, which suggests that if either bound is later improved, the other can be improved by the same means.

On constant degree expanders, we get the following corollary:

Corollary 7. *The quantum query complexity of local search on an undirected d-regular graph $G = (V, E)$ with n vertices and expansion β, where d and β are constants, is $\Omega\left(\frac{n^{0.25}}{\sqrt{\log(n)}}\right)$.*

Proof. By Lemma 44, such graphs have congestion $g \in O(n \log(n))$. Therefore by Theorem 3.3.1, the quantum query complexity on such graphs is $\Omega\left(\frac{n^{0.25}}{\sqrt{\log(n)}}\right)$. □

We compare this bound to the quantum extension of Aldous' upper bound detailed in [3]:

Lemma 38 ([3], theorem 3.2)**.** *The quantum query complexity of local search on an undirected graph $G = (V, E)$ with n vertices and maximum degree δ is $O\left(n^{1/3}\delta^{1/6}\right)$.*

In the case of constant degree, this bound simplifies to $O(n^{1/3})$. This still leaves a polynomial gap between the upper and lower bounds for quantum local search on constant degree expanders, in contrast to the classical case where the remaining gap is only logarithmic. Closing this gap remains a subject for future work.

3.4 Preliminaries

The strong weighted adversary method

The strong weighted adversary method that we use [2] is given by the next lemma.

Lemma 39 (The strong weighted adversary method [2])**.** *Let A, B be finite sets and $\mathcal{X} \subseteq B^A$ a set of functions mapping A to B. Let $\mathcal{H} : \mathcal{X} \to \{0, 1\}$ be a map that assigns to each function in $\mathcal{X}$ the label 0 or 1. We are given oracle access to a function $F_1 \in \mathcal{X}$ and the problem is to compute the label $\mathcal{H}(F_1)$ using as few queries as possible.*

Let $r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a non-zero symmetric function of our choice with $r(F_1, F_2) = 0$ when $\mathcal{H}(F_1) = \mathcal{H}(F_2)$. Additionally, define

- *$r' : \mathcal{X} \times \mathcal{X} \times A \to \mathbb{R}_{\geq 0}$ such that $r'(F_1, F_2, a) \cdot r'(F_2, F_1, a) \geq r^2(F_1, F_2)$ for all $F_1, F_2 \in \mathcal{X}$ and all $a \in A$ with $F_1(a) \neq F_2(a)$.*

- *$M : \mathcal{X} \to \mathbb{R}$ such that $M(F_1) = \sum_{F_2 \in \mathcal{X}} r(F_1, F_2)$ for all $F_1 \in \mathcal{X}$ and all $a \in A$.*

- $\nu : \mathcal{X} \times A \to \mathbb{R}$ *such that* $\nu(F_1, a) = \sum_{F_2 \in \mathcal{X}} r'(F_1, F_2, a)$ *for all* $F_1 \in \mathcal{X}$ *and all* $a \in A$.

Then the quantum query complexity of the problem is at least:

$$\min_{\substack{F_1, F_2 \in \mathcal{X},\, a \in A: \\ r(F_1, F_2) > 0 \\ F_1(a) \neq F_2(a)}} \sqrt{\frac{M(F_1)M(F_2)}{\nu(F_1, a)\nu(F_2, a)}} \,. \tag{3.1}$$

Family of input functions

Next we explain the construction from [56] that we use.

Since the graph G has vertex congestion g, there is an all-pairs set of paths $\mathcal{P} = \{P^{u,v}\}_{u,v \in [n]}$ with vertex congestion g [1].

For each $u, v \in [n]$, let $\mathfrak{q}_v(u)$ be the number of paths in $\mathcal{P}$ that start at vertex u and contain v:

$$\mathfrak{q}_v(u) = \left|\{P^{u,w} \in \mathcal{P} : w \in [n], v \in P^{u,w}\}\right| . \tag{3.2}$$

Let $L \in [n]$, with $L \geq 2$, be a parameter that we set later.

Given a sequence of k vertices $\mathbf{x} = (x_1, \ldots, x_k)$, we write $\mathbf{x}_{1 \to \mathrm{j}} = (x_1, \ldots, x_{\mathrm{j}})$ to refer to a prefix of the sequence, for an index $\mathrm{j} \in [k]$.

Given a walk $Q = (v_1, \ldots, v_k)$ in G, let Q_{i} refer to the i-th vertex in the walk (i.e. $Q_{\mathrm{i}} = v_{\mathrm{i}}$). For each vertex $u \in [n]$, let $\mu(Q, u)$ be the number of times that vertex u appears in Q.

Definition 3.4.1 (Staircase)**.** *Given a sequence* $\mathbf{x} = (x_1, \ldots, x_k)$ *of vertices in* G*, a* staircase *induced by* $\mathbf{x}$ *is a walk* $S_{\mathbf{x}} = S_{\mathbf{x},1} \circ \ldots \circ S_{\mathbf{x},k-1}$*, where each* $S_{\mathbf{x},\mathrm{i}}$ *is a path in* G *starting at vertex* x_{i} *and ending at* $x_{\mathrm{i}+1}$*. Each vertex* x_{i} *is called a* milestone *and each path* $S_{\mathbf{x},\mathrm{i}}$ *a* quasi-segment*.*

[1]↑That is, each vertex is used at most g times across all the paths.

The staircase $S_{\mathbf{x}}$ *is said to be* induced by $\mathbf{x}$ and $\mathcal{P} = \{P^{u,v}\}_{u,v\in[n]}$ *if additionally* $S_{\mathbf{x},\mathrm{i}} = P^{x_\mathrm{i},x_{\mathrm{i}+1}}$ *for all* $\mathrm{i} \in [k-1]$.

Definition 3.4.2 (Tail of a staircase)**.** *Let* $S_{\mathbf{x}} = S_{\mathbf{x},1} \circ \ldots \circ S_{\mathbf{x},k-1}$ *be a staircase induced by some sequence* $\mathbf{x} = (x_1, \ldots, x_k) \in [n]^k$*. For each* $\mathrm{j} \in [k-1]$*, let* $T = S_{\mathbf{x},\mathrm{j}} \circ \ldots \circ S_{\mathbf{x},k-1}$*. Then* $Tail(\mathrm{j}, S_{\mathbf{x}})$ *is obtained from* T *by removing the first occurrence of* x_j *in* T *(and only the first occurrence). We also define* $Tail(k, S_{\mathbf{x}})$ *to be the empty sequence.*

Next we define a set of functions $\mathcal{X}$.

Definition 3.4.3 (The functions $f_{\mathbf{x}}$ and $g_{\mathbf{x},b}$; the set $\mathcal{X}$)**.** *Suppose* $\mathcal{P} = \{P^{u,v}\}_{u,v\in[n]}$ *is an all-pairs set of paths in* G*. For each sequence of vertices* $\mathbf{x} \in \{1\} \times [n]^L$*, define a function* $f_{\mathbf{x}} : [n] \to \{-n^2 - n, \ldots, 0, \ldots, n\}$ *such that for each* $v \in [n]$*:*

- *If* $v \notin S_{\mathbf{x}}$*, then set* $f_{\mathbf{x}}(v) = dist(v, 1)$*, where* $S_{\mathbf{x}}$ *is the staircase induced by* $\mathbf{x}$ *and* $\mathcal{P}$*.*
- *If* $v \in S_{\mathbf{x}}$*, then set* $f_{\mathbf{x}}(v) = -\mathrm{i} \cdot n - \mathrm{j}$*, where* i *is the maximum index with* $v \in P^{x_\mathrm{i},x_{\mathrm{i}+1}}$*, and* v *is the* j*-th vertex in* $P^{x_\mathrm{i},x_{\mathrm{i}+1}}$*.*

Also, for each $\mathbf{x} \in \{1\}\times[n]^L$ *and* $b \in \{0,1\}$*, let* $g_{\mathbf{x},b} : [n] \to \{-n^2-n, \ldots, 0, \ldots, n\}\times\{-1, 0, 1\}$ *be such that, for all* $v \in [n]$*:*

$$g_{\mathbf{x},b}(v) = \begin{cases} \Big(f_{\mathbf{x}}(v), b\Big) & \text{if } v = x_{L+1} \\ \Big(f_{\mathbf{x}}(v), -1\Big) & \text{if } v \neq x_{L+1} \end{cases} . \tag{3.3}$$

Let $\mathcal{X} = \Big\{ g_{\mathbf{x},b} \mid \mathbf{x} \in \{1\} \times [n]^L \text{ and } b \in \{0,1\} \Big\}$.

By Lemma 41 and Lemma 42, each such function f_{vecx} has a unique local minimum at x_L.

3.5 Quantum lower bound for local search via congestion

Theorem 3.3.1. *The quantum query complexity of local search on an undirected graph* $G = (V, E)$ *with* n *vertices and vertex congestion* g *is* $\Omega\left(\frac{n^{0.75}}{\sqrt{g}}\right)$.

Proof. Let $L = \sqrt{n}$. For each function in $\mathcal{X}$, its underlying staircase has $L+1$ milestones and L quasi-segments. For all sequences of milestones $\mathbf{x}, \mathbf{y}$ let $J_{\mathbf{x},\mathbf{y}}$ be the number of milestones in the shared prefix of $\mathbf{x}$ and $\mathbf{y}$.

We borrow the relation function defined in [56] as a component r^* for our relation function r. Let $r^* : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a symmetric function defined as follows. For each $\mathbf{x}, \mathbf{y} \in \{1\} \times [n]^L$ and $b_1, b_2 \in \{0,1\}$, we have

$$r^*(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = \begin{cases} 0 & \text{if at least one of the following holds: } b_1 = b_2 \text{ or } \mathbf{x} \text{ is bad or } \mathbf{y} \text{ is bad.} \\ n^{\mathrm{j}} & \text{otherwise, where j is the maximum index for which } \mathbf{x}_{1\to \mathrm{j}} = \mathbf{y}_{1\to \mathrm{j}}\,. \end{cases}$$

We then define our relation function r as follows:

$$r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) = \begin{cases} 0 & \text{if } \mathbf{x} = \mathbf{y} \\ r^*(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) & \text{otherwise} \end{cases}$$

For all $g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2} \in \mathcal{X}$ and $v \in V$ we define $r'(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}, v)$ as follows:

$$r'(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}, v) = \begin{cases} 0 & \text{if } g_{\mathbf{x},b_1}(v) = g_{\mathbf{y},b_2}(v) \text{ or } \mathcal{H}(g_{\mathbf{x},b_1}) = \mathcal{H}(g_{\mathbf{y},b_2}) \\ r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot g/n^{1.5} & \text{otherwise, if } v \in Tail(J_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x}}) \setminus Tail(J_{\mathbf{x},\mathbf{y}}, S_{\mathbf{y}}) \\ r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \cdot n^{1.5}/g & \text{otherwise, if } v \in Tail(J_{\mathbf{x},\mathbf{y}}, S_{\mathbf{y}}) \setminus Tail(J_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x}}) \\ r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) & \text{otherwise} \end{cases}$$

The functions r and r' constitute a weight scheme for use with the strong weighted adversary method.

Take an arbitrary choice of $g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2} \in \mathcal{X}$ and $v \in V$ such that $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$ and $g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)$. Since $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$, we have that both $g_{\mathbf{x},b_1}$ and $g_{\mathbf{y},b_2}$ are good. Therefore by Lemma 40 we have $M(g_{\mathbf{x},b_1}) \geq \frac{1}{2\mathrm{e}} L n^{L+1}$ and $M(g_{\mathbf{y},b_2}) \geq \frac{1}{2\mathrm{e}} L n^{L+1}$.

Next we bound $\nu(g_{\mathbf{x},b_1}, v)$:

Case 1: $v \in \mathit{Tail}(J_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x}})$

$$\begin{aligned}
\nu(g_{\mathbf{x},b_1}, v) &= \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ J_{\mathbf{x},\mathbf{y}}=\mathrm{j}}} r'(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}, v) && \text{(By definition of } \nu(g_{\mathbf{x},b_1}, v).\text{)}\\
&= n^{L+1} + \sum_{\mathrm{j}=1}^{L} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ J_{\mathbf{x},\mathbf{y}}=\mathrm{j}}} r'(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}, v)\\
& && \text{(Since exactly one } g_{\mathbf{y},b_2} \text{ has } J_{\mathbf{x},\mathbf{y}} = L+1 \text{ and } b_1 \neq b_2.\text{)}\\
&= n^{L+1} + \sum_{\mathrm{j}=1}^{L} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ J_{\mathbf{x},\mathbf{y}}=\mathrm{j}\\ v\in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})}} n^{\mathrm{j}} + \sum_{\mathrm{j}=1}^{L} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ J_{\mathbf{x},\mathbf{y}}=\mathrm{j}\\ v\notin \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})}} n^{\mathrm{j}} \cdot g/n^{1.5}\\
&\le n^{L+1} + \sum_{\mathrm{j}=1}^{L} \left(\mathfrak{q}_v(x_{\mathrm{j}}) n^{L-\mathrm{j}} + L \cdot g \cdot n^{L-\mathrm{j}-1}\right) n^{\mathrm{j}} + \sum_{\mathrm{j}=1}^{L} n^{L+1-\mathrm{j}} n^{\mathrm{j}} \cdot g/n^{1.5}\\
&\le n^{L+1} + g n^{L} + L^2 g n^{L-1} + L n^{L+1} g/n^{1.5}\\
&\le 4 g n^{L}
\end{aligned}$$

Case 2: $v \notin \mathit{Tail}(J_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x}})$

$$\begin{aligned}
\nu(g_{\mathbf{x},b_1}, v) &= \sum_{\mathrm{j}=1}^{L+1} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ J_{\mathbf{x},\mathbf{y}}=\mathrm{j}}} r'(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}, v) && \text{(By definition of } \nu(g_{\mathbf{x},b_1}, v).\text{)}\\
&= n^{L+1} + \sum_{\mathrm{j}=1}^{L} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ J_{\mathbf{x},\mathbf{y}}=\mathrm{j}}} r'(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}, v)\\
& && \text{(Since exactly one } g_{\mathbf{y},b_2} \text{ has } J_{\mathbf{x},\mathbf{y}} = L+1 \text{ and } b_1 \neq b_2.\text{)}\\
&= n^{L+1} + \sum_{\mathrm{j}=1}^{L} \sum_{\substack{g_{\mathbf{y},b_2}\in\mathcal{X}:\\ J_{\mathbf{x},\mathbf{y}}=\mathrm{j}\\ v\in \mathit{Tail}(\mathrm{j}, S_{\mathbf{y}})}} n^{\mathrm{j}} \cdot n^{1.5}/g\\
&\le n^{L+1} + \sum_{\mathrm{j}=1}^{L} \left(\mathfrak{q}_v(x_{\mathrm{j}}) n^{L-\mathrm{j}} + L \cdot g \cdot n^{L-\mathrm{j}-1}\right) n^{\mathrm{j}} \cdot n^{1.5}/g
\end{aligned}$$

$$
\begin{aligned}
&\leq n^{L+1} + gn^{L} \cdot n^{1.5}/g + L^2 g n^{L-1} \cdot n^{1.5}/g \\
&\leq 3n^{L+1.5}
\end{aligned}
$$

Because $g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)$ and $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$, we have $v \in Tail(J_{X,Y}, S_{\mathbf{x}}) \cup Tail(J_{X,Y})$. Then

$$
\begin{aligned}
\sqrt{\frac{M(g_{\mathbf{x},b_1})M(g_{\mathbf{y},b_2})}{\nu(\mathbf{x},v)\nu(\mathbf{y},v)}} &\geq \sqrt{\frac{\frac{1}{4e^2}L^2 n^{2L+2}}{4gn^L \cdot \max\{4gn^L, 3n^{L+1.5}\}}} \\
&\geq \sqrt{\frac{1}{64e^2} \cdot \min\{\frac{n^3}{g^2}, \frac{n^{1.5}}{g}\}} \\
&= \frac{1}{8e} \cdot \min\{\frac{n^{1.5}}{g}, \frac{n^{0.75}}{\sqrt{g}}\}
\end{aligned}
$$

If $g \geq n^{1.5}$, then the claimed bound is less than or equal to $\Omega(1)$, which is trivially true. Otherwise, $g < n^{1.5}$, in which case this bound simplifies to:

$$
\sqrt{\frac{M(g_{\mathbf{x},b_1})M(g_{\mathbf{y},b_2})}{\nu(\mathbf{x},v)\nu(\mathbf{y},v)}} \geq \frac{1}{8e} \cdot \frac{n^{0.75}}{\sqrt{g}}
$$

Since this holds for arbitrary choices of $g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}, v$ such that $r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) > 0$ and $g_{\mathbf{x},b_1}(v) \neq g_{\mathbf{y},b_2}(v)$, this via Lemma 39 implies that the quantum query complexity of local search is $\Omega(n^{0.75}/\sqrt{g})$.

□

Lemma 40. *If $g_{\mathbf{x},b_1} \in \mathcal{X}$ is good, then $\sum_{g_{\mathbf{y},b_2} \in \mathcal{X}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) \geq \frac{1}{2e} \cdot L \cdot n^{L+1}$.*

Proof. The bulk of the work has already been done by Lemma 43 from [56]. However, that lemma used r^* whereas our function r has value 0 when the milestones are equal. Here we bridge that gap

$$
\begin{aligned}
\sum_{g_{\mathbf{y},b_2} \in \mathcal{X}} r(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) &\geq \sum_{g_{\mathbf{y},b_2} \in \mathcal{X}} r^*(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) - \sum_{\substack{g_{\mathbf{y},b_2} \in \mathcal{X} \\ \mathbf{y}=\mathbf{x}}} r^*(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) && \text{(By definition of } r.\text{)} \\
&\geq \frac{1}{2e} \cdot (L+1) \cdot n^{L+1} - \sum_{\substack{g_{\mathbf{y},b_2} \in \mathcal{X} \\ \mathbf{y}=\mathbf{x}}} r^*(g_{\mathbf{x},b_1}, g_{\mathbf{y},b_2}) && \text{(By Lemma 43)}
\end{aligned}
$$

$$\geq \frac{1}{2\mathrm{e}} \cdot (L+1) \cdot n^{L+1} - n^{L+1}$$

(By definition of r^*, since exactly one $\mathbf{y} = \mathbf{x}$.)

$$\geq \frac{1}{2\mathrm{e}} \cdot L \cdot n^{L+1}$$

□

3.5.1 Theorems from prior work

Definition 3.5.1 (Valid function [56])**.** *Let $\mathbf{x} = (x_0, \ldots, x_\ell)$ be a walk in G. A function $f : V \to \mathbb{R}$ is valid with respect to the walk $\mathbf{x}$ if it satisfies the next conditions:*

1. *For all $u, v \in \mathbf{x}$, if $\max\{\mathrm{i} \in \{0, \ldots, \ell\} \mid v = x_\mathrm{i}\} < \max\{\mathrm{i} \in \{0, \ldots, \ell\} \mid u = x_\mathrm{i}\}$, then $f(v) > f(u)$. In other words, as one walks along the walk $\mathbf{x}$ starting from x_0 until x_ℓ, if the last time the vertex v appears is before the last time that vertex u appears, then $f(v) > f(u)$.*

2. *For all $v \in V \setminus \mathbf{x}$, we have $f(v) = dist(x_0, v) > 0$.*

3. *$f(x_\mathrm{i}) \leq 0$ for all $\mathrm{i} \in \{0, \ldots, \ell\}$.*

Lemma 41 ([56], lemma 6)**.** *If a function f is valid for a walk $W = w_1, \ldots, w_s$, then f has a unique local minimum at w_s.*

Lemma 42 ([56], lemma 7)**.** *Every function $f_\mathbf{x}$ is valid for the walk induced by $\mathbf{x}$.*

Lemma 43 ([56], lemma 14)**.** *If $F_1 \in \mathcal{X}$ is good, then $\sum_{F_2 \in \mathcal{X}} r^*(F_1, F_2) \geq \frac{1}{2\mathrm{e}} \cdot (L+1) \cdot n^{L+1}$.*

Lemma 44 ([56], corollary D.1)**.** *If G has expansion β and is d-regular where d and β are constant, then G has vertex congestion at most $O(n \log(n))$.*

4. FROM SEARCH IN ROUNDS TO A DUALITY GAP IN YAO'S LEMMA

This chapter is based on my paper "Searching, Sorting, and Cake Cutting in Rounds", which can be found at https://arxiv.org/abs/2012.00738.

4.1 Introduction

We explore the randomized and distributional query complexity of search problems in the expected cost setting. This leads us to the discovery of a substantial gap between the randomized and distributional query complexity of a natural function induced by a search problem.

To make these concepts more precise, consider a function $f : \mathcal{X}_n \to \mathcal{Y}_m$, where $\mathcal{X}_n \subseteq \{0,1\}^n$ and $\mathcal{Y}_m \subseteq \{0,1\}^m$ with $m, n \in \mathbb{N}$. Given as input a bit vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}_n$, an algorithm can query a location j in $\mathbf{x}$ and receive the bit x_{j} in one step. The goal is to compute $f(\mathbf{x})$ with as few queries as possible.

The randomized and distributional complexity [4] of computing the function f are defined as follows. The *randomized complexity with error* δ, denoted $\mathcal{R}_\delta(f)$, is the *expected* number of queries issued on the worst-case input of an optimal randomized algorithm that computes f with an error probability of at most $\delta \in [0,1]$ on each input. See Section 4.1.1 for precise definitions.

When the input x is drawn from a distribution Ψ, a deterministic algorithm $\mathcal{A}$ (not necessarily correct on all inputs) has expected number of queries $cost(\mathcal{A}, \Psi)$ and error probability $\text{e}(\mathcal{A}, \Psi)$. Let $\mathcal{A}_{\Psi,\delta}$ be an algorithm with error probability δ and minimum expected cost for distribution Ψ. The *distributional complexity with error* δ, denoted $\mathcal{D}_\delta(f)$, represents the *expected* number of queries made on the worst case distribution Ψ by the best algorithm for it: $\mathcal{D}_\delta(f) = \sup_\Psi \{cost(\mathcal{A}, \Psi)\}$.

For error probability $\delta = 0$, von Neumann's minimax theorem [57] gives $\mathcal{R}_0(f) = \mathcal{D}_0(f)$. Clearly, we also have $\mathcal{R}_1(f) = \mathcal{D}_1(f) = 0$. For all $\delta \in [0, 1/2]$, [4] showed $\mathcal{R}_\delta(f) \geq 1/2 \cdot \mathcal{D}_{2\delta}(f)$. [58] showed an inequality in the other direction: $\mathcal{R}_\delta(f) \leq 2\mathcal{D}_{\delta/2}(f)$ for all $\delta \in [0, 1]$. [58] also showed that $\mathcal{R}_\delta(f) \leq \mathcal{D}_\delta(f)$ and observed that there can be a difference of additive 1 between $\mathcal{R}_\delta(f)$ and $\mathcal{D}_\delta(f)$ for $\delta \in [1/4, 1/2]$ when the function is $f : \{0, 1\} \to \{0, 1\}$ with $f(x) = x$.

While the expected query complexity was the focus of Yao's seminal paper [4], it has been understudied since then. Most literature has focused on the worst-case cost setting, where the randomized complexity is defined as the worst case cost incurred by the best algorithm with error probability δ and for this setting there is no gap between the two complexities [59]. In recent years, there has been renewed interest in the expected cost setting [60], as it has important applications in complexity such as the randomized composition of functions.

Our work contributes to this area by showing that a natural search problem has a large gap between the randomized and distributional complexity in the expected cost setting. Specifically, we consider a natural function u_n induced by the problem of finding an element z in an unordered array of size n. We show that for each $\delta \in (0, 1)$,

$$\lim_{n\to\infty} \frac{\mathcal{D}_\delta(u_n)}{\mathcal{R}_\delta(u_n)} = 1 + \delta \qquad \text{and} \qquad \lim_{n\to\infty} \mathcal{D}_\delta(u_n) - \mathcal{R}_\delta(u_n) = \infty\,.$$

To the best of our knowledge, this is the first example demonstrating a substantial gap. In fact, [60] asked whether there exist constants $c, d \geq 0$ such that $\mathcal{D}_\delta(f) \leq c \cdot \mathcal{R}_\delta(f) + d$ for each partial function f and $\delta > 0$. Our results show that the two complexities can be substantially different, in particular implying that if such constants c and d exist, it must be the case that $c \geq 2$.

Connections to Cake Cutting and Rounds of Interaction:

The catalyst for our findings is a cake cutting problem that we believe is of independent interest. Suppose we are given a cake represented as the interval $[0, 1]$ and n players, each with an additive valuation over the cake induced by a private value density function. The task is to compute a fair allocation using at most k rounds of interaction with the players. Each round of interaction i consists of a batch i of queries issued simultaneously. Queries in batch i can depend on the responses to queries from rounds $\mathrm{j} < \mathrm{i}$ but not to queries from rounds $\ell \geq \mathrm{i}$. When $k = 1$, all the communication between the algorithm computing the allocation and the players takes place in one simultaneous exchange, while $k = \infty$ represents the fully adaptive setting, where the algorithm issues one query at a time (see [61]).

We design an efficient protocol for proportional cake cutting in rounds, finding that this fair division problem is equivalent to sorting with rank queries in rounds, where a rank query has the form "Is $\mathrm{rank}(x_{\mathrm{i}}) \leq \mathrm{j}$?". A lower bound for sorting with rank queries in rounds was given in [62], while the first connection to proportional cake cutting was implicitly made in [63].

Inspired by the rank query model, we then consider two fundamental search problems that are implicit in sorting: ordered and unordered search. In unordered search, we get an array $\mathbf{x} = (x_1, \ldots, x_n)$ and an element z promised to be in $\mathbf{x}$. The size n is known, but z and the elements of $\mathbf{x}$ are not and cannot be accessed directly. Instead, we have access to an oracle $\mathcal{O}_z$ that receives queries of the form: $\mathcal{O}_z(\mathrm{i})$ ="How is z compared to the element at location i?", answering "$=$" or "$\neq$". The goal is to find the location of z with success probability at least $p \in (0, 1]$ using at most k rounds of interaction with the oracle.

In ordered search, the setting is the same with the difference that (1) the array $\mathbf{x} = (x_1, \ldots, x_n)$ is promised to be sorted and (2) the answer given by the oracle is one of "$<$", "$=$", or "$>$".

4.1.1 Our results

Here we summarize our results after establishing the notation necessary for stating them.

Notation.

For $m, n \in \mathbb{N}$, we consider functions of the form $f : \mathcal{X}_n \to \mathcal{Y}_m$, where $\mathcal{X}_n \subseteq \{0,1\}^n$ and $\mathcal{Y}_m \subseteq \{0,1\}^m$. For promise problems, as in our setting, the set $\mathcal{X}_n$ is a strict subset of $\{0,1\}^n$.

For each $x \in \{0,1\}^n$ and randomized algorithm R for computing f, the error probability of R on input $x \in \mathcal{X}_n$ is

$$\mathrm{err}_f(R, x) = \Pr\left[R(x) \neq f(x)\right] \qquad \forall x \in \mathcal{X}_n,$$

where $R(x)$ is the output of the algorithm and can be the empty string. For the functions we consider, the empty string is never the right answer.

For each $\delta \in [0,1]$, we consider the randomized complexity $\mathcal{R}_\delta(f)$ with error at most δ and the distributional complexity $\mathcal{D}_\delta(f)$ with error at most δ, formally defined as

$$\mathcal{R}_\delta(f) = \inf_{R \in R(f,\delta)} \max_{x \in \mathcal{X}_n} cost(R, x) \qquad \text{and} \qquad \mathcal{D}_\delta(f) = \sup_{\mu} \inf_{D \in D(f,\delta,\mu)} cost(D, \mu), \tag{4.1}$$

where

- $R(f, \delta)$ is the set of randomized algorithms R such that $\mathrm{err}_f(R, x) \leq \delta$ for all $x \in \mathcal{X}_n$.
- μ is a distribution over strings in $\mathcal{X}_n$; that is, $\sum_{x \in \mathcal{X}_n} \mu(x) = 1$.
- $D(f, \delta, \mu)$ is the set of deterministic algorithms D with $\mathbb{E}_{x \sim \mu}[\mathrm{err}_f(D, x)] \leq \delta$.
- $cost(R, x)$ is the expected number of queries issued by a randomized algorithm R on input x.
- $cost(D, \mu)$ represents the expected number of queries issued by a deterministic algorithm D when given as input a string x drawn from the distribution μ.

Unordered Search

The unordered search problem is formally defined as follows.

Definition 4.1.1 (Unordered search). *The input is a bit vector* $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ *with the promise that exactly one bit is* 1*. The vector can be accessed via oracle queries of the form: "Is the* i*-th bit equal to* 1*?". The answer to a query is "Yes' or "No".*

The task is to find the location of the hidden bit in at most $k \in \mathbb{N}$ *rounds of interaction with the oracle. An index must be queried before getting returned as the solution* [1]*.*

Let $\textit{unordered}_{n,k}$ denote the unordered search problem on an input vector of length n in k rounds. We have the following bounds for the randomized complexity of this problem.

Theorem 4.1.1 (Unordered search, randomized algorithms on worst case input). *For all* $k, n \in \mathbb{N}^*$ *and* $p \in [0,1]$*, we have:* $np\Big(\frac{k+1}{2k}\Big) \leq \mathcal{R}_{1-p}(\text{unordered}_{n,k}) \leq np\Big(\frac{k+1}{2k}\Big) + p + p/n\,.$

We also analyze the distributional complexity. We say the input is drawn from distribution $\Psi = (\Psi_1, \ldots, \Psi_n)$ if the hidden bit is at location i with probability Ψ_{i}, where $\Psi_{\mathrm{j}} \geq 0$ for all $\mathrm{j} \in [n]$ and $\sum_{\mathrm{j}=1}^{n} \Psi_{\mathrm{j}} = 1$. The distributional complexity is bounded as follows.

Theorem 4.1.2 (Unordered search, deterministic algorithms on worst case input distribution). *For all* $k, n \in \mathbb{N}^*$ *and* $p \in [0,1]$*:*

$$np\Big(1 - \frac{k-1}{2k} \cdot p\Big) \leq \mathcal{D}_{1-p}(\text{unordered}_{n,k}) \leq np\Big(1 - \frac{k-1}{2k} \cdot p\Big) + 1 + p + 2/n\,. \tag{4.2}$$

The uniform distribution is the worst case for unordered search.

Combining Theorem 4.1.1 and 4.1.2, we obtain that for each $p \in (0,1)$, there exists $n_0 = n_0(p) \in \mathbb{N}$ such that for all $k, n \in \mathbb{N}$ with $n \geq n_0$, the multiplicative gap between the

[1]↑This requirement is benign as it only makes a difference of ± 1 in the bounds.

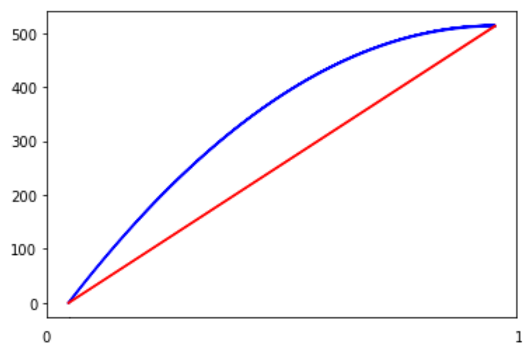


Figure 4.1. The query complexity of fully adaptive unordered search for $n = 2^{10}$ elements, with success probability p ranging from 0 to 1. The X axis is for the success probability p, while the Y axis is for the expected number of queries. The randomized complexity is plotted in red (both upper and lower bounds and they coincide) and similarly the distributional complexity in blue.

distributional and randomized complexity of unordered search in k rounds with success probability p is

$$\frac{\mathcal{D}_{1-p}(\text{unordered}_{n,k})}{\mathcal{R}_{1-p}(\text{unordered}_{n,k})} = 1 + \frac{(k-1)(1-p)}{k+1} \pm o(1)\,. \tag{4.3}$$

The gap in (4.3) grows from 1 to $\approx (2-p)$ as the number of rounds grows from $k = 1$ to $k = n$.

Fully adaptive unordered search.

By taking $k = n$, the bounds in Theorem 4.1.1 and 4.1.2 characterize the query complexity of the fully adaptive unordered search problem, denoted unordered_n.

Corollary 8 (Fully adaptive unordered search)**.** *Let $n \in \mathbb{N}^*$ and $p \in [0,1]$. The randomized and distributional complexity of fully adaptive unordered search with success probability p are:*

- $np\left(\frac{n+1}{2n}\right) \leq \mathcal{R}_{1-p}(\text{unordered}_n) \leq np\left(\frac{n+1}{2n}\right) + p + p/n\,.$

- $np\Big(1 - \frac{n-1}{2n} \cdot p\Big) \leq \mathcal{D}_{1-p}(\text{unordered}_n) \leq np\Big(1 - \frac{n-1}{2n} \cdot p\Big) + 1 + p + 2/n\,.$

The randomized complexity is roughly $np/2$ and the distributional complexity roughly $np\Big(1-\frac{p}{2}\Big)$.

Corollary 9 (Multiplicative and additive gap for fully adaptive unordered search)**.** *For each success probability $p \in (0,1]$, we have*

$$\lim_{n\to\infty} \frac{\mathcal{D}_{1-p}(\text{unordered}_n)}{\mathcal{R}_{1-p}(\text{unordered}_n)} = 2 - p \quad \textit{and} \quad \lim_{n\to\infty} \mathcal{D}_{1-p}(\text{unordered}_n) - \mathcal{R}_{1-p}(\text{unordered}_n) = \infty\,. \tag{4.4}$$

Ordered Search

The ordered search problem is formally defined next. The difference from unordered search is that the array is sorted and the oracle gives feedback about the direction in which to continue the search in case of a "No" answer.

Definition 4.1.2 (Ordered search)**.** *The input is a bit vector $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ with the promise that exactly one bit is set to 1. The vector can be accessed via oracle queries of the form: "Is the* i*-th bit equal to* 1*?". The answer to a query is: "Yes', "No, go left", or "No, go right".*

The task is to find the location of the hidden bit using at most $k \in \mathbb{N}$ rounds of interaction with the oracle. An index must be queried before getting returned as the solution.

Let $\text{ordered}_{n,k}$ denote the ordered search problem on an input vector of length n in k rounds. For ordered search the number of rounds need not be larger than $\lceil \log_2 n \rceil$, since binary search is an optimal fully adaptive algorithm for success probability 1. We have the following bounds.

Theorem 4.1.3 (Ordered search, randomized and distributional complexity)**.** *For all $k, n \in \mathbb{N}^*$ and $p \in [0,1]$, we have:*

$$kpn^{\frac{1}{k}} - 2pk \leq \mathcal{R}_{1-p}(\text{ordered}_{n,k}) \leq \mathcal{D}_{1-p}(\text{ordered}_{n,k}) \leq k\lceil pn^{\frac{1}{k}} \rceil + 2\,.$$

Moreover, $np \leq R_{1-p}(\text{ordered}_{n,k}) \leq D_{1-p}(\text{ordered}_{n,k}) \leq \lceil np \rceil$. The uniform distribution is the worst case for ordered search.

Theorem 4.1.3 shows that for ordered search in constant rounds, there is essentially no gap between the randomized and distributional complexity.

Cake Cutting in Rounds and Sorting with Rank Queries

We consider the cake cutting problem of finding a proportional allocation with contiguous pieces in k rounds. The cake is the interval $[0,1]$ and the goal is to divide it among n players with private additive valuations. A *proportional* allocation, where each player gets a piece worth $1/n$ of the total cake according to the player's own valuation, always exists and can be computed in the standard (RW) query model for cake cutting.

We establish a connection between proportional cake cutting with contiguous pieces and sorting in rounds in the *rank query model*. In the latter, we have oracle access to a list x of n elements that we cannot inspect directly. The oracle accepts rank queries of the form "How is $rank(x_{\mathrm{i}})$ compared to j?", where the answer is "<", "=", "*or* >"[2].

Theorem 4.1.4. *(Informal). For all $k, n \in \mathbb{N}^*$, the following problems are equivalent:*

- *computing a proportional cake allocation with contiguous pieces for n agents in the standard (RW) query model*
- *sorting a vector with n elements using rank queries.*

[2]↑Equivalently, the queries are "Is $rank(x_{\mathrm{i}}) \leq$ j?", where the answer is *Yes* or *No*.

The randomized query complexity of both problems (for constant success probability) is $\Theta\left(k \cdot n^{1+\frac{1}{k}}\right)$.

We prove Theorem 4.1.4 in Section 4.6. We design an optimal protocol for proportional cake cutting in k rounds. En route, we re-examine the implicit reduction from sorting with rank queries to proportional cake cutting as presented in Woeginger (2007), and make it completely precise.

[62] gave a lower bound of $\Omega(kn^{1+\frac{1}{k}})$ for sorting a vector of n elements with rank queries in $k \leq \log n$ rounds. We also show a slightly improved deterministic lower bound for sorting with rank queries that has a simpler proof.

Finally, to highlight the connection to Ordered Search, we point out that an operation implicit in sorting with rank queries is *Locate*: given a vector $x = (x_1, \ldots, x_n)$ and an element x_i, find its rank via rank queries. Locate with rank queries is equivalent to the ordered search problem.

4.1.2 Related work

Parallel complexity.

Parallel complexity is a fundamental concept with a long history in areas such as sorting and optimization; see, e.g. [64] on the parallel complexity of optimization and more recent results on submodular optimization [65]. An overview on parallel sorting algorithms is given in the book [66] and many works on sorting and selection in rounds [61, 67–71], aiming to understand the tradeoffs between the number of rounds of interaction and the query complexity.

[61] initiated the study of parallelism using the number of comparisons as a complexity measure and showed that p processor parallelism can offer speedups of at least $O\left(\frac{p}{\log\log p}\right)$ for problems such as sorting and finding the maximum of a list of $n > p$ elements. The connection to the problem of sorting in rounds is straight-forwards since one parallel step of

the p processors (e.g. p comparisons performed in parallel) can be viewed as one round of computations.

[72] showed that $O\Big(n^{\frac{3\cdot 2^{k-1}-1}{2^k-1}}\log n\Big)$ comparisons suffice to sort an array in k rounds. [73] showed a bound of $O(n^{3/2}\log n)$ for two rounds. [67] made a connection between expander graphs and sorting and proved that $O\Big(n^{1+\frac{1}{k}}(\log n)^{2-\frac{2}{k}}\Big)$ comparisons are enough. This was improved to $O\Big(n^{3/2}\frac{\log n}{\sqrt{\log\log n}}\Big)$ in [74], which also showed that $\Omega(n^{1+1/k}(\log n)^{1/k})$ comparisons are needed.

[68] generalized the latter upper bound to $O\Big(n^{1+1/k}\frac{(\log n)^{2-2/k}}{(\log\log n)^{1-1/k}}\Big)$ for k rounds. The best upper bound known to us is due to [70], which obtained a k-rounds algorithm that performs $O\Big(n^{1+1/k+o(1)}\Big)$ comparisons. For randomized algorithms, [75] obtained an algorithm that runs in k rounds and issues $O\Big(n^{1+1/k}\Big)$ queries, thus demonstrating that randomization helps in the comparison model. Local search in rounds was considered in [76].

Randomized complexity.

The expected cost setting that we consider is the one studied in [4]. However, most of the literature since then has focused on the worst case setting, where the cost of an algorithm is the worst case cost among all possible inputs and coin-flips (for randomized algorithms). In more detail, consider a function $f : \mathcal{X}_n \to \mathcal{Y}_m$, with $m, n \in \mathbb{N}$ and $\mathcal{X}_n \subseteq \{0,1\}^n$ and $\mathcal{Y}_m \subseteq \{0,1\}^m$.

The worst-case randomized complexity for error δ, denoted $\widehat{\mathcal{R}}_\delta(f)$, is defined as the maximum number of queries issued by a randomized algorithm R, where the maximum is taken over all inputs $x \in \mathcal{X}_n$ and coin-tosses, and R has the property $\mathrm{err}_f(R,x) \le \delta$ for all $x \in \mathcal{X}_n$. The worst-case distributional complexity for error δ, denoted $\widehat{\mathcal{D}}_\delta(f)$, is the maximum number of queries issued by an optimal deterministic algorithm $\mathcal{A}$ that computes f with error probability δ when the input is drawn from a worst case input distribution Ψ, where the optimality of $\mathcal{A}$ is with respect to Ψ. It is known that: $\widehat{\mathcal{R}}_\delta(f) = \widehat{\mathcal{D}}_\delta(f)$ [59].

In recent work, [60] also focus on the expected cost setting and analyze the gap between the expected query complexity of randomized algorithms on worst case input and the expected query complexity of randomized algorithms on a worst case input distribution, in the regime where the error probability is $\delta \approx 1/2$.

Group testing.

In fault detection, the goal is to identify all the defective items from a finite set items via a minimum number of tests. More formally, there is a universe of $\mathcal{U}$ of n items, d of which are defective. Each test is executed on a subset $S \subseteq \mathcal{U}$ and says whether S is contaminated (i.e. has at least one defective item) or pure (i.e. none of the items in S are defective). Questions include how many tests are needed to identify all the defective items and how many stages are needed, where the tests performed in round $k+1$ can depend on the outcome of the tests in round k. An example of group testing is to identify which people from a set are infected with a virus, given access to any combination of individual blood samples; combining their samples allows detection using a smaller number of tests compared to checking each sample individually.

The group testing problem was posed in [77] and a lower bound of $\Omega\left(d^2 \frac{\log n}{\log d}\right)$ for the number of tests required in the one round setting was given in [78]. One round group testing algorithms with an upper bound of $O(d^2 \log n)$ on the number of tests were designed in [79–82]. Two round testing algorithms were studied in [83, 84]. The setting where the number of rounds is allowed is given by some parameter r and the number of defective items is not known in advance was studied in [85–88]; see [89] for a survey.

Fair division.

The cake cutting model was introduced in [90] to study the allocation of a heterogeneous resource among agents with complex preferences. Cake cutting was studied in mathematics, political science, economics [91–93], and computer science [94–96]. There is a hierarchy of fairness notions such as proportionality, envy-freeness (where no player prefers the piece of another player), equitability, and necklace splitting [97], with special cases such as consensus halving and perfect partitions. See [98, 99] for surveys.

Cake cutting protocols are often studied in the Robertson-Webb [63] query model, where a mediator asks the players queries until it has enough information to output a fair division. [100] devise an algorithm for computing a proportional allocation with connected pieces that asks $O(n \log n)$ queries, with matching lower bounds due to [63] and [101].

For the query complexity of exact envy-free cake cutting (possibly with disconnected pieces), a lower bound of $\Omega(n^2)$ was given by [102] and an upper bound of $O\left(n^{n^{n^{n^{n^{n}}}}}\right)$ by [103]. [104] found a simpler algorithm for 4 agents. An upper bound on the query complexity of equitability was given by [105] and a lower bound by [106]. [107] analyzed the query complexity of envy-freeness, perfect, and equitable partitions with minimum number of cuts.

The issue of rounds in cake cutting was studied in [108], where the goal is to bound the communication complexity of protocols depending on the fairness notion. The query complexity of proportional cake cutting with different entitlements was studied by [109]. The query complexity of consensus halving was studied in [110] for monotone valuations, with an appropriate generalization of the Robertson-Webb query model. The query complexity of cake cutting in one round, i.e. in the simultaneous setting, was studied in [111].

Many other works analyzed the complexity of fair division in models such as cake cutting, multiple divisible goods, and indivisible goods. The complexity of cake cutting was studied, e.g., in [112–121]. Indivisible goods were studied, e.g., in [122] for their query complexity and in [123, 124] for algorithms. Cake cutting with separation was studied in [125], fair division of a graph or graphical cake cutting in [126, 127], multi-layered cakes in [128], fair cutting in practice in [129], and cake cutting where some parts are good and others bad in [130] and when the whole cake is a "bad" in [131]. Branch-choice protocols were developed and analyzed in [132] as a simpler yet expressive alternative for GCC protocols from [133]. A body of work analyzed truthful cake cutting both in the standard (Robertson-Webb) query model [134, 135] and in the direct revelation model [136–139].

4.2 Ordered search

In this section we focus on ordered search and prove Theorem 4.1.3. The omitted proofs of this section can be found in Section 4.4.

4.2.1 Deterministic ordered search algorithm on worst case input

We first design a deterministic algorithm D^o for ordered search that always succeeds and asks at most $k\lceil n^{\frac{1}{k}}\rceil$ queries on each input.

Proposition 4.2.1. *For each $n \in \mathbb{N}^*$ and $k \in [\lceil \log n \rceil]$, there is a deterministic k-round algorithm for ordered search that succeeds on every input and asks at most $k\lceil n^{\frac{1}{k}}\rceil$ queries in the worst case.*

The algorithm D^o that achieves this upper bound issues $n^{\frac{1}{k}}$ queries in the first round, which are as equally spaced as possible, partitioning the array in $n^{\frac{k-1}{k}}$ blocks. If the element is found at one of the locations queried in the first round, then D^o returns it and halts. Otherwise, D^o recurses on the block that contains the solution in the remaining $k-1$ rounds.

4.2.2 Randomized ordered search algorithm on worst case input

Using D^o, for each $p \in (0, 1]$, we design a randomized algorithm R^o that succeeds with probability at least p and asks at most $pk\lceil n^{\frac{k}{k}}\rceil$ queries in expectation.

Proposition 4.2.2. *Let $p \in (0, 1]$ and $k, n \in \mathbb{N}^*$. Then $\mathcal{R}_{1-p}(unordered_{n,k}) \leq pk\lceil n^{\frac{1}{k}}\rceil$.*

The randomized algorithm R^o has an all-or-nothing structure:

- with probability $1 - p$, do nothing (i.e. output the empty string);
- with probability p, run the deterministic algorithm D^o from Proposition 4.2.1.

4.2.3 Deterministic ordered search algorithm on worst case input distribution

Next we upper bound the distributional complexity of ordered search.

Proposition 4.2.3. *Let $p \in (0,1]$ and $k, n \in \mathbb{N}^*$. Then $\mathcal{D}_{1-p}(\text{ordered}_{n,k}) \leq k\lceil pn^{\frac{1}{k}}\rceil + 2$. Moreover, $\mathcal{D}_{1-p}(\text{ordered}_{n,1}) \leq \lceil np\rceil$.*

Proof sketch. We include the proof sketch, while the formal details can be found in Section 4.4.

Using D^o and R^o, we show how for each $p \in (0,1]$, if the input is drawn from an arbitrary distribution $\Psi = (\Psi_1, \ldots, \Psi_n)$, one can design a deterministic algorithm D^o_Ψ that asks at most $k\lceil pn^{\frac{1}{k}}\rceil + 2$ queries in expectation and succeeds with probability at least p. The distribution-dependent deterministic algorithm D^o_Ψ will simulate the execution of R^o using the following steps.

Step 1. Given Ψ, define probability density $v : [0,1] \to \mathbb{R}$ by $v(x) = n\Psi_{\mathrm{i}} \quad \forall \mathrm{i} \in [n]\ \forall x \in \left[\frac{\mathrm{i}-1}{n}, \frac{\mathrm{i}}{n}\right]$.

Let $\mathcal{C}$ denote the circle obtained by bending the interval $[0,1]$ so that the point 0 coincides with 1. A fixed point theorem (Lemma 67) ensures there is a point $c \in [0,1]$ such that the interval $[c, c+p]$ on the circle $\mathcal{C}$ has probability mass p (and length p). That is:

(a) $\int_c^{c+p} v(x)\,dx = p$, where $0 \leq c \leq 1-p$; or

(b) $\int_0^c v(x)\,dx + \int_{c+1-p}^1 v(x)\,dx = p$, where $1-p < c < 1$.

Step 2. The points c and $c+p$ can be mapped to indices $\mathrm{i} \in [n]$ and $\mathrm{j} \in [n]$, respectively, so that one of the following conditions holds:

- $\mathbf{y}_\Psi = [x_{\mathrm{i}}, \ldots, x_{\mathrm{j}}]$ has length $\approx np$ and probability mass $\sum_{\ell=\mathrm{i}}^{\mathrm{j}} \Psi_\ell \approx p$; or
- $\mathbf{y}_\Psi = [x_1, \ldots, x_{\mathrm{i}}, x_{\mathrm{j}}, \ldots, x_n]$ has length $\approx np$ and probability mass $\sum_{\ell=1}^{\mathrm{i}} \Psi_\ell + \sum_{\ell=\mathrm{j}}^{n} \Psi_\ell \approx p$.

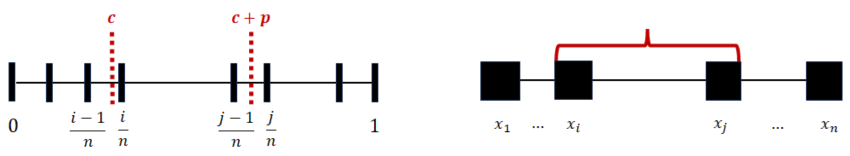


Figure 4.2. Illustration for case (a) in step 1. Given $\Psi = (\Psi_1, \ldots, \Psi_n)$, define $v : [0,1] \to \mathbb{R}_{\geq 0}$ by $v(x) = n \cdot \Psi_\ell$ for all $\ell \in [n]$ and $x \in [(\ell-1)/n, \ell/n]$. The left figure shows the point c with $\int_c^{c+p} v(x)\,dx = p$. The right figure shows the queried sub-array $\mathbf{y}_\Psi = [x_1, \ldots, x_{\mathrm{i}}, x_{\mathrm{j}}, \ldots, x_n]$, of length $\approx np$ and probability mass $\approx p$.

Step 3. In the first round, algorithm D_Ψ^o queries locations i and j, as well as $\approx pn^{\frac{1}{k}}$ other equally spaced locations in the sub-array $\mathbf{y}_\Psi$. These queries create approximately $pn^{\frac{1}{k}}$ blocks of size roughly $\left(\frac{np}{pn^{1/k}}\right) \approx n^{\frac{k-1}{k}}$ each. Then:

- If the first round queries reveal the hidden element is not in $\mathbf{y}_\Psi$, then D_Ψ^o gives up right away (i.e. outputs the empty string).
- Else, if the element is found at a location queried in round 1, then D_Ψ^o returns it and halts.
- Else, in the remaining $k-1$ rounds, run D^o on the block identified to contain the element.

Expected number of queries of D_Ψ^o.

The block identified at the end of the first round has length $\approx n^{\frac{k-1}{k}}$. Moreover, D_Ψ^o continues to the second round with probability $\approx p$. Thus the success probability is roughly p and the total expected number of queries is approximately

$$\left(pn^{\frac{1}{k}} + 2\right) + p \cdot (k-1)\left(n^{\frac{k-1}{k}}\right)^{\frac{1}{k-1}} = pkn^{\frac{1}{k}} + 2\,.$$

In summary, the deterministic algorithm D_Ψ^o is able to generate an event of probability $\approx p$ via the first round queries while also pre-partitioning a relevant sub-array. If the event does not take place, then D^o_Ψ gives up. Otherwise, it runs an optimal deterministic $(k-1)$-round algorithm on the block identified via the first round queries. This strategy enables D^o_Ψ to simulate the all-or-nothing structure of the optimal randomized algorithm and catch up with it fast enough so that the query complexity remains essentially the same. $\square$

4.2.4 Lower bounds for ordered search

We prove the next lower bound for randomized algorithms that succeed with probability p.

Proposition 4.2.4. *Let $k, n \in \mathbb{N}^*$ and $p \in (0,1]$. Then $\mathcal{R}_{1-p}(\text{ordered}_{n,k}) \geq pkn^{\frac{1}{k}} - 2pk$ for $k \geq 2$ and $\mathcal{R}_{1-p}(\text{ordered}_{n,1}) \geq np$.*

This lower bound has the same leading term as the upper bound achieved by $\mathcal{D}_\Psi$, thus showing that the randomized and distributional complexity have the same order. The uniform distribution is the worst case.

We prove Proposition 4.2.4 by induction on the number k of rounds. The induction step requires showing polynomial inequalities, where the polynomials involved have high degrees that are themselves functions of k. For $k \geq 4$, the roots of such polynomials cannot be found by a formula. To overcome this, we use delicate approximations of the polynomials by simpler ones that are more amenable to study yet close-enough to the original polynomials to yield the required inequalities.

Finally, we obtain the proof of Theorem 4.1.3 by combining the upper bound from Proposition 4.2.3 and the lower bound from Proposition 4.2.4.

4.3 Unordered search

In this section we analyze the unordered search problem and prove Theorems 4.1.1 and 4.1.2, which quantify the randomized and distributional complexity of unordered search algorithms, respectively. Theorem 4.1.1 will follow from Propositions 4.3.1 and 4.3.3 stated next. Theorem 4.1.2 will follow from Propositions 4.3.2 and 4.3.4. The omitted proofs of this section are in Section 4.5.

4.3.1 Deterministic and randomized algorithms for unordered search on a worst case input

The maximum number of rounds for unordered search is n. Since with each location queried the only information an algorithm receives is whether the element is at that location or not, a k-round deterministic unordered search algorithm that succeeds on every input cannot do better than querying roughly n/k queries in each round until finding the element. This gives a total of n queries in the worst case. However, randomized algorithms can do better by querying locations uniformly at random.

Proposition 4.3.1. *Let $p \in (0,1]$ and $k, n \in \mathbb{N}^*$. Then $\mathcal{R}_{1-p}(\mathit{unordered}_{n,k}) \le np \cdot \frac{k+1}{2k} + p + \frac{p}{n}$.*

The optimal randomized algorithm given by Proposition 4.3.1 has an all-or-nothing structure:

(i) with probability $1 - p$, do nothing;

(ii) with probability p, select a uniform random permutation $\boldsymbol{\pi}$ over $[n]$. For all $\mathrm{j} \in [k]$, define $S_{\mathrm{j}} = \{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_{m_{\mathrm{j}}}\}$, where $m_{\mathrm{j}} = \lceil n\mathrm{j}/k \rceil$. In each round j, query the locations of S_{j} that have not been queried in the previous $\mathrm{j} - 1$ rounds. Once the element is found, return it and halt.

4.3.2 Deterministic algorithms for unordered search on random input

We have the following upper bound on the distributional complexity of unordered search.

Proposition 4.3.2. *Let $p \in (0,1]$ and $k, n \in \mathbb{N}^*$. Then*

$$\mathcal{D}_{1-p}(\mathit{unordered}_{n,k}) \le np\Big(1 - \frac{k-1}{2k} \cdot p\Big) + 1 + p + \frac{2}{n}\,.$$

Since the unordered search problem has less structure than ordered search, a deterministic algorithm receiving an element drawn from some distribution Ψ will no longer be able to extract enough randomness from the answers to the first round queries to simulate the optimal randomized algorithm. Instead, the optimal deterministic algorithm will establish in advance a fixed set of np locations and query those in the same manner as step (ii) of the optimal randomized algorithm.

However, since the search space becomes smaller as an algorithm checks more locations, the fact that the deterministic algorithm is forced to stop after at most np queries regardless of whether it found the element or not (to avoid exceeding the optimal expected query bound), is a source of inefficiency. This is the main reason for which a deterministic algorithm receiving a random input cannot do as well as the optimal randomized algorithm that decided in advance to either do nothing or search all the way until finding the solution.

4.3.3 Lower bounds for unordered search

Finally, we lower bound the randomized and distributional complexity of unordered search.

Proposition 4.3.3. *Let* $p \in (0,1]$ *and* $k, n \in \mathbb{N}^*$. *Then* $\mathcal{R}_{1-p}(\mathit{unordered}_{n,k}) \geq np \cdot \frac{k+1}{2k}$.

Proposition 4.3.4. *Let* $p \in (0,1]$ *and* $k, n \in \mathbb{N}^*$. *Then* $\mathcal{D}_{1-p}(\mathit{unordered}_{n,k}) \geq np\left(1 - \frac{k-1}{2k}p\right)$.

Roadmap to the proof sections

Section 4.4 contains the analysis of ordered search. Section 4.5 contains the analysis for unordered search. Section 4.6 contains the analysis for cake cutting and sorting in rounds. Section 4.7 contains folklore lemmas that we use.

4.4 Ordered search Proofs

In this section we include the omitted proofs for ordered search, which constitute the proof of Theorem 4.1.3.

4.4.1 Ordered search upper bounds

In this section we describe an optimal deterministic algorithm for a worst case input, an optimal randomized algorithm for a worst case input, and an optimal deterministic algorithm for an arbitrary input distribution.

Deterministic algorithms for a worst case input.

The optimal deterministic algorithm for a worst case input is given in the next proposition.

Proposition 4.2.1 (restated). *For each $n \in \mathbb{N}^*$ and $k \in [\lceil \log n \rceil]$, there is a deterministic k-round algorithm for ordered search that succeeds on every input and asks at most $k \lceil n^{\frac{1}{k}} \rceil$ queries in the worst case.*

Proof. We design a k-round algorithm recursively, using induction on k.

Base case: $k = 1$. Let $\mathcal{A}_1$ be the following algorithm:

- Query all the elements of the array simultaneously. Return the correct location based on the results of the queries.

Then $\mathcal{A}_1$ runs in one round, succeeds on every input, and the number of queries is at most n.

Induction hypothesis. For $k \geq 2$, assume there is a $(k-1)$-round algorithm $\mathcal{A}_{k-1}$ that always succeeds and asks at most $(k-1) \cdot \lceil n^{\frac{1}{k-1}} \rceil$ queries on each array of length n.

Induction step. Using the induction hypothesis, we will design a k-round algorithm $\mathcal{A}_k$ with the required properties. For each $s \in [n]$, write $n = s \cdot u_s + v_s$, for $u_s = \lfloor \frac{n}{s} \rfloor$ and $v_s = n \pmod{s}$. Let $\mathcal{A}_k(s)$ be the following algorithm:

(i) In round 1, query locations $\mathrm{i}_1, \ldots, \mathrm{i}_s \in [n]$ with the property that $1 < \mathrm{i}_1 < \ldots < \mathrm{i}_s = n$. Let $\mathrm{i}_0 = 0$. Then these queries create s contiguous blocks $B_1, \ldots, B_s$, such that $B_{\mathrm{j}} = [\mathrm{i}_{\mathrm{j}-1}+1, \mathrm{i}_{\mathrm{j}}]$ for $\mathrm{j} \in [s]$.

   For each $\mathrm{j} \in [s]$, set the size of each block B_{j} to $\lfloor \frac{n}{s} \rfloor$ if $\mathrm{j} \leq s - v_s$ and to $\lceil \frac{n}{s} \rceil$ if $\mathrm{j} > s - v_s$. This uniquely determines indices $\mathrm{i}_1, \ldots, \mathrm{i}_s$.

   If the element searched for is found at one of these s locations, then return that location and halt. Otherwise, identify the index $\ell \in [s]$ for which the block B_ℓ contains the answer.

(ii) Given index ℓ from step (i) such that block $B_\ell = [\mathrm{i}_{\ell-1}+1, \mathrm{i}_\ell]$ contains the answer, we observe that position i_ℓ is the only one from block B_ℓ that has been queried so far. If $\mathrm{i}_\ell - 1 \geq \mathrm{i}_{\ell-1}+1$, let $\widetilde{B}_\ell = [\mathrm{i}_{\ell-1}+1, \mathrm{i}_\ell - 1]$ and run algorithm $\mathcal{A}_{k-1}$ on block $\widetilde{B}_\ell$. Else, halt.

We first show algorithm $\mathcal{A}_k(s)$ is correct for every choice of s, and then obtain $\mathcal{A}_k$ by optimizing s.

Algorithm $\mathcal{A}_k(s)$ is correct if the choice of indices $\mathrm{i}_1, \ldots, \mathrm{i}_s$ is valid. This is the case if the sizes of the blocks $B_1, \ldots, B_s$ sum up to n. We have $\sum_{\mathrm{j}=1}^{s} |B_{\mathrm{j}}| = \lfloor n/s \rfloor \cdot (s - v_s) + \lceil n/s \rceil \cdot v_s$.

(a) If $v_s = 0$ then $\lfloor n/s \rfloor = \lceil n/s \rceil = u_s$, so the sum of block sizes is $\sum\limits_{\mathrm{j}=1}^{s} |B_{\mathrm{j}}| = u_s \cdot (s - v_s) + u_s \cdot v_s = n$.

(b) If $v_s > 0$ then $\lceil n/s \rceil = u_s + 1$, so $\sum_{\mathrm{j}=1}^{s} |B_{\mathrm{j}}| = u_s \cdot (s - v_s) + (u_s + 1) \cdot v_s = u_s \cdot s + v_s = n$.

Combining (a) and (b), we get that the block sizes are valid. Thus $\mathcal{A}_k(s)$ does not skip any indices, so it always finds the element.

Next we argue that there is a choice of s such that by setting $\mathcal{A}_k = \mathcal{A}_k(s)$, we obtain a k-round algorithm that issues at most $k \lceil n^{\frac{1}{k}} \rceil$ queries.

For a fixed $s \in [n]$, the array size at the beginning of round 2 is at most $m(s) = \max_{\mathrm{j} \in [s]} |B_{\mathrm{j}}| - 1$, since the rightmost element of each block B_{j} has been queried in round 1 while the rest of block B_{j} has not been queried. Then $m(s) = \max\left\{\lfloor \frac{n}{s} \rfloor - 1, \lceil \frac{n}{s} \rceil - 1\right\} = \lceil \frac{n}{s} \rceil - 1$.

The total number of queries of algorithm $\mathcal{A}_k(s)$ is at most

$$f(s) = s + (k-1) \cdot \left\lceil m(s)^{\frac{1}{k-1}} \right\rceil = s + (k-1) \cdot \left\lceil \left(\left\lceil \frac{n}{s} \right\rceil - 1 \right)^{\frac{1}{k-1}} \right\rceil . \tag{4.5}$$

Taking $s = \lceil n^{\frac{1}{k}} \rceil$ in (4.5), we get

$$\begin{aligned} f\left(\lceil n^{\frac{1}{k}} \rceil\right) &= \lceil n^{\frac{1}{k}} \rceil + (k-1) \cdot \left\lceil \left(\left\lceil \frac{n}{\lceil n^{\frac{1}{k}} \rceil} \right\rceil - 1 \right)^{\frac{1}{k-1}} \right\rceil \\ &\leq \lceil n^{\frac{1}{k}} \rceil + (k-1) \cdot \left\lceil \left(\left\lceil \frac{n}{n^{\frac{1}{k}}} \right\rceil - 1 \right)^{\frac{1}{k-1}} \right\rceil \\ &\leq \lceil n^{\frac{1}{k}} \rceil + (k-1) \cdot \left\lceil \left(\frac{n}{n^{\frac{1}{k}}} \right)^{\frac{1}{k-1}} \right\rceil \\ &= k \cdot \lceil n^{\frac{1}{k}} \rceil . \end{aligned}$$

Setting $\mathcal{A}_k = \mathcal{A}_k(\lceil n^{\frac{1}{k}} \rceil)$, we obtain a correct k-round algorithm that issues at most $k \cdot \lceil n^{\frac{1}{k}} \rceil$ queries on every array with n elements. This completes the induction step and the proof. □

Randomized algorithms for a worst case input.

Building on the optimal deterministic algorithm for worst case input, we design next an optimal randomized algorithm.

Proposition 4.2.2 (restated). *Let $p \in (0, 1]$ and $k, n \in \mathbb{N}^*$. Then $\mathcal{R}_{1-p}(\mathit{unordered}_{n,k}) \leq pk\lceil n^{\frac{1}{k}} \rceil$.*

Proof. Consider the following randomized algorithm:

- With probability p, run the deterministic algorithm $\mathcal{A}_k$ from Proposition 4.2.1.

- With probability $1-p$, do nothing.

On each input, by Proposition 4.2.1, this algorithm succeeds with probability p and issues at most $pk\lceil n^{\frac{1}{k}}\rceil$ queries in expectation, as required. $\square$

Deterministic algorithms for a random input.

We consider first the case of $k=1$ rounds. With one round, there is no distinction between ordered and unordered search.

Proposition 4.4.1. *Let $p \in (0,1]$ and $n \in \mathbb{N}^*$. Then*

$$\mathcal{D}_{1-p}(\text{unordered}_{n,1}) \leq \lceil np \rceil \;\; \textit{and} \;\; \mathcal{D}_{1-p}(\text{ordered}_{n,1}) \leq \lceil np \rceil \,. \tag{4.6}$$

Proof. Sort the elements of $\mathbf{x}$ in decreasing order by Ψ and let $\boldsymbol{\pi}$ be the permutation obtained, that is, $\Psi_{\pi_1} \geq \ldots \geq \Psi_{\pi_n}$. Let ℓ be the smallest index for which $\sum_{\mathrm{i}=1}^{\ell} \Psi_{\pi_\mathrm{i}} \geq p$. Let $q = \sum_{\mathrm{i}=1}^{\ell} \Psi_{\pi_\mathrm{i}} \geq p$. Consider the following algorithm $\mathcal{A}$:

- Query elements $x_{\pi_1}, \ldots, x_{\pi_\ell}$, i.e. compare each of them with z. If there is $\mathrm{i} \in [\ell]$ such that $z = x_{\pi_\mathrm{i}}$, then return $\boldsymbol{\pi}_\mathrm{i}$.

By choice of ℓ, the success probability of this algorithm is $q \geq p$. The number of queries is ℓ. Let $m = \lceil np \rceil$. Then $(m-1)/n < p \leq m/n$. By Lemma 65, we have $\Psi_{\pi_1} + \ldots + \Psi_{\pi_m} \geq m/n\,$. Since ℓ is the smallest index with $\Psi_{\pi_1} + \ldots + \Psi_{\pi_\ell} \geq p$, it follows that $\ell \leq m = \lceil np \rceil$. $\square$

Using the deterministic algorithm of Proposition 4.2.1 and the randomized algorithm of Proposition 4.2.2, we can now design a deterministic algorithm that is designed to be optimal when the input is drawn from a distribution Ψ.

Proposition 4.2.3 (restated). *Let $p \in (0,1]$ and $k, n \in \mathbb{N}^*$. Then $\mathcal{D}_{1-p}(\text{ordered}_{n,k}) \leq k\lceil pn^{\frac{1}{k}}\rceil + 2$. Moreover, $\mathcal{D}_{1-p}(\text{ordered}_{n,1}) \leq \lceil np \rceil$.*

Proof. The upper bound of $D_{1-p}(\mathit{ordered}_{n,1}) \leq \lceil np \rceil$ for $k = 1$ rounds holds by Proposition 4.4.1. Thus from now on we can assume $k \geq 2$.

At a high level, given input distribution Ψ, the deterministic algorithm for this distribution will consists of two steps:

- First, observe there exists an interval $[\mathrm{i}, \mathrm{j}]$ on the array viewed on the circle (i.e. where index $n+1$ is the same as index 1) that has probability mass roughly pn and length roughly pn as well. Find this interval offline without any queries.
- Second, use the interval identified in the first step to generate an event with probability p, thus simulating the randomized algorithm from Proposition 4.2.1.

Formally, given input distribution Ψ, define a probability density function $v : [0,1] \to \mathbb{R}_{\geq 0}$ by

$$v(x) = n \cdot \Psi_{\mathrm{i}} \qquad \forall \mathrm{i} \in [n] \text{ and } x \in [(\mathrm{i}-1)/n, \mathrm{i}/n)\,.$$

Then $\int_0^1 v(x)\,dx = \sum_{\mathrm{i}=1}^{n} \frac{1}{n} \cdot n\Psi_{\mathrm{i}} = \sum_{\mathrm{i}=1}^{n} \Psi_{\mathrm{i}} = 1$. By Lemma 67, there exists a point $c \in [0,1]$ such that one of the following holds:

(a) $\int_c^{c+p} v(x)\,dx = p$, where $0 \leq c \leq 1-p$;

(b) $\int_0^c v(x)\,dx + \int_{c+1-p}^1 v(x)\,dx = p$, where $1-p < c < 1$.

Case (a).

In this case there exists $c \in [0, 1-p]$ such that $\int_c^{c+p} v(x)\,dx = p$.

We first make a few observations and then define the protocol. Let $\mathrm{i}, \mathrm{j} \in [n]$ be such that

$$\frac{\mathrm{i}-1}{n} \leq c < \frac{\mathrm{i}}{n} \qquad \text{and} \qquad \frac{\mathrm{j}-1}{n} \leq c+p < \frac{\mathrm{j}}{n}\,.$$

Let $T = \mathrm{j} - \mathrm{i} + 1$. Then $np \leq T \leq np + 2$. Since each interval $[(\ell-1)/n, \ell/n]$ corresponds to element x_ℓ of the array, we have $\sum_{\ell=\mathrm{i}}^{\mathrm{j}} \Psi_{\mathrm{i}} \geq p$. By choice of i and j, we have:

- the sub-array $\mathbf{y} = [x_{\mathrm{i}}, \ldots, x_{\mathrm{j}}]$ has length $T \leq np + 2$ and probability mass $\sum_{\ell=\mathrm{i}}^{\mathrm{j}} \Psi_{\mathrm{i}} \geq p$.
- if $T \geq 2$, the sub-array $\tilde{\mathbf{y}} = [x_{\mathrm{i}+1}, \ldots, x_{\mathrm{j}-1}]$ has length $T - 2 \leq np$ and probability mass $\sum_{\ell=\mathrm{i}+1}^{\mathrm{j}-1} \Psi_{\mathrm{i}} \leq p$.

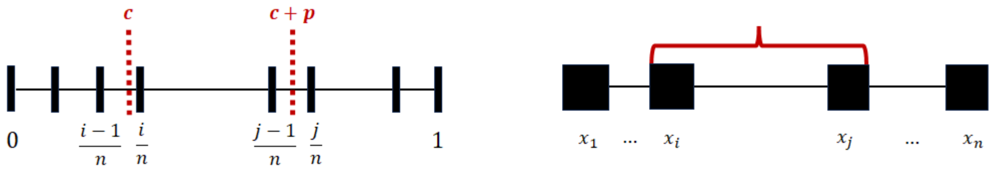


Figure 4.3. Given distribution $\Psi = (\Psi_1, \ldots, \Psi_n)$, define probability density $v : [0,1] \to \mathbb{R}_{\geq 0}$ by $v(x) = n \cdot \Psi_\ell$ for all $\ell \in [n]$ and $x \in [(\ell-1)/n, \ell/n]$. The left figure shows an interval $[c, c+p]$ of length p and probability mass $\int_c^{c+p} v(x)\, dx = p$. The right figure shows the queried sub-array $\mathbf{y} = [x_{\mathrm{i}}, \ldots, x_{\mathrm{j}}]$, which has length $T = \mathrm{j} - \mathrm{i} + 1 \leq np + 2$ and probability mass $\sum_{\ell=\mathrm{i}}^{\mathrm{j}} \Psi_\ell \geq p$. When $T \geq 2$, the sub-array $\tilde{\mathbf{y}} = [x_{\mathrm{i}+1}, \ldots, x_{\mathrm{j}-1}]$ has length $T-2 \leq np$ and probability mass $\sum_{\ell=\mathrm{i}+1}^{\mathrm{j}-1} \Psi_\ell \leq p$.

Let $\mathcal{A}$ be the following k-round protocol:

Step a.(i) **If $T \leq 2$:** query locations i and j in round 1. If the element is found, return it and halt.

Else: since the element is guaranteed to be in the array $\mathbf{x}$, it must be the case that $T \geq 3$. Let $r = \lceil p \cdot n^{\frac{1}{k}} \rceil$. Query in round 1 locations i and j, together with additional locations $t_1, \ldots, t_r$ set as equally spaced as possible.

More precisely, require $\mathrm{i} + 1 \leq t_1 \leq \ldots \leq t_r = \mathrm{j} - 1$, with $t_0 = \mathrm{i}$. For each $\ell \in [r]$, let

$$B_\ell = [x_{(t_{\ell-1}+1)}, \ldots, x_{t_\ell}]$$

be the ℓ-th block created by the queries $t_1, \ldots, t_r$. Define indices $t_1, \ldots, t_r$ so that each block B_ℓ has size at most $\left\lceil \frac{T-2}{r} \right\rceil$, which is possible since the sub-array $\tilde{\mathbf{y}}$ has length $T - 2$ and there are r blocks.

If the element is found at one of the indices $\mathrm{i}, \mathrm{j}, t_1, \ldots, t_r$ queried in round 1, then return it and halt. Otherwise, continue to step *a.(ii)*.

Step a.(ii) If the answers to round 1 queries show the element is not at one of the indices $[\mathrm{i}, \ldots, \mathrm{j}]$, then halt. Else, let $B_\ell = [x_{(t_{\ell-1}+1)}, \ldots, x_{t_\ell}]$ be the block identified to contain the element, where location t_ℓ has been queried. Run the $(k-1)$-round deterministic protocol from Proposition 4.2.1 on the sub-array $\overline{\mathbf{y}} = [x_{(t_{\ell-1}+1)}, \ldots, x_{(t_\ell - 1)}]$, which always succeeds and asks at most $(k-1) \cdot (\mathrm{len}(\overline{\mathbf{y}}))^{\frac{1}{k-1}}$ queries.

We now analyze the success probability and expected number of queries of algorithm $\mathcal{A}$ described in steps *a.(i-ii)*.

Success probability. The algorithm is guaranteed to find the element precisely when it is located in the sub-array $[x_{\mathrm{i}}, \ldots, x_{\mathrm{j}}]$. Since $\sum_{\ell=\mathrm{i}}^{\mathrm{j}} \Psi_\ell \geq p$, the success probability of the algorithm is at least p.

Expected number of queries. We count separately the expected queries for round 1 and the remainder. The number of queries issued in round 1 is at most

$$2 + r = 2 + \lceil p \cdot n^{\frac{1}{k}} \rceil . \tag{4.7}$$

The algorithm continues beyond round 1 when the element is in the sub-array $\widetilde{\mathbf{y}} = [x_{\mathrm{i}+1}, \ldots, x_{\mathrm{j}-1}]$, which has length $T - 2 \leq np$ and probability mass $\sum_{\ell=\mathrm{i}+1}^{\mathrm{j}-1} \leq p$.

Thus with probability at least $1 - p$, the algorithm halts at the end of round 1. With probability at most p, it continues beyond round 1 by running step *a.(ii)*. The number of queries in step *a.(ii)* is bounded by

$$(k-1)\Big(\lceil (T-2)/r \rceil - 1\Big)^{\frac{1}{k-1}}$$

by Proposition 4.2.1 since $\mathrm{len}(\overline{\mathbf{y}}) \leq \lceil \frac{T-2}{r} \rceil - 1$. Since $T - 2 \leq np$ and $r = \lceil p \cdot n^{\frac{1}{k}} \rceil$, the expected number of queries from step *a.(ii)* can be bounded by

$$p \cdot (k-1) \left(\left\lceil \frac{T-2}{r} \right\rceil - 1 \right)^{\frac{1}{k-1}} + (1-p) \cdot 0 = p \cdot (k-1) \left(\left\lceil \frac{T-2}{\lceil p \cdot n^{\frac{1}{k}} \rceil} \right\rceil - 1 \right)^{\frac{1}{k-1}}$$

$$\leq p\cdot(k-1)\left(\left\lceil\frac{np}{\lceil p\cdot n^{\frac{1}{k}}\rceil}\right\rceil-1\right)^{\frac{1}{k-1}} \qquad \text{(Since } T-2\leq np\text{)}$$
$$\leq p\cdot(k-1)\cdot\left(\left\lceil\frac{np}{p\cdot n^{\frac{1}{k}}}\right\rceil-1\right)^{\frac{1}{k-1}} \qquad \left(\text{Since } \tfrac{np}{\lceil p\cdot n^{1/k}\rceil}\leq\tfrac{np}{p\cdot n^{1/k}}\right)$$
$$\leq p\cdot(k-1)\cdot\left(\frac{np}{p\cdot n^{\frac{1}{k}}}\right)^{\frac{1}{k-1}}=p\cdot(k-1)\cdot n^{\frac{1}{k}}\,. \tag{4.8}$$

Combining (4.7) and (4.8), the expected number of queries of algorithm $\mathcal{A}$ is at most

$$2+\lceil p\cdot n^{\frac{1}{k}}\rceil+p\cdot(k-1)\cdot n^{\frac{1}{k}}\leq k\lceil pn^{\frac{1}{k}}\rceil+2\,. \tag{4.9}$$

Case (b).

In this case, there exists $c\in(1-p,1)$ such that $\int_0^c v(x)\,dx+\int_{c+1-p}^1 v(x)\,dx=p$. Let $\mathrm{i},\mathrm{j}\in[n]$ be such that $(\mathrm{i}-1)/n\leq c\leq \mathrm{i}/n$ and $(\mathrm{j}-1)/n\leq c+p-1\leq \mathrm{j}/n$. By choice of i and j, we have $np\leq T\leq np+2$. Then

- the sub-array $\mathbf{y}=[x_1,\ldots,x_{\mathrm{i}},x_{\mathrm{j}},\ldots,x_n]$ has length $T=n+\mathrm{i}-\mathrm{j}+1\leq np+2$ and probability mass $\sum_{\ell=1}^{\mathrm{i}}\Psi_\ell+\sum_{\ell=\mathrm{j}}^{n}\Psi_\ell\geq p$.
- the sub-array $\tilde{\mathbf{y}}=[x_1,\ldots,x_{\mathrm{i}-1},x_{\mathrm{j}+1},\ldots,x_n]$ has length $T-2\leq np$ and probability mass $\sum_{\ell=1}^{\mathrm{i}-1}\Psi_\ell+\sum_{\ell=\mathrm{j}+1}^{n}\Psi_\ell\geq p$.

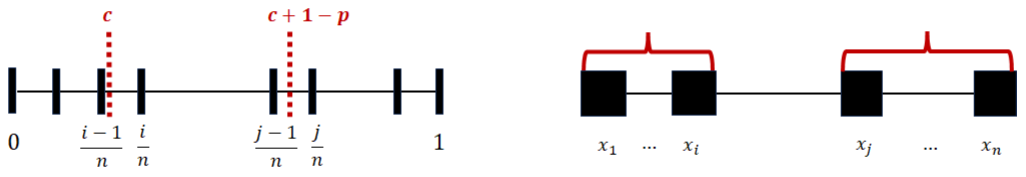


Figure 4.4. Given distribution $\Psi=(\Psi_1,\ldots,\Psi_n)$, define $v:[0,1]\to\mathbb{R}_{\geq 0}$ by $v(x)=n\cdot\Psi_\ell$ for all $\ell\in[n]$ and $x\in[(\ell-1)/n,\ell/n]$. The left figure shows point c with probability mass $\int_0^c v(x)\,dx+\int_{c+1-p}^1 v(x)\,dx=p$. The right figure shows the queried sub-array consisting of two parts: $\mathbf{y}=[x_1,\ldots,x_{\mathrm{i}},x_{\mathrm{j}},\ldots,x_n]$, of length $T=n+\mathrm{i}-\mathrm{j}+1\leq np+2$ and probability mass $\sum_{\ell=1}^{\mathrm{i}}\Psi_\ell+\sum_{\ell=\mathrm{j}}^{n}\Psi_\ell\geq p$. When $T\geq 2$, the sub-array $\tilde{\mathbf{y}}=[x_1,\ldots,x_{\mathrm{i}-1},x_{\mathrm{j}+1},\ldots,x_n]$ has length $T-2\leq np$ and probability mass $\sum_{\ell=1}^{\mathrm{i}-1}\Psi_\ell+\sum_{\ell=\mathrm{j}+1}^{n}\Psi_\ell\leq p$.

Let $\mathcal{A}$ be the same k-round protocol as in case (a), but where the array $\mathbf{y}$ is treated as if it were contiguous when making queries:

Step b.(i) **If $T \leq 2$:** query locations i and j in round 1. If the element is found, return it.

Else, $T \geq 3$. Let $r = \lceil p \cdot n^{\frac{1}{k}} \rceil$. Query in round 1 locations i and j, together with additional locations $t_1, \ldots, t_r \in \{1, \ldots, \mathrm{i}-1, \mathrm{j}+1, \ldots, n\}$, set as equally spaced as possible so that for each $\ell \in [r]$, the size of each block $B_\ell = [x_{(t_{\ell-1}+1)}, \ldots, x_{t_\ell}]$ is at most $\lceil \frac{T-2}{r} \rceil$. At most one of the blocks may skip over the the indices in $\{\mathrm{i}, \ldots, \mathrm{j}\}$. If the element is found at one of the queried locations then return it and halt. Else, go to step *b.(ii)*.

Step b.(ii) If round 1 indicates that the element is not at one of the indices $\{1, \ldots, \mathrm{i}, \mathrm{j}, \ldots, n\}$, then halt. Otherwise, let $B_\ell = [x_{(t_{\ell-1}+1)}, \ldots, x_{t_\ell}]$ be the block identified to contain the element, where location t_ℓ has been queried. Run the $(k-1)$-round deterministic protocol from Proposition 4.2.1 on the sub-array $\overline{\mathbf{y}} = [x_{(t_{\ell-1}+1)}, \ldots, x_{(t_\ell-1)}]$, which always succeeds and asks at most $(k-1) \cdot (\mathrm{len}(\overline{\mathbf{y}}))^{\frac{1}{k-1}}$ queries.

Next we bound the success probability and expected number of queries when the algorithm executes steps b.(i) and b.(ii).

Success probability. The algorithm finds the element when its location is one of $[1, \ldots, \mathrm{i}, \mathrm{j}, \ldots, n]$. Since $\sum_{\ell=1}^{\mathrm{i}} \Psi_\ell + \sum_{\ell=\mathrm{j}}^{n} \Psi_\ell \geq p$, the success probability is at least p.

Expected number of queries. The expected number of queries in round 1 is at most $2 + r = 2 + \lceil pn^{\frac{1}{k}} \rceil$, while the number of queries after round 2 is at most

$$p \cdot (k-1)\Big(\lceil (T-2)/r \rceil - 1\Big)^{\frac{1}{k-1}} \leq p \cdot (k-1) \cdot n^{\frac{1}{k}} .$$

Thus the total expected number of queries is at most $2 + \lceil pn^{\frac{1}{k}} \rceil + p \cdot (k-1) \cdot n^{\frac{1}{k}} \leq 2 + k \lceil pn^{\frac{1}{k}} \rceil$, which completes the proof. $\square$

4.4.2 Ordered search lower bounds

In this section we prove a lower bound that applies to both randomized algorithms on a worst case input and deterministic algorithms on a worst case input distribution. The lower bound considers the expected query complexity of randomized algorithms on the uniform distribution, which turns out to be the hardest distribution for ordered search.

Proposition 4.2.4 (restated). *Let $k, n \in \mathbb{N}^*$ and $p \in (0,1]$. Then $\mathcal{R}_{1-p}(\text{ordered}_{n,k}) \geq pkn^{\frac{1}{k}} - 2pk$ for $k \geq 2$ and $\mathcal{R}_{1-p}(\text{ordered}_{n,1}) \geq np$.*

Proof. For proving the required lower bound, it will suffice to assume the input is drawn from the uniform distribution. This means the algorithm is given a bit vector where the location of the unique bit with value 1 is chosen uniformly at random from $\{1, \dots, n\}$. If a lower bound holds for a randomized algorithm when the input is uniformly distributed, then by an average argument the same lower bound also holds for a worst case input.

Let $\mathcal{A}_k$ be an optimal k-round randomized algorithm that succeeds with probability p when facing the uniform distribution as input. Let $q_k(n,p)$ be the expected number of queries of algorithm $\mathcal{A}_k$ as a function of n and p.

In round 1, the algorithm has some probability δ_m of asking m queries, for each $m \in \{0, \dots, n\}$. Moreover, for each such m, there are different (but finitely many) choices for the positions of the m queries of round 1. However, since the algorithm is optimal, it suffices to restrict attention to the best way of positioning the queries in round 1, breaking ties arbitrarily if there are multiple equally good options.

For each $m \in \{0, \dots, n\}$, we define the following variables:

- δ_m is the probability that the algorithm asks m queries in round one.
- $b_{m,\text{i}}$ is the size of the i-th block demarcated by the indices queried in round 1, excluding those indices, counting from left to right, for all $\text{i} \in \{0, \dots, m\}$. Thus $\sum_{\text{i}=0}^{m} b_{m,\text{i}} = n - m$.

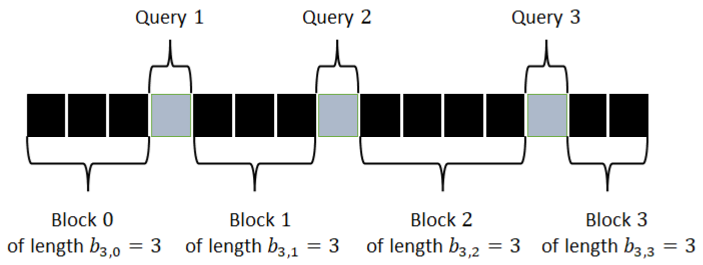


Figure 4.5. Array with $n = 15$ elements. The $m = 3$ locations issued in round 1 are illustrated in gray. The resulting blocks demarcated by these queries are marked, such that the i-th block has length $b_{m,\mathrm{i}}$, for $\mathrm{i} \in \{0, 1, 2, 3\}$.

An illustration with an array and the blocks formed by the queries issued in round 1 can be found in Figure 1.

- $\alpha_{m,\mathrm{i}}$ the success probability of finding the element in the i-th block (as demarcated by the indices queried in round 1), given that the element is in this block.

The expected number of queries of the randomized algorithm is

$$
\begin{aligned}
q_k(n,p) &= \sum_{m=0}^{n} \delta_m \left[m + \left(\frac{n-m}{n}\right) \sum_{\mathrm{i}=0}^{m} \left(\frac{b_{m,\mathrm{i}}}{n-m}\right) \cdot q_{k-1}(b_{m,\mathrm{i}}, \alpha_{m,\mathrm{i}}) \right] \\
&= \sum_{m=0}^{n} \delta_m \left[m + \frac{1}{n} \sum_{\mathrm{i}=0}^{m} b_{m,\mathrm{i}} \cdot q_{k-1}(b_{m,\mathrm{i}}, \alpha_{m,\mathrm{i}}) \right], \qquad (4.10)
\end{aligned}
$$

where the variables are related by the following constraints:

$$\sum_{\mathrm{i}=0}^{m} b_{m,\mathrm{i}} = n - m, \qquad \forall m \in \{0, \ldots, n\} \qquad (4.11)$$

$$\sum_{m=0}^{n} \delta_m = 1 \qquad (4.12)$$

$$p_m = \frac{m}{n} + \frac{n-m}{n} \cdot \sum_{\mathrm{i}=0}^{m} \frac{b_{m,\mathrm{i}}}{n-m} \cdot \alpha_{m,\mathrm{i}} = \frac{m}{n} + \frac{1}{n} \cdot \sum_{\mathrm{i}=0}^{m} b_{m,\mathrm{i}} \cdot \alpha_{m,\mathrm{i}}, \qquad \forall m \in \{0, \ldots, n\} \qquad (4.13)$$

$$p = \sum_{m=0}^{n} \delta_m \cdot p_m \qquad (4.14)$$

$$b_{m,\mathrm{i}} \geq 0, \qquad \forall m \in \{0, \ldots, n\}, \mathrm{i} \in \{0, \ldots, m\} \tag{4.15}$$

$$0 \leq \alpha_{m,\mathrm{i}} \leq 1, \qquad \forall m \in \{0, \ldots, n\}, \mathrm{i} \in \{0, \ldots, m\} \tag{4.16}$$

$$\delta_m \geq 0, \qquad \forall m \in \{0, \ldots, n\}\,. \tag{4.17}$$

Let $\{\gamma_\ell\}_{\ell=1}^{\infty}$ be the sequence given by $\gamma_1 = 0$ and $\gamma_\ell = 2\ell$ for $\ell \geq 2$. We will prove by induction on k that

$$q_k(n,p) \geq p\Big(k \cdot n^{\frac{1}{k}} - \gamma_k\Big) \qquad \forall n, k \geq 1 \text{ and } p \in [0,1]\,. \tag{4.18}$$

Base case.

Proposition 4.4.2 gives $q_1(n,p) \geq np$, so inequality (4.18) holds with $\gamma_1 = 0$ for all $n \geq 1$ and $p \in [0,1]$.

Induction hypothesis.

Suppose $q_\ell(m,s) \geq s\Big(\ell \cdot m^{\frac{1}{\ell}} - \gamma_\ell\Big)$ for all $\ell \in [k-1]$, $m \geq 1$, $s \in [0,1]$.

Induction step.

We will show that (4.18) holds for $k \geq 2$, where $n \geq 1$ and $p \in [0,1]$. The bound clearly holds when $p = 0$, so we will focus on the scenario $p > 0$. For each $m \in \{0, \ldots, n\}$, define

$$r_k(n,m,p) = m + \frac{1}{n} \cdot \sum_{\mathrm{i}=0}^{m} b_{m,\mathrm{i}} \cdot q_{k-1}(b_{m,\mathrm{i}}, \alpha_{m,\mathrm{i}})\,. \tag{4.19}$$

By definition of $q_k(n,p)$,

$$q_k(n,p) = \sum_{m=0}^{n} \delta_m \cdot r_k(n,m,p)\,. \tag{4.20}$$

The induction hypothesis implies $q_{k-1}(b_{m,\mathrm{i}}, \alpha_{m,\mathrm{i}}) \geq \alpha_{m,\mathrm{i}} \cdot \left((k-1)\,(b_{m,\mathrm{i}})^{\frac{1}{k-1}} - \gamma_{k-1} \right)$, which substituted in (4.19) gives

$$r_k(n,m,p) \geq m + \left(\frac{k-1}{n}\right) \cdot \sum_{\mathrm{i}=0}^{m} \alpha_{m,\mathrm{i}} \cdot (b_{m,\mathrm{i}})^{\frac{k}{k-1}} - \left(\frac{\gamma_{k-1}}{n}\right) \cdot \sum_{\mathrm{i}=0}^{m} \alpha_{m,\mathrm{i}} \cdot b_{m,\mathrm{i}}\,. \tag{4.21}$$

Given a choice of $\alpha_{m,\mathrm{i}}, b_{m,\mathrm{i}}$ for all $m \in \{0,\dots,n\}$ and $\mathrm{i} \in \{0,\dots,m\}$, let $\mathrm{i}_0,\dots,\mathrm{i}_m \in \{0,\dots,m\}$ be such that $0 \leq \alpha_{m,\mathrm{i}_0} \leq \dots \leq \alpha_{m,\mathrm{i}_m} \leq 1$. Then we can decompose p_m using a telescoping sum:

$$\begin{aligned}
p_m &= \frac{m}{n} + \frac{1}{n} \cdot \sum_{\mathrm{i}=0}^{m} b_{m,\mathrm{i}} \cdot \alpha_{m,\mathrm{i}} && \text{(By (4.13))}\\
&= \alpha_{m,\mathrm{i}_0} \cdot \left[\frac{m}{n} + \frac{1}{n} \cdot \sum_{\ell=0}^{m} b_{m,\mathrm{i}_\ell}\right] + \left\{\sum_{\mathrm{j}=1}^{m} (\alpha_{m,\mathrm{i}_\mathrm{j}} - \alpha_{m,\mathrm{i}_{\mathrm{j}-1}}) \cdot \left[\frac{m}{n} + \frac{1}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell}\right]\right\} + (1 - \alpha_{m,\mathrm{i}_m}) \cdot \frac{m}{n}\,.
\end{aligned} \tag{4.22}$$

We can similarly decompose the right hand side of inequality (4.21), obtaining:

$$\begin{aligned}
r_k(n,m,p) &\geq m + \frac{k-1}{n} \cdot \sum_{\mathrm{i}=0}^{m} \alpha_{m,\mathrm{i}} \cdot (b_{m,\mathrm{i}})^{\frac{k}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot \sum_{\mathrm{i}=0}^{m} \alpha_{m,\mathrm{i}} \cdot b_{m,\mathrm{i}} && \text{(By (4.21))}\\
&= \alpha_{m,\mathrm{i}_0} \cdot \left[m + \frac{k-1}{n} \cdot \sum_{\ell=0}^{m} (b_{m,\mathrm{i}_\ell})^{\frac{k}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot \sum_{\ell=0}^{m} b_{m,\mathrm{i}_\ell}\right]\\
&\quad + \sum_{\mathrm{j}=1}^{m} (\alpha_{m,\mathrm{i}_\mathrm{j}} - \alpha_{m,\mathrm{i}_{\mathrm{j}-1}}) \cdot \left[m + \frac{k-1}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} (b_{m,\mathrm{i}_\ell})^{\frac{k}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell}\right]\\
&\quad + (1 - \alpha_{m,\mathrm{i}_m}) \cdot m\,.
\end{aligned} \tag{4.23}$$

Let $w_{m,0} = \alpha_{m,\mathrm{i}_0}$, $w_{m,\mathrm{j}} = \alpha_{m,\mathrm{i}_\mathrm{j}} - \alpha_{m,\mathrm{i}_{\mathrm{j}-1}}$ for all $\mathrm{j} \in \{1,\dots,m\}$, and $w_{m,m+1} = 1 - \alpha_{m,\mathrm{i}_m}$. Then we can rewrite (4.22) and (4.23) as follows:

$$r_k(n,m,p) \geq \sum_{\mathrm{j}=0}^{m} w_{m,\mathrm{j}} \cdot \left[m + \frac{k-1}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} (b_{m,\mathrm{i}_\ell})^{\frac{k}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell}\right] + \left(w_{m,m+1} \cdot m\right) \tag{4.24}$$

$$p_m = \sum_{\mathrm{j}=0}^{m} w_{m,\mathrm{j}} \cdot \left[\frac{m}{n} + \frac{1}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell}\right] + \left(w_{m,m+1} \cdot \frac{m}{n}\right). \tag{4.25}$$

For each $m \in \{0, \dots, n\}$ and $\mathrm{j} \in \{0, \dots, m+1\}$, define

$$p_{m,\mathrm{j}} = \begin{cases} \frac{m}{n} + \frac{1}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell} & \text{if } \mathrm{j} \in \{0, \dots, m\}. \\ \\ \frac{m}{n} & \text{if } \mathrm{j} = m+1. \end{cases} \tag{4.26}$$

$$r_k^{\mathrm{j}}(n,m,p) = \begin{cases} m + \frac{k-1}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} \left(b_{m,\mathrm{i}_\ell}\right)^{\frac{k}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell} & \text{if } \mathrm{j} \in \{0, \dots, m\}. \\ m & \text{if } \mathrm{j} = m+1. \end{cases} \tag{4.27}$$

Substituting the definition of $r_k^{\mathrm{j}}(n,m,p)$ in (4.24) and that of $p_{m,\mathrm{j}}$ in (4.25) yields

$$r_k(n,m,p) \geq \sum_{\mathrm{j}=0}^{m+1} w_{m,\mathrm{j}} \cdot r_k^{\mathrm{j}}(n,m,p) \quad \text{and} \quad p_m = \sum_{\mathrm{j}=0}^{m+1} w_{m,\mathrm{j}} \cdot p_{m,\mathrm{j}}\,. \tag{4.28}$$

Combining (4.14), (4.20), and (4.28), we obtain

$$q_k(n,p) = \sum_{m=0}^{n} \delta_m \cdot r_k(n,m,p) \geq \sum_{m=0}^{n} \delta_m \cdot \left(\sum_{\mathrm{j}=0}^{m+1} w_{m,\mathrm{j}} \cdot r_k^{\mathrm{j}}(n,m,p)\right).$$

$$p = \sum_{m=0}^{n} p_m = \sum_{m=0}^{n} \left(\sum_{\mathrm{j}=0}^{m+1} w_{m,\mathrm{j}} \cdot p_{m,\mathrm{j}}\right). \tag{4.29}$$

Let $S_{m,\mathrm{j}} = \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell}$, for all $m \in \{0, \dots, n\}$ and $\mathrm{j} \in \{0, \dots, m\}$. Then for $\mathrm{j} \in \{0, \dots, m\}$, we have $n \cdot p_{m,\mathrm{j}} = m + S_{m,\mathrm{j}}$, so $S_{m,\mathrm{j}} = n \cdot p_{m,\mathrm{j}} - m$. Since $S_{m,\mathrm{j}} \geq 0$, we have $n \cdot p_{m,\mathrm{j}} \geq m$. In summary,

$$\sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell} = n \cdot p_{m,\mathrm{j}} - m, \qquad \forall \mathrm{j} \in \{0, \dots, m\} \tag{4.30}$$

$$n \cdot p_{m,\mathrm{j}} \geq m, \qquad \forall \mathrm{j} \in \{0, \dots, m\}. \tag{4.31}$$

Next we will lower bound $r_k^{\mathrm{j}}(n,m,p)$ and consider two cases, for $m \geq 1$ and $m = 0$.

Case $m \geq 1$.

If $\mathrm{j} = m+1$, we are in the scenario where the algorithm asks m queries in round 1 and no queries in the later rounds. Formally, since $p_{m,m+1} = m/n$, we have $m = n \cdot p_{m,m+1}$. Using the identity for $r_k^{m+1}(n,m,p)$ in (4.27), we obtain

$$\begin{aligned} r_k^{m+1}(n,m,p) &= m = n \cdot p_{m,m+1} \\ &\geq p_{m,m+1} \cdot kn^{\frac{1}{k}} - \gamma_k \cdot p_{m,m+1} \,. && \text{(By Corollary 11.)} \end{aligned}$$

Thus from now on we can assume $\mathrm{j} \in \{0, \ldots, m\}$. Observe that by definition of $p_{m,0}$ in (4.26), we have $p_{m,0} = m/n + \sum\limits_{\ell=0}^{m} b_{m,\mathrm{i}_\ell}/n = m/n + (n-m)/n = 1$. For all $\mathrm{j} \in \{0, \ldots, m\}$, using (4.27) and Jensen's inequality, we obtain

$$\begin{aligned} r_k^{\mathrm{j}}(n,m,p) &= m + \frac{k-1}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} \left(b_{m,\mathrm{i}_\ell}\right)^{\frac{k}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell} \\ &\geq m + \frac{(k-1)\,(m-\mathrm{j}+1)}{n} \left(\sum_{\ell=\mathrm{j}}^{m} \frac{b_{m,\mathrm{i}_\ell}}{m-\mathrm{j}+1} \right)^{\frac{k}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot \sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell} \,. \end{aligned} \tag{4.32}$$

Since $\sum_{\ell=\mathrm{j}}^{m} b_{m,\mathrm{i}_\ell} = n \cdot p_{m,\mathrm{j}} - m$ by (4.30), the inequality in (4.32) can be rewritten as

$$r_k^{\mathrm{j}}(n,m,p) \geq m\left(1 + \frac{\gamma_{k-1}}{n}\right) + \frac{(k-1)(n \cdot p_{m,\mathrm{j}} - m)^{\frac{k}{k-1}}}{n \cdot (m-\mathrm{j}+1)^{\frac{1}{k-1}}} - \gamma_{k-1} \cdot p_{m,\mathrm{j}} \,. \tag{4.33}$$

When $\mathrm{j} = 0$, substituting $\sum_{\ell=0}^{m} b_{m,\mathrm{i}_\ell} = n - m$ in (4.33), we obtain

$$\begin{aligned} r_k^{0}(n,m,p) &\geq m\left(1 + \frac{\gamma_{k-1}}{n}\right) + \frac{(k-1)(n-m)^{\frac{k}{k-1}}}{n \cdot (m+1)^{\frac{1}{k-1}}} - \gamma_{k-1} \\ &\geq kn^{\frac{1}{k}} - \gamma_k && \text{(By Lemma 45.)} \\ &= p_{m,0} \cdot kn^{\frac{1}{k}} - p_{m,0} \cdot \gamma_k \,. && \text{(Since } p_{m,0} = 1\text{)} \end{aligned}$$

Thus from now on we can assume $\mathrm{j} \in \{1, \ldots, m\}$. Using $\mathrm{j} \geq 1$ in (4.33), we further get

$$r_k^{\mathrm{j}}(n, m, p) \geq m\left(1 + \frac{\gamma_{k-1}}{n}\right) + \frac{(k-1)(n \cdot p_{m,\mathrm{j}} - m)^{\frac{k}{k-1}}}{n \cdot m^{\frac{1}{k-1}}} - \gamma_{k-1} \cdot p_{m,\mathrm{j}}\,. \tag{4.34}$$

In this range of m and j, we have $m/n \leq p_{m,\mathrm{j}} \leq 1$ and $1/2 < m \leq n \cdot p_{m,\mathrm{j}}$ by inequality (4.31). Applying Lemma 56 with $c = p_{m,\mathrm{j}}$ in (4.34), we obtain:

$$\begin{aligned} r_k^{\mathrm{j}}(n, m, p) &\geq m\left(1 + \frac{\gamma_{k-1}}{n}\right) + \frac{(k-1)(n \cdot p_{m,\mathrm{j}} - m)^{\frac{k}{k-1}}}{n \cdot m^{\frac{1}{k-1}}} - \gamma_{k-1} \cdot p_{m,\mathrm{j}} && \text{(By (4.34))} \\ &\geq p_{m,\mathrm{j}} \cdot kn^{\frac{1}{k}} - \gamma_k \cdot p_{m,\mathrm{j}}\,. && \text{(By Lemma 56)} \end{aligned}$$

Case $m = 0$.

This corresponds to the scenario where the algorithm asks zero queries in round 1. Since $\mathrm{j} \in \{0, \ldots, m+1\}$, it follows that $\mathrm{j} = 0$ or $\mathrm{j} = 1$.

If $\mathrm{j} = 0$, then by definition of $p_{m,\mathrm{j}}$ we have $p_{0,0} = 0/n + (1/n) \cdot \sum_{\ell=0}^{0} b_{0,\mathrm{i}_\ell} = 0 + b_{0,\mathrm{i}_0}/n$. Since there is only one block, $b_{0,\mathrm{i}_0} = n$. Thus $p_{0,0} = 1$. We get

$$\begin{aligned} r_k^{0}(n, 0, p) &= 0 + \frac{k-1}{n} \cdot \sum_{\ell=0}^{0} (b_{0,\mathrm{i}_\ell})^{\frac{k}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot \sum_{\ell=0}^{0} b_{0,\mathrm{i}_\ell} \\ &= (k-1) \cdot n^{\frac{1}{k-1}} - \frac{\gamma_{k-1}}{n} \cdot n && \text{(Since } b_{0,\mathrm{i}_0} = n\text{)} \\ &\geq kn^{\frac{1}{k}} - \gamma_k && \text{(By Corollary 10)} \\ &= p_{0,0} \cdot kn^{\frac{1}{k}} - p_{0,0} \cdot \gamma_k\,. && \text{(Since } p_{0,0} = 1\text{)} \end{aligned}$$

If $\mathrm{j} = 1$, then since $m = 0$ we are in the case $\mathrm{j} = m + 1$. Since $p_{m,m+1} = m/n$, we have $p_{0,1} = 0/n = 0$. Informally, this corresponds to the scenario where the algorithm asks $m = 0$ queries in round 1 and no queries in the later rounds either. Formally,

$$r_k^{1}(n, 0, p) = 0 = p_{0,1} \cdot kn^{\frac{1}{k}} - p_{0,1} \cdot \gamma_k\,. \tag{4.35}$$

Combining cases $m \geq 1$ and $m = 0$.

We obtain

$$r_k^{\mathrm{j}}(n, m, p) \geq p_{m,\mathrm{j}} \cdot kn^{\frac{1}{k}} - p_{m,\mathrm{j}} \cdot \gamma_k \quad \forall m \in \{0, \ldots, n\}, \forall \mathrm{j} \in \{0, \ldots, m+1\} \tag{4.36}$$

Summing inequality (4.36) over all $m \in \{0, \ldots, n\}$ and $\mathrm{j} \in \{0, \ldots, m+1\}$ and using identity (4.29) that expresses the total expected number of queries $q_k(n, p)$ as a weighted sum of the $r_k^{\mathrm{j}}(n, m, p)$ terms, we obtain

$$\begin{aligned} q_k(n, p) &= \sum_{m=0}^{n} \delta_m \cdot \left(\sum_{\mathrm{j}=0}^{m+1} w_{m,\mathrm{j}} \cdot r_k^{\mathrm{j}}(n, m, p) \right) \\ &\geq \sum_{m=0}^{n} \delta_m \cdot \left(\sum_{\mathrm{j}=0}^{m+1} w_{m,\mathrm{j}} \cdot \left(p_{m,\mathrm{j}} \cdot kn^{\frac{1}{k}} - p_{m,\mathrm{j}} \cdot \gamma_k \right) \right) && \text{(By inequality (4.36))} \\ &= \sum_{m=0}^{n} \delta_m \cdot p_m \cdot \left(kn^{\frac{1}{k}} - \gamma_k \right) && \text{(Since } p_m = \textstyle\sum_{\mathrm{j}=0}^{m+1} w_{m,\mathrm{j}} \cdot p_{m,\mathrm{j}} \text{ by (4.28))} \\ &= p \cdot \left(kn^{\frac{1}{k}} - \gamma_k \right) . && \text{(Since } p = \textstyle\sum_{m=0}^{n} \delta_m \cdot p_m \text{ by (4.14))} \end{aligned}$$

This completes the induction step and the proof. □

We consider separately the case of $k = 1$ rounds, giving a lower bound that applies to both ordered and unordered search.

Proposition 4.4.2. *Let $p \in (0, 1]$ and $n \in \mathbb{N}_{\geq 1}$. Then*

$$\mathcal{R}_{1-p}(\text{unordered}_{n,1}) \geq np \quad \textit{and} \quad \mathcal{R}_{1-p}(\text{ordered}_{n,1}) \geq np\,. \tag{4.37}$$

Proof. We show the lower bound for randomized algorithms when facing the uniform distribution. For one round, there is no distinction between ordered and unordered search. By an average argument, the lower bound obtained applies to a worst case input.

Let $\mathcal{A}_1$ be a randomized algorithm that runs in one round and succeeds with probability p when given an input drawn from the uniform distribution. Let $q_1(n, p)$ be the expected number of queries of $\mathcal{A}_1$ as a function of the input size n and the success probability p. Denote by δ_m the probability that the algorithm issues m queries in round 1. Since there is no second round and the input distribution is uniform, the location of these queries does not matter.

The expected number of queries issued by the algorithm on the uniform input distribution can be written as

$$q_1(n,p) = \sum_{m=0}^{n} \delta_m \cdot m,$$

where

$$\sum_{m=0}^{n} \delta_m = 1 \tag{4.38}$$

$$p = \sum_{m=0}^{n} \delta_m \cdot \left(\frac{m}{n}\right) \tag{4.39}$$

$$\delta_m \geq 0 \;\; \forall m \in \{0, \ldots, n\}\,. \tag{4.40}$$

Thus we have $q_1(n,p) = \sum_{m=0}^{n} \delta_m \cdot m = np$. □

4.4.3 Lemmas for ordered search proofs

In this section we include the lemmas used to prove the ordered search upper and lower bounds.

Lemma 45. *Let $k \geq 2, n \geq 1$, and the sequence $\{\gamma_\ell\}_{\ell=1}^{\infty}$ with $\gamma_1 = 0$ and $\gamma_\ell = 2\ell$ for all $\ell \geq 2$. Then*

$$x\left(1 + \frac{\gamma_{k-1}}{n}\right) + (k-1) \cdot \frac{(n-x)^{\frac{k}{k-1}}}{n \cdot (x+1)^{\frac{1}{k-1}}} - \gamma_{k-1} \geq kn^{\frac{1}{k}} - \gamma_k \;\; \forall x \in (1/2, n]\,. \tag{4.41}$$

Proof. Let $t = \left(\frac{n-x}{x+1}\right)^{\frac{1}{k-1}}$. Then t is decreasing in x. Since $x \in (1/2, n]$, we have $0 \le t < \left(\frac{2n-1}{3}\right)^{\frac{1}{k-1}}$. Expressing x in terms of t we get

$$x = \frac{n - t^{k-1}}{t^{k-1} + 1}\,. \tag{4.42}$$

Substituting (4.42) in (4.41), we get that (4.41) is equivalent to

$$t^k \cdot (k-1)(n+1) - t^{k-1} \cdot \left(kn^{\frac{k+1}{k}} + n + \gamma_{k-1}(n+1) - n\gamma_k\right) + \left(n^2 + n\gamma_k - kn^{\frac{k+1}{k}}\right) \ge 0 \qquad \forall\, 0 \le t < \left(\frac{2n-1}{3}\right)^{\frac{1}{k-1}}. \tag{4.43}$$

We consider two cases, for $k = 2$ and $k \ge 3$.

Case $k = 2$.

Since $\gamma_1 = 0$ and $\gamma_2 = 4$, inequality (4.43) is equivalent to

$$t^2 \cdot (n+1) - t \cdot \left(2n\sqrt{n} - 3n\right) + n^2 - 2n\sqrt{n} + 4n \ge 0 \qquad \forall\, 0 \le t < \frac{2n-1}{3}\,. \tag{4.44}$$

Inequality (4.44) holds by Lemma 46.

Case $k \ge 3$.

Since $\gamma_1 = 0$ and $\gamma_\ell = 2\ell$ for $\ell \ge 2$, inequality (4.43) can be simplified to

$$t^k \cdot (k-1)(n+1) - t^{k-1} \cdot \left(k \cdot n^{\frac{k+1}{k}} - n + 2k - 2\right) + n^2 + 2kn - kn^{1+\frac{1}{k}} \ge 0 \qquad \forall\, 0 \le t < \left(\frac{2n-1}{3}\right)^{\frac{1}{k-1}}. \tag{4.45}$$

Inequality (4.45) holds by Lemma 47 for all $k \ge 3$. This completes the proof. □

Lemma 46. *Let $n \geq 1$. Then for all $t \in \Big[0, (2n-1)/3\Big)$, we have*

$$t^2 \cdot (n+1) - t \cdot \Big(2n\sqrt{n} - 3n\Big) + n^2 - 2n\sqrt{n} + 4n \geq 0\,. \tag{4.46}$$

Proof. Let $f : \mathbb{R} \to \mathbb{R}$ be $f(t) = t^2 \cdot (n+1) - t \cdot (2n\sqrt{n} - 3n) + n^2 - 2n\sqrt{n} + 4n$. Then

$$f'(t) = 2t(n+1) - (2n\sqrt{n} - 3n) \quad \text{and} \quad f''(t) = 2(n+1)\,. \tag{4.47}$$

Thus f is convex and the global minimum is at t^* for which $f'(t^*) = 0$, that is, $t^* = \frac{2n\sqrt{n}-3n}{2n+2}$.

Evaluating $f(t^*)$ gives

$$\begin{aligned} f(t^*) &= \left(\frac{2n\sqrt{n}-3n}{2n+2}\right)^2 \cdot (n+1) - \left(\frac{2n\sqrt{n}-3n}{2n+2}\right) \cdot \Big(2n\sqrt{n} - 3n\Big) + n^2 - 2n\sqrt{n} + 4n \\ &= \frac{11n^2 - 8n\sqrt{n} + 16n + 4n^2\sqrt{n}}{4n+4} \\ &> 0\,. && \text{(Since } 11n^2 > 8n\sqrt{n} \text{ for } n \geq 1) \end{aligned}$$

Thus $f(t) \geq f(t^*) > 0$ for all $t \in \mathbb{R}$, which implies the inequality required by the lemma. □

Lemma 47. *Let $n \geq 1$ and $k \geq 3$. Then*

$$t^k \cdot (k-1)(n+1) - t^{k-1} \cdot \Big(k \cdot n^{\frac{k+1}{k}} - n + 2k - 2\Big) + n^2 + 2kn - kn^{1+\frac{1}{k}} \geq 0, \quad \forall t \in \Big[0, n^{\frac{1}{k-1}}\Big)\,. \tag{4.48}$$

Proof. Dividing both sides of (4.48) by n^2, we get that (4.48) holds if and only if

$$\left(\frac{t}{n^{\frac{1}{k-1}}}\right)^k \cdot (k-1)\left(1+\frac{1}{n}\right) \cdot n^{\frac{1}{k-1}} - \left(\frac{t}{n^{\frac{1}{k-1}}}\right)^{k-1} \cdot \left(k \cdot n^{\frac{1}{k}} - 1 + \frac{2k-2}{n}\right) + 1 + \frac{2k}{n} - \frac{k}{n^{\frac{k-1}{k}}} \geq 0 \quad \forall t \in \Big[0, n^{\frac{1}{k-1}}\Big)\,. \tag{4.49}$$

If $t = 0$ then (4.49) is equivalent to $n + 2k - kn^{\frac{1}{k}} \geq 0$, which holds by Corollary 11.

For $t > 0$, let $x = n^{\frac{1}{k-1}}/t$. Since $0 < t < n^{\frac{1}{k-1}}$, we have $x > 0$. Substituting t by x we obtain that (4.49) is equivalent to

$$x^k \cdot \left(1 + \frac{2k}{n} - \frac{k}{n^{\frac{k-1}{k}}}\right) - x \cdot \left(k \cdot n^{\frac{1}{k}} - 1 + \frac{2k-2}{n}\right) + (k-1)\left(1 + \frac{1}{n}\right) \cdot n^{\frac{1}{k-1}} \geq 0, \quad \forall x > 1\,. \tag{4.50}$$

Define the function $f : (0, \infty) \to \mathbb{R}$, where $f(x)$ is given by the left hand side of (4.50). Then

$$\begin{aligned} f'(x) &= k\left(1 + \frac{2k}{n} - \frac{k}{n^{\frac{k-1}{k}}}\right) x^{k-1} - \left(k \cdot n^{\frac{1}{k}} - 1 + \frac{2k-2}{n}\right) \\ f''(x) &= k(k-1)\left(1 + \frac{2k}{n} - \frac{k}{n^{\frac{k-1}{k}}}\right) x^{k-2}\,. \end{aligned} \tag{4.51}$$

By Corollary 11, we have $1 + 2k/n - k/n^{\frac{k-1}{k}} > 0$ for all $n \geq 1$ and $k \geq 3$. Thus $f''(x) > 0$ for all $x > 0$, so f is convex on $(0, \infty)$. Observe that $k \cdot n^{\frac{1}{k}} - 1 + \frac{2k-2}{n} > 0$ for all $n \geq 1$ and $k \geq 3$. Then there is a global minimum of f at a point $\overline{x} \in (0, \infty)$ with $f'(\overline{x}) = 0$, or equivalently,

$$\overline{x} = \left(\frac{k \cdot n^{\frac{1}{k}} - 1 + \frac{2k-2}{n}}{k\left(1 + \frac{2k}{n} - \frac{k}{n^{\frac{k-1}{k}}}\right)}\right)^{\frac{1}{k-1}}. \tag{4.52}$$

Evaluating f at $\overline{x}$ and rearranging terms gives

$$\begin{aligned} f(\overline{x}) &= \overline{x}\left[\left(1 + \frac{2k}{n} - \frac{k}{n^{\frac{k-1}{k}}}\right)\overline{x}^{k-1} - \left(k \cdot n^{\frac{1}{k}} - 1 + \frac{2k-2}{n}\right)\right] + (k-1)\left(1 + \frac{1}{n}\right) \cdot n^{\frac{1}{k-1}} \\ &= \overline{x}\left(\frac{1}{k} - 1\right)\left(k \cdot n^{\frac{1}{k}} - 1 + \frac{2k-2}{n}\right) + (k-1)\left(1 + \frac{1}{n}\right) \cdot n^{\frac{1}{k-1}} \\ &= (k-1)\left(1 + \frac{1}{n}\right) \cdot n^{\frac{1}{k-1}} - (k-1)\left(n^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{n} - \frac{2}{kn}\right)^{\frac{k}{k-1}} \left(\frac{n}{n + 2k - kn^{\frac{1}{k}}}\right)^{\frac{1}{k-1}}. \end{aligned} \tag{4.53}$$

Thus $f(x) > 0 \;\; \forall x > 1$ whenever the next two properties are met

1. $f(\overline{x}) > 0$ when $\overline{x} > 1$. This is equivalent to

$$\left(n + 2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{1}{n}\right)^{k-1} > \left(n^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{n} - \frac{2}{kn}\right)^{k} \tag{4.54}$$

$$\text{whenever } n^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{n} - \frac{2}{kn} > 1 + \frac{2k}{n} - \frac{k}{n^{\frac{k-1}{k}}}\,. \tag{4.55}$$

Lemma 49 implies that inequality (4.54) holds under condition (4.55).

2. $f(1) \geq 0$ when $\overline{x} < 1$. To show this, observe that for all $n \geq 1$ and $k \geq 3$, we have

$$\begin{aligned} f(1) &= \left(1 + \frac{1}{n}\right)\left(2 - k \cdot n^{\frac{1}{k}} + (k-1) \cdot n^{\frac{1}{k-1}}\right) \\ &\geq 0\,. && \text{(By Lemma 48)} \end{aligned}$$

Thus $f(1) \geq 0$ for all $n \geq 1$ and $k \geq 3$, which completes property 2.

Since both properties 1 and 2 hold, we have that $f(x) > 0$ for all $x > 1$, so (4.50) holds. Equivalently, (4.48) holds as required by the lemma. □

Lemma 48. *Let $n \geq 1$ and $k \geq 3$, where $k, n \in \mathbb{N}$. Then $2 - k \cdot n^{\frac{1}{k}} + (k-1) \cdot n^{\frac{1}{k-1}} \geq 0$.*

Proof. Consider the function $f : [2, \infty) \to \mathbb{R}$ given by $f(x) = xn^{\frac{1}{x}}$. We first show an upper bound on f' and then use it to upper bound $f(k) - f(k-1)$. We have

$$f'(x) = n^{\frac{1}{x}}\left(1 - \frac{\ln(n)}{x}\right)\,.$$

Let $y = n^{\frac{1}{x}}$. Then $\ln(y) = \frac{1}{x}\ln(n)$. Since $x \geq 2$, we have $y \in (1, \sqrt{n}]$. Then

$$n^{\frac{1}{x}}\left(1 - \frac{\ln(n)}{x}\right) = y\left(1 - \ln(y)\right)\,. \tag{4.56}$$

The function $g : (1, \infty) \to \mathbb{R}$ given by $g(y) = y\left(1 - \ln(y)\right)$ has $g'(y) = -\ln(y) < 0$. Therefore $g(y) < g(1) = 1$ for all $y > 1$. Using the identity in (4.56), we get that $f'(x) < 1$ for all $x \geq 2$.

Then for all $k \geq 3$,

$$k \cdot n^{\frac{1}{k}} = f(k) \leq f(k-1) + \max_{x \in [2,\infty)} f'(x) < f(k-1) + 1 = (k-1) \cdot n^{\frac{1}{k-1}} + 1\,. \tag{4.57}$$

Inequality (4.57) implies the lemma statement. □

Corollary 10. *Let $n \geq 1$ and $k \geq 2$. Suppose $\{\gamma_\ell\}_{\ell=1}^{\infty}$ is the sequence given by $\gamma_1 = 0$ and $\gamma_\ell = 2\ell$ for $\ell \geq 2$. Then $(k-1) \cdot n^{\frac{1}{k-1}} - \gamma_{k-1} \geq kn^{\frac{1}{k}} - \gamma_k$.*

Proof. If $k = 2$, the required inequality is $n \geq 2\sqrt{n} - 4$, or $(\sqrt{n} - 1)^2 + 3 \geq 0$. The latter holds for all $n \geq 1$. If $k \geq 3$, the required inequality is $(k-1) \cdot n^{\frac{1}{k-1}} + 2 \geq kn^{\frac{1}{k}}$, which holds by Lemma 48. □

Lemma 49. *Let $n \geq 1$ and $k \geq 3$, where $k, n \in \mathbb{N}$. Then*

$$(n + 2k - kn^{\frac{1}{k}}) \left(1 + \frac{1}{n}\right)^{k-1} > \left(n^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{n} - \frac{2}{kn}\right)^k \tag{4.58}$$

$$\text{whenever } \; n^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{n} - \frac{2}{kn} > 1 + \frac{2k}{n} - \frac{k}{n^{\frac{k-1}{k}}}\,. \tag{4.59}$$

Proof. If $n = 1$ then the condition in (4.59) is equivalent to $1 - 1/k + 2 - 2/k > 1 + 2k - k$, which holds if and only if $2 - 3/k > k$ (†). Since $k \geq 3$, inequality (†) does not hold so condition (4.59) is not met either.

Thus from now on we can assume $n \geq 2$. By Lemma 53, condition (4.59) implies $k < n$. We show (4.58) holds when $n \geq 2$ and $k < n$ by considering separately a few ranges of k.

Case I: $n/2 < k < n$ and $k \geq 3$.

Then $k < n < 2k$. When $n = 2k - 1$ inequality (4.58) holds by Lemma 52.

Thus from now on we can assume $k < n \leq 2k - 2$. To show inequality (4.58), we will first bound separately several of the terms in the inequality and then combine the bounds.

For $k \geq 3$, we have $k \leq 2^{k-1}$. Moreover, since $n < 2k$, we have $n^{\frac{1}{k}} < (2k)^{\frac{1}{k}} \leq 2$, and so $2k > kn^{\frac{1}{k}}$. Thus $n + 2k - kn^{\frac{1}{k}} > n$, which implies

$$\left(n + 2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{1}{n}\right)^{k-1} > n\left(1 + \frac{1}{n}\right)^{k-1}. \tag{4.60}$$

Moreover, since $n \geq 2$, we have $2k \leq kn \cdot n^{\frac{1}{k}}$, and so $2k - 2 - n < kn \cdot n^{\frac{1}{k}}$. Since $n \leq 2k - 2$, we also have $2k - 2 - n \geq 0$, and so

$$0 \leq \frac{2k - 2 - n}{kn \cdot n^{\frac{1}{k}}} < 1\,. \tag{4.61}$$

Let $r = (2k - 2 - n)/(kn \cdot n^{\frac{1}{k}})$. Inequality (4.61) gives $0 \leq r < 1$. We consider two sub-cases:

- If $n = 2k - 2$ then $r = 0$. We have

$$\left(1 + \frac{2k - 2 - n}{kn \cdot n^{\frac{1}{k}}}\right)^k = (1 + r)^k = 1 = \mathrm{e}^0 = \mathrm{e}^{\left(\frac{2k-2-n}{n \cdot n^{\frac{1}{k}}}\right)}. \tag{4.62}$$

- Else $k < n < 2k - 2$. Then $0 < r < 1$. We have

$$\begin{aligned}\left(1 + \frac{2k - 2 - n}{kn \cdot n^{\frac{1}{k}}}\right)^k &= (1 + r)^k && \text{(By definition of } r\text{)}\\ &= \left[(1 + r)^{\frac{1}{r}}\right]^{kr} \\ &\leq \mathrm{e}^{kr} && \text{(Since } (1 + r)^{\frac{1}{r}} \leq \mathrm{e} \text{ for } r \in (0, 1).\text{)}\\ &= \mathrm{e}^{\left(\frac{2k-2-n}{n \cdot n^{\frac{1}{k}}}\right)}. \end{aligned} \tag{4.63}$$

Combining inequalities (4.62) and (4.63) from the two sub-cases, we obtain

$$\left(1 + \frac{2k - 2 - n}{kn \cdot n^{\frac{1}{k}}}\right)^k \leq \mathrm{e}^{\left(\frac{2k-2-n}{n \cdot n^{\frac{1}{k}}}\right)} \qquad \forall n \in \mathbb{N} \text{ with } k < n \leq 2k - 2\,. \tag{4.64}$$

Using (4.64), we can upper bound the right hand side of inequality (4.58) as follows:

$$\begin{aligned}\left(n^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{n} - \frac{2}{kn}\right)^k &= \left(n^{\frac{1}{k}} + \frac{2k-2-n}{kn}\right)^k \\ &= n\left(1 + \frac{2k-2-n}{kn \cdot n^{\frac{1}{k}}}\right)^k \\ &\leq n\mathrm{e}^{\left(\frac{2k-2-n}{n \cdot n^{\frac{1}{k}}}\right)} . \end{aligned} \tag{4.65}$$

By Lemma 54, we have

$$\mathrm{e}^{\left(\frac{2k-2-n}{n \cdot n^{\frac{1}{k}}}\right)} \leq \left(1 + \frac{1}{n}\right)^{k-1} . \tag{4.66}$$

Combining (4.60), (4.65), and (4.66), gives:

$$\begin{aligned}\left(n^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{n} - \frac{2}{kn}\right)^k &\leq n\mathrm{e}^{\left(\frac{2k-2-n}{n \cdot n^{\frac{1}{k}}}\right)} && \text{(By (4.65))} \\ &\leq n\left(1 + \frac{1}{n}\right)^{k-1} && \text{(By (4.66))} \\ &< \left(n + 2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{1}{n}\right)^{k-1} . && \text{(By (4.60))}\end{aligned}$$

In summary,

$$\left(n^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{n} - \frac{2}{kn}\right)^k \leq \left(n + 2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{1}{n}\right)^{k-1} \qquad \forall n \in \mathbb{N} \text{ with } k < n \leq 2k-2 .$$

This is the required inequality (4.58), which completes case I.

Case II: $3 \le k \le n/2$.

Then $n \ge 2k$ and $k \ge 3$. Then $t^k \ge 2k$. For $k = 3$, the required inequality (4.58) is equivalent to

$$\left(x + 6 - 3x^{\frac{1}{3}}\right)\left(1 + \frac{1}{x}\right)^2 - \left(x^{\frac{1}{3}} - \frac{1}{3} + \frac{2}{x} - \frac{2}{3x}\right)^3 > 0 \qquad \forall x \ge 6^{\frac{1}{3}},$$

which can be easily checked to hold (see, e.g., [140]).

Thus from now on we can assume $k \ge 4$ with $k \in \mathbb{N}$. Let $f : (0, \infty) \to \mathbb{R}$ be

$$f(x) = \left(x + 2k - kx^{\frac{1}{k}}\right)\left(1 + \frac{1}{x}\right)^{k-1} - \left(x^{\frac{1}{k}} - \frac{1}{k} + \frac{2}{x} - \frac{2}{kx}\right)^k .$$

Using Bernoulli's inequality gives

$$\left(1 + \frac{1}{n}\right)^{k-1} \ge 1 + \frac{k-1}{n} . \tag{4.67}$$

Since $n \ge 2k$, we have $2/n \le 1/k$. Thus

$$-\frac{1}{k} + \frac{2k-2}{kn} \le -\frac{1}{k} + \frac{k-1}{k^2} = -\frac{1}{k^2} . \tag{4.68}$$

Using (4.67) and (4.68), we can lower bound $f(n)$ as follows:

$$\begin{aligned} f(n) &= \left(n + 2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{1}{n}\right)^{k-1} - \left(n^{\frac{1}{k}} - \frac{1}{k} + \frac{2k-2}{kn}\right)^k \\ &\ge \left(n + 2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{k-1}{n}\right) - \left(n^{\frac{1}{k}} - \frac{1}{k^2}\right)^k && \text{(By (4.67) and (4.68))} \\ &\ge \left(2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{k-1}{n}\right) + n - \left(n^{\frac{1}{k}} - \frac{1}{k^2}\right)^k . && (4.69) \end{aligned}$$

If $n^{\frac{1}{k}} \le 2$, then $2k - kn^{\frac{1}{k}} \ge 0$, which together with (4.69) yields $f(n) \ge n - \left(n^{\frac{1}{k}} - \frac{1}{k^2}\right)^k \ge 0$, as required.

Thus from now on we will assume $n^{\frac{1}{k}} > 2$, that is, $n > 2^k$. Then $2k - kn^{\frac{1}{k}} < 0$. Together with $n \geq 2k$, this implies

$$\left(2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{k-1}{n}\right) > \left(2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{k-1}{2k}\right). \tag{4.70}$$

Inequality (4.70) together with $(k-1)/2k < 1/2$ yields

$$\left(2k - kn^{\frac{1}{k}}\right)\left(1 + \frac{k-1}{n}\right) > 1.5 \cdot \left(2k - kn^{\frac{1}{k}}\right). \tag{4.71}$$

Combining (4.69) and (4.71) gives

$$f(n) \geq 1.5 \cdot \left(2k - kn^{\frac{1}{k}}\right) + n - \left(n^{\frac{1}{k}} - \frac{1}{k^2}\right)^k. \tag{4.72}$$

Next we expand and truncate $\left(n^{\frac{1}{k}} - \frac{1}{k^2}\right)^k$ via Lemma 50, yielding

$$-\left(n^{\frac{1}{k}} - \frac{1}{k^2}\right)^k \geq -n + \frac{n^{1-\frac{1}{k}}}{k} - \frac{(k-1)n^{1-\frac{2}{k}}}{2k^3}. \tag{4.73}$$

Using (4.73), we can further bound $f(n)$ by

$$\begin{aligned} f(n) &\geq 1.5 \cdot \left(2k - kn^{\frac{1}{k}}\right) + \frac{n^{1-\frac{1}{k}}}{k} - \frac{(k-1)n^{1-\frac{2}{k}}}{2k^3} && \text{(Combining (4.72) and (4.73))} \\ &\geq 0. && \text{(By Lemma 51)} \end{aligned}$$

Thus $f(n) > 0$, as required. This completes the analysis for the range $n > 2^k$ and case II.

Wrapping up.

We obtain that inequality (4.58) holds under condition (4.59), as required. □

Lemma 50. *Let $k, n \in \mathbb{N}$ with $n \geq 1$ and $k \geq 3$. Then*

$$\left(n^{\frac{1}{k}} - \frac{1}{k^2}\right)^k \leq n - \frac{n^{1-\frac{1}{k}}}{k} + \frac{(k-1) \cdot n^{1-\frac{2}{k}}}{2k^3}\,. \tag{4.74}$$

Proof. Let $t = n^{\frac{1}{k}}$. Then $t \geq 1$. The required inequality (4.74) is equivalent to

$$\left(t - \frac{1}{k^2}\right)^k \leq t^k - \frac{t^{k-1}}{k} + \frac{(k-1) \cdot t^{k-2}}{2k^3}\,. \tag{4.75}$$

For $\mathrm{i} \in [k+1]$ let c_i be the i-th term in the binomial expansion of $(t - 1/k^2)^k$. In particular,

$$c_1 = t^k; \qquad c_2 = -\frac{t^{k-1}}{k}; \qquad c_3 = \frac{(k-1)t^{k-2}}{2k^3}; \qquad c_{k+1} = (-1)^k \frac{1}{k^{2k}}\,. \tag{4.76}$$

Let us bound the ratio $|c_\mathrm{i}/c_{\mathrm{i}+1}|$ for $\mathrm{i} \in [k]$:

$$\left|\frac{c_\mathrm{i}}{c_{\mathrm{i}+1}}\right| = \frac{k^2 \cdot t\mathrm{i}}{k - \mathrm{i} + 1} \geq tk > 1\,. \tag{4.77}$$

Since $c_{2\mathrm{i}} < 0$ and $c_{2\mathrm{i}+1} > 0$ for all i, inequality (4.77) implies

$$c_\mathrm{i} + c_{\mathrm{i}+1} \leq 0 \qquad \forall \mathrm{i} \in [k] \text{ with } \mathrm{i} \in 2\mathbb{N}\,. \tag{4.78}$$

We bound the term $\left(t - \frac{1}{k^2}\right)^k$ by considering two cases. If k is even, then

$$\begin{aligned}\left(t - \frac{1}{k^2}\right)^k &= c_1 + c_2 + c_3 + \sum_{\mathrm{i}=2}^{k/2} (c_{2\mathrm{i}} + c_{2\mathrm{i}+1}) && \text{(By definition of } c_\mathrm{i}.\text{)}\\ &\leq c_1 + c_2 + c_3\,. && \text{(Since } c_{2\mathrm{i}} + c_{2\mathrm{i}+1} < 0 \text{ by (4.78))}\end{aligned}$$

If k is odd, then

$$\begin{aligned}\left(t - \frac{1}{k^2}\right)^k &= \sum_{\mathrm{i}=1}^{k+1} c_\mathrm{i} && \text{(By definition of } c_\mathrm{i}.\text{)}\\ &< \sum_{\mathrm{i}=1}^{k} c_\mathrm{i} = c_1 + c_2 + c_3 + \sum_{\mathrm{i}=2}^{(k-1)/2} (c_{2\mathrm{i}} + c_{2\mathrm{i}+1}) && \text{(Since } c_{k+1} < 0.\text{)}\end{aligned}$$

$$\leq c_1 + c_2 + c_3\,. \qquad \text{(Since } c_{2i} + c_{2i+1} < 0 \text{ by (4.78))}$$

Thus for both odd and even k, we have

$$\left(t - \frac{1}{k^2}\right)^k \leq c_1 + c_2 + c_3 = t^k - \frac{t^{k-1}}{k} + \frac{(k-1)t^{k-2}}{2k^3}\,. \tag{4.79}$$

Thus in both cases (4.75) holds, as required. ☐

Lemma 51. *Let $k, n \in \mathbb{N}$ with $n \geq 2$ and $k \geq 4$. Then*

$$1.5\left(2k - kn^{\frac{1}{k}}\right) + \frac{n^{1-\frac{1}{k}}}{k} - \frac{(k-1)\cdot n^{1-\frac{2}{k}}}{2k^3} \geq 0\,. \tag{4.80}$$

Proof. Let $t = n^{\frac{1}{k}}$. Then $t > 1$ since $n \geq 2$. For $k = 4$, inequality (4.80) with n substituted by t^4 is equivalent to $1.5(8-4t) + t^3/4 - 3t^2/128 \geq 0$, which holds for all $t > 1$.

Thus from now on we can assume $k \geq 5$. The left hand side of (4.80), where n is substituted by t^k, can be bounded as follows:

$$\begin{aligned} 1.5\,(2k - kt) + \frac{t^{k-1}}{k} - \frac{(k-1)\cdot t^{k-2}}{2k^3} &\geq 1.5(2k-kt) + \frac{t^{k-2}}{2k^2}\Big(2kt - 1\Big) && \text{(Since } k-1 < k\text{)} \\ &\geq 1.5\left(2k - kt + \frac{t^{k-1}}{2k}\right). && \text{(Since } t > 1 \text{ and } k \geq 4\text{)} \end{aligned} \tag{4.81}$$

Let $g : (0,\infty) \to \mathbb{R}$ be $g(t) = 2k - kt + \frac{t^{k-1}}{2k}$. We will show that $g(t) \geq 0$ for all $t > 1$. We have

$$g'(t) = -k + \frac{k-1}{2k}\cdot t^{k-2} \qquad \text{and} \qquad g''(t) = \frac{(k-1)(k-2)}{2k}\cdot t^{k-3}\,. \tag{4.82}$$

Thus g is convex on $(0,\infty)$. The global minimum is t^* with $g'(t^*)=0$, so $t^* = \left(\frac{2k^2}{k-1}\right)^{\frac{1}{k-2}}$. Then

$$g(t) \geq g(t^*) = 2k - k\cdot t^* + t^* \cdot \frac{(t^*)^{k-2}}{2k} = 2k - \left(\frac{2k^2}{k-1}\right)^{\frac{1}{k-2}} \left(k - \frac{k}{k-1}\right)$$
$$= k\left(2 - \left(\frac{2k^2}{k-1}\right)^{\frac{1}{k-2}} \cdot \left(\frac{k-2}{k-1}\right)\right). \tag{4.83}$$

Since $2^{k-2} > \left(\frac{2k^2}{k-1}\right)$ for $k \geq 5$, we get

$$2 > \left(\frac{2k^2}{k-1}\right)^{\frac{1}{k-2}} > \left(\frac{2k^2}{k-1}\right)^{\frac{1}{k-2}} \cdot \left(\frac{k-2}{k-1}\right) \qquad \forall k \geq 5. \tag{4.84}$$

Using (4.84) in (4.83), we obtain

$$g(t) \geq k\left(2 - \left(\frac{2k^2}{k-1}\right)^{\frac{1}{k-2}} \cdot \left(\frac{k-2}{k-1}\right)\right) > 0 \qquad \forall t > 1, k \geq 5. \tag{4.85}$$

Combining (4.81) and (4.85), we obtain $1.5\,(2k-kt) + t^{k-1}/k - (k-1)\cdot t^{k-2}/(2k^3) \geq 0$ for all $t > 1$ and $k \geq 5$. This completes the proof. □

Lemma 52. *Let $k \in \mathbb{N}$ with $k \geq 3$. Then*

$$\left(4k - 1 - k\cdot(2k-1)^{\frac{1}{k}}\right)\left(1 + \frac{1}{2k-1}\right)^{k-1} > \left((2k-1)^{\frac{1}{k}} - \frac{1}{k\cdot(2k-1)}\right)^{k}. \tag{4.86}$$

Proof. Since $k \geq 3$, we have $2k-1 \leq 2^k$, so $(2k-1)^{\frac{1}{k}} \leq 2$. Then

$$4k - 1 - k\cdot(2k-1)^{\frac{1}{k}} \geq 2k - 1. \tag{4.87}$$

Meanwhile,

$$\left((2k-1)^{\frac{1}{k}} - \frac{1}{k\cdot(2k-1)}\right)^{k} = (2k-1)\cdot\left(1 - \frac{1}{k\cdot(2k-1)^{1+\frac{1}{k}}}\right)^{k} < 2k-1. \tag{4.88}$$

Combining (4.87) and (4.88), we obtain

$$\begin{aligned}\left(4k-1-k\cdot(2k-1)^{\frac{1}{k}}\right)\left(1+\frac{1}{2k-1}\right)^{k-1} &> \left(4k-1-k\cdot(2k-1)^{\frac{1}{k}}\right) \\ &\qquad\qquad (\text{Since } 1+1/(2k-1)>1) \\ &\geq 2k-1 \qquad (\text{By } (4.87)) \\ &> \left((2k-1)^{\frac{1}{k}}-\frac{1}{k\cdot(2k-1)}\right)^{k}. \qquad (\text{By } (4.88))\end{aligned}$$

Thus the required inequality (4.87) holds, which completes the proof. □

Lemma 53. *Let $n \geq 2$ and $k \geq 3$, where $k, n \in \mathbb{N}$. Suppose*

$$n^{\frac{1}{k}}-\frac{1}{k}+\frac{2}{n}-\frac{2}{kn}>1+\frac{2k}{n}-\frac{k}{n^{\frac{k-1}{k}}}. \tag{4.89}$$

Then $k < n$.

Proof. We will show the constraint in (4.89) is incompatible with the range $k \geq n$. Let $t = n^{\frac{1}{k}}$. Then $n = t^k$. Since $k \geq n$, we have $t = n^{\frac{1}{k}} \leq k^{\frac{1}{k}}$. We have

$$n^{\frac{1}{k}}-\frac{1}{k}+\frac{2}{n}-\frac{2}{kn}>1+\frac{2k}{n}-\frac{kn^{\frac{1}{k}}}{n},\quad \forall k \geq n \iff \tag{4.90}$$

$$kt^{k+1}-t^k(k+1)+k^2t-2k^2+2k-2>0,\ \forall t \in \left(0, k^{\frac{1}{k}}\right], \tag{4.91}$$

where (4.91) is obtained from (4.90) by multiplying both sides by kn, substituting $n = t^k$, and rearranging.

In order to upper bound the left hand side of (4.91), we define a function $f : [0, \infty) \to \mathbb{R}$ by

$$f(x) = x^k\left(k^{1+\frac{1}{k}}-k-1\right)+k^2x-2k^2+2k-2\,.$$

For $0 \leq t \leq k^{\frac{1}{k}}$, we have $kt^{k+1} \leq kt^k k^{\frac{1}{k}}$, so the left hand side of (4.91) can be upper bounded as follows:

$$\begin{aligned} kt^{k+1} - t^k(k+1) + k^2 t - 2k^2 + 2k - 2 &\leq t^k \left(k^{1+\frac{1}{k}} - k - 1\right) + k^2 t - 2k^2 + 2k - 2 \\ &= f(t)\,. \end{aligned} \tag{4.92}$$

Observe $f'(x) = kx^{k-1}\left(k^{1+\frac{1}{k}} - k - 1\right) + k^2$ and $f''(x) = k(k-1)x^{k-2}\left(k^{1+\frac{1}{k}} - k - 1\right)$. By Lemma 66 the function f is convex for all $k \geq 3$. Thus f has a global maximum on the interval $[0, k^{\frac{1}{k}}]$ which is attained at one of the endpoints. We check the value of the function is negative at both endpoints of $[0, k^{\frac{1}{k}}]$:

- $f(0) = -2k^2 + 2k - 2 = -k^2 - (k-1)^2 - 1 < 0\,.$
- $f(k^{\frac{1}{k}}) = 2k^2 k^{\frac{1}{k}} - 3k^2 + k - 2 < 0$ by Lemma 55.

By convexity, it follows that $f(t) < 0$ for all $t \in [0, k^{\frac{1}{k}}]$. Combining this fact with (4.92), we get

$$kt^{k+1} - t^k(k+1) + k^2 t - 2k^2 + 2k - 2 \leq f(t) < 0 \qquad \forall t \in [0, k^{\frac{1}{k}}], \tag{4.93}$$

which implies (4.91) cannot hold when $k \geq n$. Thus condition (4.89) in the lemma statement rules out the range $k \geq n$. This completes the proof. □

Lemma 54. *Let $k, n \in \mathbb{N}$ such that $k \geq 3$ and $k < n \leq 2k-2$. Then*

$$\left(1 + \frac{1}{n}\right)^{k-1} \geq \mathrm{e}^{\left(\frac{2k-2-n}{n \cdot n^{\frac{1}{k}}}\right)}\,. \tag{4.94}$$

Proof. Taking log on both sides of (4.94), the required inequality is equivalent to

$$\ln\left(1 + \frac{1}{n}\right) \geq \frac{2}{n^{1+\frac{1}{k}}} - \frac{1}{(k-1) \cdot n^{\frac{1}{k}}}\,. \tag{4.95}$$

We first show several independent inequalities and then combine them to obtain the inequality required by the lemma. Recall that $\ln(1+x) \geq \frac{x}{1+x}$ for all $x > -1$ (see, e.g., [141]). Taking $x = 1/n$ yields

$$\ln\left(1+\frac{1}{n}\right) \geq \frac{1/n}{1+1/n} = \frac{1}{n+1}\,. \tag{4.96}$$

Since $n > k \geq 3$, we get $n \geq 3$. By Lemma 66, we obtain $n^{\left(1+\frac{1}{n}\right)} \geq n+1$. Since $k < n$, we get

$$n^{\left(1+\frac{1}{k}\right)} \geq n^{\left(1+\frac{1}{n}\right)} \geq n+1\,. \tag{4.97}$$

Next we will show that

$$\frac{1}{n+1} \geq \frac{2}{n^{1+\frac{1}{k}}} - \frac{1}{(k-1)\cdot n^{\frac{1}{k}}}, \tag{4.98}$$

which is equivalent to

$$(k-1)\cdot n^{1+\frac{1}{k}} \geq 2(k-1)(n+1) - n(n+1)\,. \tag{4.99}$$

By (4.97) we have

$$(k-1)n^{\left(1+\frac{1}{k}\right)} \geq (k-1)(n+1)\,. \tag{4.100}$$

Since $n > k-1$, we have $k-1 > 2(k-1)-n$, which multiplied by $n+1$ on both sides gives

$$(k-1)(n+1) \geq 2(k-1)(n+1) - n(n+1)\,. \tag{4.101}$$

Combining (4.100) and (4.101) yields

$$\begin{aligned}
(k-1)n^{\left(1+\frac{1}{k}\right)} &\geq (k-1)(n+1) && \text{(By (4.100))}\\
&\geq 2(k-1)(n+1) - n(n+1)\,. && \text{(By (4.101))}
\end{aligned}$$

Thus (4.99) holds, so (4.98) holds as well. Combining (4.96) and (4.98) yields

$$\begin{aligned}\ln\left(1+\frac{1}{n}\right) &\geq \frac{1}{n+1} && \text{(By (4.96))}\\ &\geq \frac{2}{n^{1+\frac{1}{k}}} - \frac{1}{(k-1)\cdot n^{\frac{1}{k}}} && \text{(By (4.98))}\end{aligned}$$

Thus (4.95) holds, which is equivalent to the required inequality (4.94). This completes the proof. □

Lemma 55. *Let $k \in \mathbb{N}$ such that $k \geq 3$. Then $2k^2 \cdot k^{\frac{1}{k}} - 3k^2 + k - 2 < 0$.*

Proof. Let $f : (0,\infty) \to \mathbb{R}$ be $f(x) = 2x^2 \cdot x^{\frac{1}{x}} - 3x^2 + x - 2$. We check separately for $k \in \{3,4,5,6\}$:

- $f(3) = 18 \cdot 3^{\frac{1}{3}} - 26 < -0.01 < 0$ and $f(4) = 32 \cdot 4^{\frac{1}{4}} - 46 < -0.7 < 0$.
- $f(5) = 50 \cdot 5^{\frac{1}{5}} - 72 < -3 < 0$ and $f(6) = 72 \cdot 6^{\frac{1}{6}} - 104 < -6 < 0$.

Thus it remains to show the required inequality when $k \geq 7$. The function $x^{\frac{1}{x}}$ has a global maximum at $\mathrm{e}^{\frac{1}{\mathrm{e}}}$ (see, e.g., Wolfram Alpha [140]). Then

$$\begin{aligned}f(x) = 2x^2 \cdot x^{\frac{1}{x}} - 3x^2 + x - 2 &\leq 2x^2 \cdot \mathrm{e}^{\frac{1}{\mathrm{e}}} - 3x^2 + x - 2\\ &< -0.11x^2 + x - 2\\ &< 0 \qquad \forall x \geq 7\,. \end{aligned} \tag{4.102}$$

Thus $f(k) < 0$ for all $k \geq 3, k \in \mathbb{N}$, as required. □

Lemma 56. *Let $k \geq 2, n \geq 1$, $c \in [1/n, 1]$, and the sequence $\{\gamma_\ell\}_{\ell=1}^{\infty}$ with $\gamma_1 = 0$ and $\gamma_\ell = 2\ell$ for $\ell \geq 2$. Then*

$$x\left(1+\frac{\gamma_{k-1}}{n}\right) + (k-1)\cdot\frac{(nc-x)^{\frac{k}{k-1}}}{n\cdot x^{\frac{1}{k-1}}} - \gamma_{k-1}\cdot c \geq c \cdot kn^{\frac{1}{k}} - \gamma_k \cdot c, \qquad \forall x \in (1/2, nc]\,. \tag{4.103}$$

Proof. When $x = nc$, inequality (4.103) is equivalent to

$$nc\left(1+\frac{\gamma_{k-1}}{n}\right) - \gamma_{k-1}\cdot c \geq c\cdot kn^{\frac{1}{k}} - \gamma_k \cdot ct \iff \tag{4.104}$$

$$n - kn^{\frac{1}{k}} \geq -\gamma_k\,. \qquad \text{(Dividing both sides by } c \text{ and re-arranging terms.)}$$

Since $n - kn^{\frac{1}{k}} \geq 1-k$ for all $n \geq 1$, it follows that (4.104) holds if $\gamma_k \geq k-1$, which is the case since $\gamma_k = 2k$. Thus (4.103) holds when $x = nc$.

From now on we can assume $x \in (1/2, nc)$. Let $t = (nc/x-1)^{\frac{1}{k-1}}$. Then $0 < t < (2nc-1)^{\frac{1}{k-1}}$. Equivalently, $x = \frac{nc}{1+t^{k-1}}$, which substituted in (4.103) gives

$$x\left(1+\frac{\gamma_{k-1}}{n}\right) + (k-1)\cdot\frac{(nc-x)^{\frac{k}{k-1}}}{n\cdot x^{\frac{1}{k-1}}} - \gamma_{k-1}\cdot c \geq ckn^{\frac{1}{k}} - \gamma_k\cdot c \iff$$

$$\left(\frac{nc}{1+t^{k-1}}\right)\left(1+\frac{\gamma_{k-1}}{n}\right) + (k-1)\cdot\frac{\left(nc-\frac{nc}{1+t^{k-1}}\right)^{\frac{k}{k-1}}}{n\cdot\left(\frac{nc}{1+t^{k-1}}\right)^{\frac{1}{k-1}}} - \gamma_{k-1}\cdot c \geq ckn^{\frac{1}{k}} - \gamma_k\cdot c\,. \tag{4.105}$$

Multiplying both sides of (4.105) by $\left(1+t^{k-1}\right)/c$ and simplifying, we see (4.105) is equivalent to

$$(k-1)\cdot t^k - t^{k-1}\cdot\left(kn^{\frac{1}{k}} - \gamma_k + \gamma_{k-1}\right) + \left(n - kn^{\frac{1}{k}} + \gamma_k\right) \geq 0\,. \tag{4.106}$$

We will show that (4.106) holds, which will imply inequality (4.103) for all $x \in (1/2, nc)$. We consider two cases, depending on whether $k = 2$ or $k \geq 3$.

Case $k = 2$.

Since $\gamma_1 = 0$ and $\gamma_2 = 4$, inequality (4.106) is equivalent to

$$t^2 - t\left(2\sqrt{n} - 4 + 0\right) + \left(n - 2\sqrt{n} + 4\right) \geq 0 \iff \tag{4.107}$$

$$\left(t - \left(\sqrt{n}-2\right)\right)^2 + 2\sqrt{n} \geq 0, \tag{4.108}$$

where (4.108) was obtained from (4.107) by re-arranging terms. Inequality (4.108) clearly holds, which implies (4.106) and completes the analysis for $k = 2$.

Case $k \geq 3$.

We define a function $h : (0, \infty) \to \mathbb{R}$ to capture the left hand side of (4.106). Then we will show h is non-negative on the entire domain, which will imply (4.106). Let

$$h(t) = (k-1) \cdot t^k - t^{k-1} \cdot \left(kn^{\frac{1}{k}} - \gamma_k + \gamma_{k-1}\right) + \left(n - kn^{\frac{1}{k}} + \gamma_k\right) . \tag{4.109}$$

The first and second derivatives of h are

$$\begin{aligned} h'(t) &= t^{k-1} \cdot k(k-1) - t^{k-2} \cdot (k-1)\left(kn^{\frac{1}{k}} - \gamma_k + \gamma_{k-1}\right) \\ h''(t) &= t^{k-2} \cdot k(k-1)^2 - t^{k-3} \cdot (k-1)(k-2)\left(kn^{\frac{1}{k}} - \gamma_k + \gamma_{k-1}\right) . \end{aligned} \tag{4.110}$$

On $(0, \infty)$ we have:

- the function h' has a unique root at $t_1 = n^{\frac{1}{k}} + \frac{\gamma_{k-1} - \gamma_k}{k}$;
- the function h'' has a unique root at $t_2 = \left(\frac{k-2}{k-1}\right)\left(n^{\frac{1}{k}} + \frac{\gamma_{k-1} - \gamma_k}{k}\right) = \left(\frac{k-2}{k-1}\right) t_1$.

Clearly $t_2 < t_1$. Since $n \geq 1$ and $\gamma_k - \gamma_{k-1} = 2$ when $k \geq 3$, we have

$$n^{\frac{1}{k}} \geq 1 > \frac{2}{k} = \frac{\gamma_k - \gamma_{k-1}}{k} \geq 0 .$$

Thus $n^{\frac{1}{k}} + (\gamma_{k-1} - \gamma_k)/k > 0$, so $t_2 > 0$. We obtain $0 < t_2 < t_1$. Moreover, $h'(t) < 0$ for $t < t_1$ and $h'(t) > 0$ for $t > t_1$; similarly $h''(t) < 0$ for $t < t_2$ and $h''(t) > 0$ for $t > t_2$. Thus h is

- concave and decreasing on $(0, t_2)$;
- convex and decreasing on (t_2, t_1);

- convex and increasing on (t_1, ∞).

Thus h has a unique global minimum at t_1, so the required inequality (4.106) holds if $h(t_1) \geq 0$. We have

$$\begin{aligned} h(t_1) &= (k-1) \cdot t_1^k - t_1^{k-1} \cdot \left(kn^{\frac{1}{k}} - \gamma_k + \gamma_{k-1}\right) + \left(n - kn^{\frac{1}{k}} + \gamma_k\right) \\ &= n - kn^{\frac{1}{k}} + \gamma_k - \left(n^{\frac{1}{k}} + \frac{\gamma_{k-1} - \gamma_k}{k}\right)^k . \end{aligned} \tag{4.111}$$

Since $\gamma_k = 2k$ and $\gamma_{k-1} = 2(k-1)$ for $k \geq 3$, we have

$$n^{\frac{1}{k}} + \frac{\gamma_{k-1} - \gamma_k}{k} = n^{\frac{1}{k}} - \frac{2}{k} \geq 1 - \frac{2}{k} > 0 . \tag{4.112}$$

Using (4.112) in (4.111) gives

$$\begin{aligned} h(t_1) &= n - kn^{\frac{1}{k}} + 2k - \left(n^{\frac{1}{k}} - \frac{2}{k}\right)^k \\ &> 0 . && \text{(By Lemma 57.)} \end{aligned}$$

Thus $h(t_1) \geq 0$, and so inequality (4.106) also holds in the case $k \geq 3$.

Combining the cases.

In both cases $k = 2$ and $k \geq 3$, inequality (4.106) holds, which implies (4.103) for all $x \in (1/2, nc)$. This completes the proof. □

Lemma 57. *Let $n \geq 1$ and $k \geq 3$, where $k, n \in \mathbb{N}$. Then* $n - kn^{\frac{1}{k}} + 2(k-1) - \left(n^{\frac{1}{k}} - \frac{2}{k}\right)^k \geq 0$.

Proof. Define $f : \left[1 - \frac{2}{k}, \infty\right) \to \mathbb{R}$ as

$$f(x) = \left(x + \frac{2}{k}\right)^k - k\left(x + \frac{2}{k}\right) + 2(k-1) - x^k = \left(x + \frac{2}{k}\right)^k - kx - x^k + 2k - 4 . \tag{4.113}$$

The lemma statement requires showing $f\left(n^{\frac{1}{k}} - \frac{2}{k}\right) \geq 0$. We will show that $f(x) \geq 0$ for all $x \geq 1 - 2/k$, which will imply the required inequality. We divide the range of x in two parts and analyze each separately.

Case $x \in \left[1 - \frac{2}{k}, 1\right]$.

We consider a few sub-cases depending on the value of k:

- If $k = 3$, then $f(x) = \left(x + \frac{2}{3}\right)^3 - 3x - x^3 + 2 \cdot 3 - 4 = \frac{1}{27}\left(54x^2 - 45x + 62\right)$. Then $\Delta < 0$, so $f(x) > 0$ for all $x \in \mathbb{R}$.
- If $k \geq 4$, then using the inequalities $1 - 2/k \leq x \leq 1$ in the definition of f from (4.113) gives

$$f(x) \geq 1^k - k \cdot 1 - 1^k + 2k - 4 = k - 4 \geq 0\,. \tag{4.114}$$

Case $x > 1$.

Then

$$\begin{aligned} f(x) &\geq x^k + \binom{k}{1} \cdot x^{k-1} \cdot \frac{2}{k} + \binom{k}{2} \cdot x^{k-2} \cdot \left(\frac{2}{k}\right)^2 - kx - x^k + 2k - 4 \\ &= 2x^{k-1} + \frac{2(k-1)}{k} \cdot x^{k-2} - kx + 2k - 4\,. \end{aligned} \tag{4.115}$$

When $k = 3$, using inequality (4.115), we obtain $f(x) \geq 2x^2 - 5x/3 + 2 \geq 0 \;\; \forall x \in [1, \infty)$.

Thus from now on we can assume $k \geq 4$. Using $x > 1$ and $k \geq 4$, we obtain

$$\begin{aligned} f(x) &\geq 2x^{k-1} + \frac{2(k-1)}{k} \cdot x^{k-2} - kx + 2k - 4 && \text{(By (4.115))} \\ &> 2x^{k-2} - kx + k, && (4.116) \end{aligned}$$

Let $f_1 : (0, \infty) \to \mathbb{R}$ be $f_1(x) = 2x^{k-2} - kx + k$. The derivatives are $f_1'(x) = 2(k-2)x^{k-3} - k$ and $f_1''(x) = 2(k-2)(k-3)x^{k-4}$. Since we are in the case $k > 3$, we have $f_1''(x) > 0$ for $x > 0$. Thus the function f_1 is convex and has a unique global minimum at the point x^* for which $f_1'(x^*) = 0$, that is, at

$$x^* = \left(\frac{k}{2(k-2)}\right)^{\frac{1}{k-3}} . \tag{4.117}$$

Since $k \geq 4$, we have $\frac{k}{2(k-2)} \leq 1$, and so

$$f_1(x^*) = 2\left(\frac{k}{2(k-2)}\right)^{\frac{k-2}{k-3}} - k\left(\frac{k}{2(k-2)}\right)^{\frac{1}{k-3}} + k \geq 2\left(\frac{k}{2(k-2)}\right)^{\frac{k-2}{k-3}} - k \cdot 1 + k > 0\,. \tag{4.118}$$

Combining (4.116) and (4.118) gives $f(x) \geq f_1(x) \geq f_1(x^*) > 0 \;\; \forall x > 0\,$. In particular, the required inequality holds for all $x > 1$, which completes the case.

Combining the cases, we obtain $f(x) \geq 0$ for all $x \geq 1 - 2/k$, and so $f\left(n^{\frac{1}{k}} - 2/k\right) \geq 0$. This completes the proof of the lemma. □

Corollary 11. *For each $n \geq 1$ and $k \geq 3$, we have: $n + 2k > kn^{\frac{1}{k}} + 2\,$.*

Proof. Lemma 57 yields $n+2k \geq kn^{\frac{1}{k}}+2+\left(n^{\frac{1}{k}} - \frac{2}{k}\right)^k$. Since $k \geq 3$, we also have $n^{\frac{1}{k}} \geq 1 > \frac{2}{k}$, so $\left(n^{\frac{1}{k}} - \frac{2}{k}\right)^k > 0$. Thus $n + 2k > kn^{\frac{1}{k}} + 2$, as required. □

4.5 Unordered search Proofs

In this section we include the omitted proofs for unordered search.

4.5.1 Unordered search upper bounds

Here we give the optimal randomized algorithms on a worst case input and deterministic algorithms for any input distribution for unordered search.

Deterministic algorithms for a worst case input.

We start with a simple observation, namely that the optimal k-round deterministic algorithm in the worst case just queries n/k locations in each round.

Observation 4.5.1. *For each $k \in \{1, \dots, n\}$, there is a deterministic k-round algorithm for ordered search that always succeeds and asks at most n queries in the worst case:*

- *In each round $\mathrm{j} \in [k]$, issue $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$ at locations not previously queried. When the item is found, return it and halt.*

Proof. This algorithm queries n locations in the worst case, and so always finds the element using at most n queries. □

Randomized algorithms for a worst case input.

The optimal randomized algorithm is described next.

Proposition 4.3.1 (restated). *Let $p \in (0, 1]$ and $k, n \in \mathbb{N}_{\geq 1}$. Then*

$$\mathcal{R}_{1-p}(\text{unordered}_{n,k}) \leq np \cdot \frac{k+1}{2k} + p + \frac{p}{n} \,.$$

Proof. Consider the following algorithm, which has an all-or-nothing structure.

◇ **With probability** $1 - p$**:** do nothing.

◇ **With probability** p**:** run the following protocol:

- Choose a uniform random permutation $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_n)$ of $[n]$. For each $\mathrm{j} \in [k]$, define

$$m_{\mathrm{j}} = \lceil n \cdot \mathrm{j}/k \rceil \qquad \text{and} \qquad S_{\mathrm{j}} = \{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_{m_{\mathrm{j}}}\} \,.$$

- In each round $\mathrm{j} \in [k]$: query all the locations in S_{j} that have not been queried yet. Whenever the element is found, return its location and halt immediately.

We bound the success probability and the expected number of queries of the algorithm.

Success probability.

If the algorithm finishes execution in exactly $\mathrm{j} \geq 1$ rounds, then the number of queries issued is $|S_{\mathrm{j}}| = \lceil n\mathrm{j}/k \rceil$. By the end of the k-th round, the number of queries issued would be $\lceil nk/k \rceil = n$. Thus if the algorithm enters round 1 then it doesn't stop until finding where the element is, so the success probability is exactly p.

Expected number of queries.

Let A_{j} be the event that the algorithm halts exactly at the end of round j. On event A_{j}, the algorithm issues $\lceil n\mathrm{j}/k \rceil$ queries. The probability of event A_{j} is

$$\Pr(A_{\mathrm{j}}) = p \cdot \frac{\lceil n \cdot \frac{\mathrm{j}}{k} \rceil - \lceil n \cdot \frac{\mathrm{j}-1}{k} \rceil}{n} . \tag{4.119}$$

Then the expected number of queries issued by the algorithm is $q_k = \sum_{\mathrm{j}=1}^{k} \Pr(A_{\mathrm{j}}) \cdot \lceil n\mathrm{j}/k \rceil$. Using (4.119), we can rewrite this as

$$\begin{aligned} q_k &= \sum_{\mathrm{j}=1}^{k} p \cdot \frac{\lceil n \cdot \frac{\mathrm{j}}{k} \rceil - \lceil n \cdot \frac{\mathrm{j}-1}{k} \rceil}{n} \cdot \left\lceil n \cdot \frac{\mathrm{j}}{k} \right\rceil \\ &= np + \frac{p}{n} \cdot \sum_{\mathrm{j}=1}^{k-1} \left(\left\lceil n \cdot \frac{\mathrm{j}}{k} \right\rceil - \left\lceil n \cdot \frac{\mathrm{j}+1}{k} \right\rceil \right) \left\lceil n \cdot \frac{\mathrm{j}}{k} \right\rceil . \end{aligned} \tag{4.120}$$

Applying Lemma 58 with $x = n$ to bound the expression in (4.120) yields

$$q_k \leq np + \frac{p}{n} \cdot \left(-\frac{n^2(k-1)}{2k} + \lceil n \rceil + 1 \right) = np \cdot \frac{k+1}{2k} + p + \frac{p}{n} . \tag{4.121}$$

This completes the proof. □

Deterministic algorithms for a random input.

Given an input distribution $\Psi = (\Psi_1, \dots, \Psi_n)$, we next design an optimal deterministic algorithm for it.

Proposition 4.3.2 (restated). *Let $p \in (0, 1]$ and $k, n \in \mathbb{N}_{\geq 1}$. Then*

$$\mathcal{D}_{1-p}(\text{unordered}_{n,k}) \leq np\left(1 - \frac{k-1}{2k} \cdot p\right) + 1 + p + \frac{2}{n}\,.$$

Proof. Suppose the input distribution is $\Psi = (\Psi_1, \dots, \Psi_n)$. Let $\boldsymbol{\pi}$ be a permutation of $[n]$ such that $\Psi_{\pi_1} \geq \dots \geq \Psi_{\pi_n}$. For each $\mathrm{j} \in [k]$, let $S_\mathrm{j} \subseteq [n]$ be the top $\lceil np \cdot \frac{\mathrm{j}}{k} \rceil$ array positions in the ordering given by $\boldsymbol{\pi}$, that is:

$$S_\mathrm{j} = \left\{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_{m_\mathrm{j}}\right\}, \text{ where } m_\mathrm{j} = \left\lceil np \cdot \frac{\mathrm{j}}{k} \right\rceil\,.$$

Consider the following algorithm.

> *In each round $\mathrm{j} \in [k]$: Query the locations in S_j that have not been queried in the previous $\mathrm{j}-1$ rounds. Once the element is found, return its location and halt immediately.*

Success probability.

To bound the success probability of the algorithm, observe that the subsets S_j are nested, that is: $S_1 \subseteq \dots \subseteq S_k$. By the end of round k, the algorithm has only queried locations from S_k and either found the element or exhausted S_k.

For all $\mathrm{j} \in [k]$, denote the probability that the sought element is in S_j by

$$\phi_\mathrm{j} = \sum_{\ell \in S_\mathrm{j}} \Psi_\ell\,.$$

Lemma 65 gives $\phi_{\mathrm{j}} \geq |S_{\mathrm{j}}|/n$. Then the success probability is $\widetilde{p} = \sum_{\ell \in S_k} \Psi_\ell \geq \frac{|S_k|}{n} = \frac{\lceil np \cdot \frac{k}{k} \rceil}{n} \geq p$.

Expected number of queries.

Next we bound the expected number of queries. For each $\mathrm{j} \in [k]$, let A_{j} be the event that the algorithm halts exactly at the end of round j. On event A_{j}, the algorithm issues a total of $|S_{\mathrm{j}}|$ queries. Moreover, the probability of event A_{j} is

$$\Pr(A_{\mathrm{j}}) = \begin{cases} \phi_{\mathrm{j}} - \phi_{\mathrm{j}-1} & \text{if } 1 \leq \mathrm{j} \leq k-1, \text{ where } \phi_0 = 0\,. \\ 1 - \phi_{k-1} & \text{if } \mathrm{j} = k\,. \end{cases} \tag{4.122}$$

Let $S_0 = \emptyset$. For each $\mathrm{j} \in \{0, \ldots, k\}$, define

$$\eta_{\mathrm{j}} = \phi_{\mathrm{j}} - \frac{|S_{\mathrm{j}}|}{n}\,. \tag{4.123}$$

We have $\eta_{\mathrm{j}} \geq 0$ since $\phi_{\mathrm{j}} \geq |S_{\mathrm{j}}|/n$.

Then the expected number q_k of queries issued on input distribution Ψ can be bounded by:

$$\begin{aligned} q_k &= \sum_{\mathrm{j}=1}^{k} \Pr(A_{\mathrm{j}}) \cdot |S_{\mathrm{j}}| = (1 - \phi_{k-1}) \cdot |S_k| + \sum_{\mathrm{j}=1}^{k-1} (\phi_{\mathrm{j}} - \phi_{\mathrm{j}-1}) \cdot |S_{\mathrm{j}}| && \text{(By (4.122))} \\ &= \left(1 - \frac{|S_{k-1}|}{n} - \eta_{k-1}\right) \cdot |S_k| + \sum_{\mathrm{j}=1}^{k-1} \left(\frac{|S_{\mathrm{j}}|}{n} + \eta_{\mathrm{j}} - \frac{|S_{\mathrm{j}-1}|}{n} - \eta_{\mathrm{j}-1}\right) \cdot |S_{\mathrm{j}}| && \text{(By definition of } \eta_{\mathrm{j}}) \\ &= \left[\left(1 - \frac{|S_{k-1}|}{n}\right) \cdot |S_k| + \sum_{\mathrm{j}=1}^{k-1} \left(\frac{|S_{\mathrm{j}}| - |S_{\mathrm{j}-1}|}{n}\right) \cdot |S_{\mathrm{j}}|\right] - \eta_{k-1} \cdot |S_k| + \sum_{\mathrm{j}=1}^{k-1} (\eta_{\mathrm{j}} - \eta_{\mathrm{j}-1}) \cdot |S_{\mathrm{j}}|\,. \end{aligned} \tag{4.124}$$

We have $0 = |S_0| \leq |S_1| \leq \ldots \leq |S_k|$, and so $\sum_{\mathrm{j}=1}^{k-1} (\eta_{\mathrm{j}} - \eta_{\mathrm{j}-1}) \cdot |S_{\mathrm{j}}| \leq \eta_{k-1} \cdot |S_{k-1}|$. Thus

$$-\eta_{k-1} \cdot |S_k| + \sum_{\mathrm{j}=1}^{k-1} (\eta_{\mathrm{j}} - \eta_{\mathrm{j}-1}) \cdot |S_{\mathrm{j}}| \leq -\eta_{k-1} \cdot |S_k| + \eta_{k-1} \cdot |S_{k-1}| \leq 0\,. \tag{4.125}$$

Using (4.125) in (4.124) gives

$$q_k \leq \left(1 - \frac{|S_{k-1}|}{n}\right) \cdot |S_k| + \sum_{\mathrm{j}=1}^{k-1} \left(\frac{|S_\mathrm{j}| - |S_{\mathrm{j}-1}|}{n}\right) \cdot |S_\mathrm{j}| \,. \tag{4.126}$$

We observe that

$$\left(1 - \frac{|S_{k-1}|}{n}\right) \cdot |S_k| - \sum_{\mathrm{j}=1}^{k-1} \frac{|S_{\mathrm{j}-1}|}{n} \cdot |S_\mathrm{j}| = |S_k| - \sum_{\mathrm{j}=1}^{k-1} \frac{|S_{\mathrm{j}+1}|}{n} \cdot |S_\mathrm{j}| \,. \tag{4.127}$$

Adding $\sum_{\mathrm{j}=1}^{k-1} |S_\mathrm{j}|^2/n$ to both sides of (4.127), we obtain

$$\left(1 - \frac{|S_{k-1}|}{n}\right) \cdot |S_k| + \sum_{\mathrm{j}=1}^{k-1} \left(\frac{|S_\mathrm{j}| - |S_{\mathrm{j}-1}|}{n}\right) \cdot |S_\mathrm{j}| = |S_k| + \sum_{\mathrm{j}=1}^{k-1} \left(\frac{|S_\mathrm{j}| - |S_{\mathrm{j}+1}|}{n}\right) \cdot |S_\mathrm{j}| \,. \tag{4.128}$$

Substituting (4.128) in (4.126) and using the identity $|S_\mathrm{j}| = \lceil np \cdot \mathrm{j}/k \rceil$ gives

$$q_k \leq |S_k| + \sum_{\mathrm{j}=1}^{k-1} \left(\frac{|S_\mathrm{j}| - |S_{\mathrm{j}+1}|}{n}\right) \cdot |S_\mathrm{j}| = \lceil np \rceil + \sum_{\mathrm{j}=1}^{k-1} \left(\frac{\lceil np \cdot \frac{\mathrm{j}}{k} \rceil - \lceil np \cdot \frac{\mathrm{j}+1}{k} \rceil}{n}\right) \cdot \left\lceil np \cdot \frac{\mathrm{j}}{k} \right\rceil \,. \tag{4.129}$$

Applying Lemma 58 with $x = np$ gives

$$\sum_{\mathrm{j}=1}^{k-1} \left(\left\lceil np \cdot \frac{\mathrm{j}}{k} \right\rceil - \left\lceil np \cdot \frac{\mathrm{j}+1}{k} \right\rceil\right) \cdot \left\lceil np \cdot \frac{\mathrm{j}}{k} \right\rceil \leq -\frac{(np)^2(k-1)}{2k} + \lceil np \rceil + 1 \,. \tag{4.130}$$

Combining (4.129) and (4.130) gives:

$$\begin{aligned} q_k &\leq \lceil np \rceil + \frac{1}{n}\left(-\frac{(np)^2(k-1)}{2k} + \lceil np \rceil + 1\right) = np\left(1 - \frac{p(k-1)}{2k}\right) + \frac{\lceil np \rceil}{n} + \frac{1}{n} + \lceil np \rceil - np \\ &\leq np\left(1 - \frac{p(k-1)}{2k}\right) + p + \frac{2}{n} + \lceil np \rceil - np \,. \end{aligned} \tag{4.131}$$

This completes the proof. □

4.5.2 Unordered search lower bounds

In this section we include the unordered search lower bounds.

Proposition 4.3.3 (restated). *Let $p \in (0, 1]$ and $k, n \in \mathbb{N}_{\geq 1}$. Then $\mathcal{R}_{1-p}(unordered_{n,k}) \geq np \cdot \frac{k+1}{2k}$.*

Proof. For proving the required lower bound, it will suffice to assume the input is drawn from the uniform distribution. By an average argument, such a lower bound will also hold for a worst case input.

Let $\mathcal{A}_k$ be a k-round randomized algorithm that succeeds with probability p when facing the uniform distribution as input and denote by $q_k(n, p)$ the expected number of queries asked by $\mathcal{A}_k$ on the uniform distribution.

In round 1, the algorithm has some probability δ_m of asking m queries, for each $m \in \{0, \ldots, n\}$. Moreover, for each such m, there are different (but finitely many) choices for the positions of the m queries of round 1. However, since the goal is to minimize the number of queries, it suffices to restrict attention to the best way of positioning the queries in round 1, breaking ties arbitrarily between different equally good options. For unordered search, each queried location is equivalent to any other since a query only reveals whether the element is there or not.

For each $m \in \{0, \ldots, n\}$, we define the following variables:

- δ_m is the probability that the algorithm asks m queries in the first round.
- α_m is the probability that the algorithm finds the element in one of the rounds in $\{2, \ldots, k\}$, given that it didn't find it in the first round.

The probability of finding the element in the first round is m/n, so the probability that the algorithm may need to continue to one of the rounds in $\{2, \ldots, k\}$ is $(n-m)/n$. The expected number of queries of $\mathcal{A}_k$ on the uniform distribution is

$$q_k(n,p) = \sum_{m=0}^{n} \delta_m \left(m + \left(\frac{n-m}{n} \right) \cdot q_{k-1}(n-m, \alpha_m) \right), \tag{4.132}$$

where the variables are related by the following constraints:

$$\sum_{m=0}^{n} \delta_m = 1 \tag{4.133}$$

$$p_m = \frac{m}{n} + \left(\frac{n-m}{n} \right) \cdot \alpha_m, \qquad \forall m \in \{0, \ldots, n\} \tag{4.134}$$

$$p = \sum_{m=0}^{n} \delta_m \cdot p_m \tag{4.135}$$

$$0 \leq \alpha_m \leq 1, \qquad \forall m \in \{0, \ldots, n\} \tag{4.136}$$

$$\delta_m \geq 0, \qquad \forall m \in \{0, \ldots, n\}. \tag{4.137}$$

Base case.

Proposition 4.4.2 gives $q_1(n,p) \geq np$, as required.

Induction hypothesis.

Suppose $q_\ell(v,s) \geq vs \cdot \frac{\ell+1}{2\ell}$ for all $\ell \in [k-1]$, $v \in \mathbb{N}$, and $s \in [0,1]$.

Induction step.

Using the induction hypothesis in (4.132) gives

$$\begin{aligned} q_k(n,p) &= \sum_{m=0}^{n} \delta_m \left(m + \frac{n-m}{n} \cdot q_{k-1}(n-m, \alpha_m) \right) \\ &\geq \sum_{m=0}^{n} \delta_m \left(m + \frac{(n-m)^2}{n} \cdot \alpha_m \cdot \frac{k}{2k-2} \right). \end{aligned} \tag{4.138}$$

Substituting $\alpha_m = (n \cdot p_m - m)/(n-m)$ from (4.134) in (4.138) gives

$$q_k(n,p) \geq \sum_{m=0}^{n} \delta_m \left(m + \frac{(n-m)(n \cdot p_m - m)}{n} \cdot \frac{k}{2k-2} \right) . \tag{4.139}$$

Lemma 60 gives

$$m + \frac{(n-m)(n \cdot p_m - m)}{n} \cdot \frac{k}{2k-2} \geq n \cdot p_m \left(\frac{k+1}{2k} \right) . \tag{4.140}$$

Using (4.140) in (4.139) gives

$$\begin{aligned} q_k(n,p) &\geq \sum_{m=0}^{n} \delta_m \left(n \cdot p_m \cdot \frac{k+1}{2k} \right) = n \cdot \left(\frac{k+1}{2k} \right) \cdot \sum_{m=0}^{n} \delta_m \cdot p_m \\ &= np \cdot \left(\frac{k+1}{2k} \right) . && \text{(Since } p = \textstyle\sum_{m=0}^{n} \delta_m \cdot p_m \text{ by (4.135))} \end{aligned}$$

□

Next we give the lower bound on the distributional complexity.

Proposition 4.3.4 (restated). *Let $p \in (0,1]$ and $k, n \in \mathbb{N}_{\geq 1}$. Then $\mathcal{D}_{1-p}(unordered_{n,k}) \geq np\left(1 - \frac{k-1}{2k}p\right)$.*

Proof. For each $\ell \in \mathbb{N}$, let $\mathcal{A}_\ell$ be an optimal ℓ-round randomized algorithm that succeeds with probability p when facing the uniform distribution as input. Let $q_\ell(n,p)$ be the expected number of queries of algorithm $\mathcal{A}_\ell$ when given an array of length n.

Since $\mathcal{A}_k$ is deterministic, it asks a fixed number m of queries in round 1. Moreover, since the input is drawn from the uniform distribution, each location is equally likely to contain the answer, and so the actual locations do not matter, but rather only their number. Thus the probability of finding the answer in round 1 is m/n. Let α be the probability that the algorithm finds the element in one of the later rounds in $\{2, \ldots, k\}$, given that the element was not found in the first round.

Given these observations, the expected number of queries of the deterministic algorithm can be written as

$$q_k(n,p) = m + \left(\frac{n-m}{n}\right) \cdot q_{k-1}(n-m,\alpha), \tag{4.141}$$

where the variables are related by the following constraints:

$$\begin{cases} p = \frac{m}{n} + \left(\frac{n-m}{n}\right) \cdot \alpha \\ 0 \leq \alpha \leq 1\,. \end{cases} \tag{4.142}$$

We prove by induction on k that that

$$q_k(n,p) \geq np\left(1 - \frac{k-1}{2k} \cdot p\right)\,. \tag{4.143}$$

Base case.

Proposition 4.4.2 shows that $q_1(n,p) \geq np$.

Induction hypothesis.

Suppose $q_\ell(v,s) \geq vs\left(1 - \frac{\ell-1}{2\ell} \cdot s\right)$ for all $\ell \in [k-1]$, $v \in \mathbb{N}$, and $s \in [0,1]$.

Induction step.

We prove (4.143) holds for k and all $n \in \mathbb{N}, p \in [0,1]$. The induction hypothesis gives

$$q_{k-1}(n-m,\alpha) \geq (n-m)\alpha\left(1 - \frac{k-2}{2k-2} \cdot \alpha\right), \tag{4.144}$$

which substituted in (4.141) yields

$$q_k(n,p) = m + \left(\frac{n-m}{n}\right) \cdot q_{k-1}(n-m,\alpha) \geq m + \frac{\alpha(n-m)^2}{n}\left(1 - \frac{k-2}{2k-2}\cdot\alpha\right) . \tag{4.145}$$

Since $\alpha = (np-m)/(n-m)$ by (4.142), we obtain

$$\begin{aligned} q_k(n,p) &\geq m + \frac{(np-m)\,(n-m)}{n}\left(1 - \frac{k-2}{2k-2}\cdot\left(\frac{np-m}{n-m}\right)\right) \\ &\geq np\left(1 - \frac{k-1}{2k}\cdot p\right) . && \text{(By Lemma 59)} \end{aligned}$$

This completes the induction step and the proof. □

4.5.3 Lemmas for unordered search

In this section we include the lemmas used to prove the unordered search bounds.

Lemma 58. *Let $x \in \mathbb{R}$ and $k \in \mathbb{N}$, where $x, k > 0$. Then*

$$\sum_{\mathrm{j}=1}^{k-1}\left(\left\lceil x\cdot\frac{\mathrm{j}}{k}\right\rceil - \left\lceil x\cdot\frac{\mathrm{j}+1}{k}\right\rceil\right)\left\lceil x\cdot\frac{\mathrm{j}}{k}\right\rceil \leq -\frac{x^2(k-1)}{2k} + \lceil x\rceil + 1 . \tag{4.146}$$

Proof. For every $\mathrm{j} \in [k]$, let $b_\mathrm{j} = \lceil x\mathrm{j}/k\rceil - x\mathrm{j}/k$. The left hand side of (4.146) can be rewritten as

$$\sum_{\mathrm{j}=1}^{k-1}\left(\left\lceil x\cdot\frac{\mathrm{j}}{k}\right\rceil - \left\lceil x\cdot\frac{\mathrm{j}+1}{k}\right\rceil\right)\left\lceil x\cdot\frac{\mathrm{j}}{k}\right\rceil = \sum_{\mathrm{j}=1}^{k-1}\left(x\cdot\frac{\mathrm{j}}{k} + b_\mathrm{j} - x\cdot\frac{\mathrm{j}+1}{k} - b_{\mathrm{j}+1}\right)\left(x\cdot\frac{\mathrm{j}}{k} + b_\mathrm{j}\right) \tag{4.147}$$

$$= \sum_{\mathrm{j}=1}^{k-1}\left(-\frac{x^2\mathrm{j}}{k^2} + b_\mathrm{j}\left(b_\mathrm{j} - b_{\mathrm{j}+1}\right) + \frac{x}{k}\Big(\mathrm{j}(b_\mathrm{j} - b_{\mathrm{j}+1}) - b_\mathrm{j}\Big)\right) . \tag{4.148}$$

The last term of the sum in (4.148) almost entirely cancels:

$$\sum_{\mathrm{j}=1}^{k-1} \frac{x}{k} \cdot \Big(\mathrm{j}\left(b_{\mathrm{j}} - b_{\mathrm{j}+1}\right) - b_{\mathrm{j}}\Big) = -b_k \cdot \frac{x(k-1)}{k} \leq 0\,. \tag{4.149}$$

Combining (4.148) with (4.149), we get

$$\begin{aligned}\sum_{\mathrm{j}=1}^{k-1} \left(\left\lceil x \cdot \frac{\mathrm{j}}{k}\right\rceil - \left\lceil x \cdot \frac{\mathrm{j}+1}{k}\right\rceil\right) \left\lceil x \cdot \frac{\mathrm{j}}{k}\right\rceil &\leq \sum_{\mathrm{j}=1}^{k-1} \left(-\frac{x^2 \mathrm{j}}{k^2} + b_{\mathrm{j}}\left(b_{\mathrm{j}} - b_{\mathrm{j}+1}\right)\right) \\ &= -\frac{x^2(k-1)}{2k} + \sum_{\mathrm{j}=1}^{k-1} b_{\mathrm{j}}\left(b_{\mathrm{j}} - b_{\mathrm{j}+1}\right)\,. \end{aligned} \tag{4.150}$$

Next we bound the summation term in (4.150). If $b_{\mathrm{j}} \geq b_{\mathrm{j}+1} \geq b_{\mathrm{j}+2}$ for some $\mathrm{j} \in [k-2]$, then

$$b_{\mathrm{j}}(b_{\mathrm{j}} - b_{\mathrm{j}+1}) + b_{\mathrm{j}+1}(b_{\mathrm{j}+1} - b_{\mathrm{j}+2}) \leq b_{\mathrm{j}}(b_{\mathrm{j}} - b_{\mathrm{j}+2})\,. \tag{4.151}$$

Thus if there is a (weakly) decreasing sequence $b_{\mathrm{j}} \geq b_{\mathrm{j}+1} \geq \ldots \geq b_{\mathrm{j}+t}$ for some $t \geq 2$ and $\mathrm{j} \in [k-t]$, then applying inequality (4.151) iteratively gives

$$\sum_{\mathrm{i}=\mathrm{j}}^{\mathrm{j}+t-1} b_{\mathrm{i}}\left(b_{\mathrm{i}} - b_{\mathrm{i}+1}\right) \leq b_{\mathrm{j}}\left(b_{\mathrm{j}} - b_{\mathrm{j}+t}\right)\,. \tag{4.152}$$

We will use inequality (4.151) to collapse some of the terms in the sum $\sum_{\mathrm{j}=1}^{k-1} b_{\mathrm{j}}(b_{\mathrm{j}} - b_{\mathrm{j}+1})$.

Towards this end, let $G = ([k], E)$ be a line graph where the vertices are $\{1, \ldots, k\}$ and the edges $E = \{(\mathrm{j}, \mathrm{j}+1) \mid \mathrm{j} \in [k-1]\}$. For each $\mathrm{j} \in [k-1]$, if $b_{\mathrm{j}} \geq b_{\mathrm{j}+1}$ then edge $(\mathrm{j}, \mathrm{j}+1)$ is colored with black and depicted as oriented down, and otherwise it is colored with yellow and oriented up.

We also give each vertex $\mathrm{j} \in [k]$ a color $c_{\mathrm{j}} \in \{R, B\}$, such that $c_1 = c_k = R$. Furthermore, for each $\mathrm{j} \in [k-1]$, if $b_{\mathrm{j}} < b_{\mathrm{j}+1}$ then both endpoints of the edge are colored red: $c_{\mathrm{j}} = c_{\mathrm{j}+1} = R$. All other vertices are colored blue (B). See Figure 4.6 for an illustration.

Let $\ell_1 = 1 < \ldots < \ell_m = k$ be the red vertices in G and $L = \{\ell_1, \ldots, \ell_m\}$. For all $\mathrm{i} \in [m-1]$:

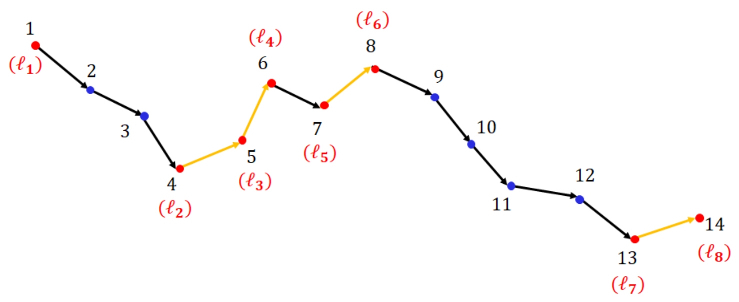


Figure 4.6. Given $k \geq 2$ and numbers $b_1, \ldots, b_k \in [0,1)$, we construct a graph with edges $(\mathrm{j}, \mathrm{j}+1)$ for each $\mathrm{j} \in [k-1]$. For each $\mathrm{j} \in [k]$, if $b_\mathrm{j} \geq b_{\mathrm{j}+1}$, the edge from j to $\mathrm{j}+1$ is oriented downwards and is colored with black. If $b_\mathrm{j} < b_{\mathrm{j}+1}$, the edge from j to $\mathrm{j}+1$ is oriented upwards and is colored with yellow. The endpoints of all the yellow edges are added to the set L, together with special vertices 1 and k. All the vertices in L are colored red and the vertices in $[k] \setminus L$ are colored blue. For the graph in the picture we have $k = 14$ and $L = \{1,4,5,6,7,8,13,14\}$. Each element ℓ_j of L is marked in red near the corresponding node.

- if the path from ℓ_i to $\ell_{\mathrm{i}+1}$ has black edges, then $b_{\ell_\mathrm{i}} \geq \ldots \geq b_{\ell_{\mathrm{i}+1}}$ and so inequality (4.151) gives

$$\sum_{\mathrm{j}=\ell_\mathrm{i}}^{\ell_{\mathrm{i}+1}-1} b_\mathrm{j}(b_\mathrm{j} - b_{\mathrm{j}+1}) \leq b_{\ell_\mathrm{i}} \left(b_{\ell_\mathrm{i}} - b_{\ell_{(\mathrm{i}+1)}} \right) . \tag{4.153}$$

- else, the path from ℓ_i to $\ell_{\mathrm{i}+1}$ has no black edges. Then $\ell_{\mathrm{i}+1} = \ell_\mathrm{i} + 1$, and so the next inequality trivially holds:

$$b_{\ell_\mathrm{i}} \left(b_{\ell_\mathrm{i}} - b_{\ell_{(\mathrm{i}+1)}} \right) \leq b_{\ell_\mathrm{i}} \left(b_{\ell_\mathrm{i}} - b_{\ell_{(\mathrm{i}+1)}} \right) . \tag{4.154}$$

Combining (4.153) and (4.154), we can bound the sum of all b_j's as follows:

$$\sum_{\mathrm{j}=1}^{k-1} b_\mathrm{j} \left(b_\mathrm{j} - b_{\mathrm{j}+1} \right) \leq \sum_{\mathrm{i}=1}^{m-1} b_{\ell_\mathrm{i}} \left(b_{\ell_\mathrm{i}} - b_{\ell_{(\mathrm{i}+1)}} \right) . \tag{4.155}$$

Since $b_{\mathrm{j}} \in [0,1)$ for all j, we have $b_{\ell_{\mathrm{i}}}\left(b_{\ell_{\mathrm{i}}} - b_{\ell_{\mathrm{i}+1}}\right) \leq (1 - b_{\ell_{\mathrm{i}+1}})$ and $b_{\ell_{\mathrm{i}+1}}\left(b_{\ell_{\mathrm{i}+1}} - b_{\ell_{\mathrm{i}+2}}\right) \leq b_{\ell_{\mathrm{i}+1}}$. Thus adjacent terms in (4.155) sum to at most 1. Then

- If $m-1$ is even, then

$$\sum_{\mathrm{i}=1}^{m-1} b_{\ell_{\mathrm{i}}}\left(b_{\ell_{\mathrm{i}}} - b_{\ell_{(\mathrm{i}+1)}}\right) \leq \frac{m-1}{2} < \frac{m}{2}\,. \tag{4.156}$$

- If $m-1$ is odd, then

$$\sum_{\mathrm{i}=1}^{m-1} b_{\ell_{\mathrm{i}}}\left(b_{\ell_{\mathrm{i}}} - b_{\ell_{(\mathrm{i}+1)}}\right) \leq \left\lfloor \frac{m-1}{2} \right\rfloor + b_{\ell_{\mathrm{i}}}\left(b_{\ell_{\mathrm{i}}} - b_{\ell_{(\mathrm{i}+1)}}\right) \leq \left\lfloor \frac{m-1}{2} \right\rfloor + 1 = \frac{m}{2}\,. \tag{4.157}$$

Combining (4.156) and (4.157), we obtain

$$\sum_{\mathrm{i}=1}^{m-1} b_{\ell_{\mathrm{i}}}\left(b_{\ell_{\mathrm{i}}} - b_{\ell_{(\mathrm{i}+1)}}\right) \leq \frac{m}{2}\,. \tag{4.158}$$

Combining (4.155) with (4.158) while summing over all $\mathrm{j} \in [k-1]$ gives

$$\sum_{\mathrm{j}=1}^{k-1} b_{\mathrm{j}}\left(b_{\mathrm{j}} - b_{\mathrm{j}+1}\right) \leq \frac{m}{2}\,. \tag{4.159}$$

Let $D = \{\mathrm{j} \in [k-1] \mid b_{\mathrm{j}} < b_{\mathrm{j}+1}\}$ and $\Delta = |D|$. Since $b_{\mathrm{j}} = \lceil x\mathrm{j}/k \rceil - x\mathrm{j}/k$, we have $b_{\mathrm{j}} \leq b_{\mathrm{j}+1}$ if and only if $\lceil x\mathrm{j}/k \rceil - x\mathrm{j}/k \leq \lceil x(\mathrm{j}+1)/k \rceil - x(\mathrm{j}+1)/k$ (†). Since $x/k > 0$, inequality (†) implies

$$\lceil x\mathrm{j}/k \rceil + 1 \leq \lceil x(\mathrm{j}+1)/k \rceil \qquad \forall \mathrm{j} \in D\,. \tag{4.160}$$

Consider the elements of D in sorted order: $d_1 < \ldots < d_\Delta$. We will show by induction that

$$\lceil x \cdot d_{\mathrm{i}}/k \rceil \geq \mathrm{i} \;\text{ for all } \mathrm{i} \in [\Delta]\,. \tag{4.161}$$

The base case is $\mathrm{i} = 1$. Indeed $\lceil x \cdot d_1/k \rceil \geq 1$ since $x \cdot d_1/k > 0$. We assume inequality (4.161) holds for i and show this implies the inequality for $\mathrm{i}+1$. We have

$$\begin{aligned}
\left\lceil x \cdot \frac{d_{\mathrm{i}+1}}{k} \right\rceil &\geq \left\lceil x \cdot \frac{d_{\mathrm{i}}+1}{k} \right\rceil && \text{(Since } d_{\mathrm{i}+1} > d_{\mathrm{i}} \text{ and } d_{\mathrm{i}}, d_{\mathrm{i}+1} \in \mathbb{N}.\text{)} \\
&\geq \left\lceil x \cdot \frac{d_{\mathrm{i}}}{k} \right\rceil + 1 && \text{(By (4.160).)} \\
&\geq \mathrm{i}+1\,. && \text{(By the inductive hypothesis.)}
\end{aligned}$$

This completes the induction, so (4.161) holds. Now we can bound the size of D. Since $d_\Delta \in [k-1]$, we have $\lceil x \cdot k/k \rceil \geq \lceil x \cdot d_\Delta/k \rceil$. By (4.161), we have $\lceil x \cdot d_\Delta/k \rceil \geq \Delta$. Thus

$$\lceil x \rceil = \left\lceil x \cdot \frac{k}{k} \right\rceil \geq \left\lceil x \cdot \frac{d_\Delta}{k} \right\rceil \geq \Delta\,. \tag{4.162}$$

Observe that Δ is equal to the number of yellow edges in the graph G, since $\mathrm{j} \in D$ if and only if the edge $(\mathrm{j}, \mathrm{j}+1)$ is yellow. Thus the number of endpoints of yellow edges in G is at most 2Δ. Since $|L| = m$ and L consists precisely of all the endpoints of yellow edges together with vertices 1 and k, we have $m = |L| \leq |\{1, k\}| + 2\Delta = 2 + 2\Delta$. Since $\Delta \leq \lceil x \rceil$ by (4.162), we obtain

$$m \leq 2 + 2\Delta \leq 2 + 2\lceil x \rceil\,. \tag{4.163}$$

Combining (4.163) with (4.159), we get

$$\sum_{\mathrm{j}=1}^{k-1} b_{\mathrm{j}} \left(b_{\mathrm{j}} - b_{\mathrm{j}+1} \right) \leq \lceil x \rceil + 1\,. \tag{4.164}$$

Combining (4.164) with (4.150) gives

$$\sum_{\mathrm{j}=1}^{k-1} \left(\left\lceil x \cdot \frac{\mathrm{j}}{k} \right\rceil - \left\lceil x \cdot \frac{\mathrm{j}+1}{k} \right\rceil \right) \left\lceil x \cdot \frac{\mathrm{j}}{k} \right\rceil \leq -\frac{x^2(k-1)}{2k} + \sum_{\mathrm{j}=1}^{k-1} b_{\mathrm{j}} \left(b_{\mathrm{j}} - b_{\mathrm{j}+1} \right) \leq -\frac{x^2(k-1)}{2k} + \lceil x \rceil + 1\,.$$

This completes the proof. □

Lemma 59. *Let $k, n \in \mathbb{N}$, $x \in [0, n]$, and $p \in [0, 1]$. Suppose $k \geq 2$. Then*

$$x + \frac{(np - x)(n - x)}{n} \left(1 - \frac{k-2}{2k-2} \cdot \frac{np - x}{n - x}\right) \geq np \left(1 - \frac{k-1}{2k} \cdot p\right) . \tag{4.165}$$

Proof. Let $f : \mathbb{R} \to \mathbb{R}$ be

$$f(x) = x + \frac{(np - x)(n - x)}{n} \left(1 - \frac{k-2}{2k-2} \cdot \frac{np - x}{n - x}\right) - np \left(1 - \frac{k-1}{2k} \cdot p\right) . \tag{4.166}$$

Then the required inequality (4.165) is equivalent to showing $f(x) \geq 0$ for all $x \in [0, n]$.

Expanding the terms in the expression for $f(x)$, we get

$$\begin{aligned} & f(x) \geq 0 \iff \\ & x + np - x - \frac{x(np - x)}{n} - \left(\frac{k-2}{2k-2}\right) \frac{(np - x)^2}{n} - np + np^2 \left(\frac{k-1}{2k}\right) \geq 0, \end{aligned} \tag{4.167}$$

which after simplification is equivalent to

$$x^2 k^2 - x \cdot 2knp + n^2 p^2 \geq 0 . \tag{4.168}$$

The quadratic equation in (4.168) has a unique global minimum at $x^* = np/k$, with $f(x^*) = 0$. Thus (4.168) holds, so (4.167) holds and so $f(x) \geq f(x^*) = 0$ $\forall x \in [0, n]$ as required. $\square$

Lemma 60. *Let $k, n, m \in \mathbb{N}$, where $k \geq 2$, $n \geq 1$, and $m \in \{0, \ldots, n\}$. Let $\gamma \in [m/n, 1]$. Then*

$$m + \frac{(n - m)(n \cdot \gamma - m)}{n} \cdot \frac{k}{2k-2} - n \cdot \gamma \left(\frac{k+1}{2k}\right) \geq 0 . \tag{4.169}$$

Proof. Inequality (4.169) is equivalent to

$$m + n\gamma \cdot \frac{k}{2k-2} - m \cdot \frac{k}{2k-2} - m\gamma \cdot \frac{k}{2k-2} + \frac{m^2}{n} \cdot \frac{k}{2k-2} - n \cdot \gamma \cdot \frac{k+1}{2k} \geq 0 . \tag{4.170}$$

Multiplying both sides of (4.170) by $2nk(k-1)$ and rearranging, we get that (4.169) is equivalent to

$$\gamma n(n - mk^2) + mn(k^2 - 2k) + m^2k^2 \geq 0 \,. \tag{4.171}$$

If $n \geq mk^2$ then (4.171) clearly holds. Else, assume $n < mk^2$. Since $m/n \leq \gamma \leq 1$, we have

$$\gamma n(n - mk^2) \geq n(n - mk^2) \,. \tag{4.172}$$

Using (4.172), we can bound the left hand side of (4.171) as follows:

$$\gamma n(n - mk^2) + mn(k^2 - 2k) + m^2k^2 \geq n(n - mk^2) + mn(k^2 - 2k) + m^2k^2 = (n - mk)^2 \geq 0 \,.$$

Thus (4.171) holds when $n < mk^2$ as well, which implies (4.169) holds in all cases, as required. $\square$

4.6 Cake cutting and sorting in rounds proofs

In this section we study cake cutting in rounds and discuss the connection between sorting with rank queries and proportional cake cutting. We first introduce the cake cutting model.

Cake cutting model.

The resource (cake) is represented as the interval $[0, 1]$. There is a set of players $N = \{1, \ldots, n\}$, such that each player $\mathrm{i} \in N$ is endowed with a private *valuation function* V_i that assigns a value to every subinterval of $[0, 1]$. These values are induced by a non-negative integrable *value density function* v_i, so that for an interval I, $V_\mathrm{i}(I) = \int_{x \in I} v_\mathrm{i}(x)\, dx$. The valuations are additive, so $V_\mathrm{i}\left(\bigcup_{\mathrm{j}=1}^m I_\mathrm{j}\right) = \sum_{\mathrm{j}=1}^m V_\mathrm{i}(I_\mathrm{j})$ for any disjoint intervals $I_1, \ldots, I_m \subseteq [0, 1]$. The value densities are non-atomic, and sets of measure zero are worth zero to a player. W.l.o.g., the valuations are normalized to $V_\mathrm{i}([0, 1]) = 1$, for all $\mathrm{i} = 1 \ldots n$.

A *piece of cake* is a finite union of disjoint intervals. A piece is *connected* (or contiguous) if it consists of a single interval. An *allocation* $A = (A_1, \ldots, A_n)$ is a partition of the cake among the players, such that each player i receives the piece A_{i}, the pieces are disjoint, and $\bigcup_{\mathrm{i} \in N} A_{\mathrm{i}} = [0, 1]$. An allocation A is said to be *proportional* if $V_{\mathrm{i}}(A_{\mathrm{i}}) \geq 1/n$ for all $\mathrm{i} \in N$.

Query complexity of cake cutting.

All the discrete cake cutting protocols operate in a query model known as the Robertson-Webb model (see, e.g., the book of [91]), which was explicitly stated by [63]. In this model, the protocol communicates with the players using the following types of queries:

- ***Cut***$_{\mathrm{i}}(\alpha)$: Player i cuts the cake at a point y where $V_{\mathrm{i}}([0, y]) = \alpha$, where $\alpha \in [0, 1]$ is chosen arbitrarily by the center [3]. The point y becomes a *cut point*.
- ***Eval***$_{\mathrm{i}}(y)$: Player i returns $V_{\mathrm{i}}([0, y])$, where y is a previously made cut point.

An RW protocol asks the players a sequence of cut and evaluate queries, at the end of which it outputs an allocation demarcated by cut points from its execution (i.e. cuts discovered through queries). Note that the value of a piece $[x, y]$ can be determined with two Eval queries, Eval$_{\mathrm{i}}(x)$ and Eval$_{\mathrm{i}}(y)$.

When a protocol runs in k rounds, then multiple RW queries (to the same or different agents) can be issued at once in each round. Note the choice of queries submitted in round j cannot depend on the results of queries from the same or later rounds (i.e. $\mathrm{j}, \mathrm{j}+1, \ldots, k$).

4.6.1 Upper bounds

We will devise a protocol that finds a proportional allocation of the cake in k rounds of interaction, which will also give a protocol for sorting with rank queries. For the special case of one round, a proportional protocol was studied in [111, 142]. Our high level approach is to iteratively divide the cake into subcakes and assign agents to each subcake.

[3]↑Ties are resolved deterministically, using for example the leftmost point with this property.

Proposition 4.6.1. *There is an algorithm that runs in k rounds and computes a proportional allocation with a total of $O(kn^{1+1/k})$ RW queries.*

We first describe the algorithm, and then prove Proposition 4.6.1. The idea behind the algorithm is to partition the cake into $n^{1/k}$ subcakes and assign $n^{1-1/k}$ agents to each section, such that every agent believes that if they ultimately get a proportional share of their subcake, then they will have a proportional slice overall. Then all that remains is to recurse on each subcake in parallel in the successive rounds.

One complication is that our only method of asking agents to cut a subcake, the *Cut* query, requires that we know the values of the boundary of the subcake to that agent. However, the boundaries of the subcakes are known only with respect to one agent (possibly different agents for each boundary). We circumvent this difficulty by instead asking each agent to divide a further subset of their subcake whose boundary values for their valuation are known. In Algorithm 1, this further subset for each agent i is the interval $[\mathrm{Cut}_{\mathrm{i}}(a_{\mathrm{i}}), \mathrm{Cut}_{\mathrm{i}}(b_{\mathrm{i}}))$.

Algorithm 1.

Input:

- Cake interval $[x, y]$ to be divided.
- Agent set A among whom the cake is to be allocated.
- For each agent $\mathrm{i} \in A$, values a_{i} and b_{i} in $[0, 1]$.
- Number of remaining rounds k.

Procedure:

1. If $|A| = 1$, allocate the whole interval to the sole agent. Otherwise, continue.
2. Define $z = \lceil |A|^{1/k} \rceil$ and define $m_{\mathrm{j}} = \lceil |A| \cdot \frac{\mathrm{j}}{z} \rceil - \lceil |A| \cdot \frac{\mathrm{j}-1}{z} \rceil$ for each $\mathrm{j} \in [z]$.
3. Query $\mathrm{Cut}_{\mathrm{i}}\left(a_{\mathrm{i}} + (b_{\mathrm{i}} - a_{\mathrm{i}}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}} m_\ell\right)$ for all agents $\mathrm{i} \in A$ and all $\mathrm{j} \in [z - 1]$.

4. For $\mathrm{j} = 1, 2, \ldots, z-1$:

   (a) Select S_{j} to be the m_{j} agents i among $A \setminus \left(\bigcup_{\ell=1}^{\mathrm{j}-1} S_\ell\right)$ with the smallest values for $\mathrm{Cut}_{\mathrm{i}}\left(a_{\mathrm{i}} + (b_{\mathrm{i}} - a_{\mathrm{i}}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}} m_\ell\right)$.

   (b) Set c_{j} to be the m_{j}th smallest value for $\mathrm{Cut}_{\mathrm{i}}\left(a_{\mathrm{i}} + (b_{\mathrm{i}} - a_{\mathrm{i}}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}} m_\ell\right)$ among all $\mathrm{i} \in A \setminus \left(\bigcup_{\ell=1}^{\mathrm{j}-1} S_\ell\right)$.

5. Set $S_z = A \setminus \left(\bigcup_{\ell=1}^{z-1} S_\ell\right)$, $c_0 = 0$, and $c_z = 1$.

6. In parallel in the following rounds, recurse on the the following instance for each $\mathrm{j} \in [z]$:

   - The cake interval to be divided is $[c_{\mathrm{j}-1}, c_{\mathrm{j}}]$.
   - The set of agents is S_{j}.
   - For each agent $\mathrm{i} \in S_{\mathrm{j}}$, set $new(a_{\mathrm{i}}) = a_{\mathrm{i}} + (b_{\mathrm{i}} - a_{\mathrm{i}}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}-1} m_\ell$.
   - For each agent $\mathrm{i} \in S_{\mathrm{j}}$, set $new(b_{\mathrm{i}}) = a_{\mathrm{i}} + (b_{\mathrm{i}} - a_{\mathrm{i}}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}} m_\ell$.
   - The number of remaining rounds is $k - 1$.

To initially run Algorithm 1, use as input the following parameters. The cake interval to be divided is $[0, 1]$. The set of agents is $[n]$. For each agent $\mathrm{i} \in N$, set $a_{\mathrm{i}} = 0$ and $b_{\mathrm{i}} = 1$. The number of (remaining) rounds is k.

Example of running Algorithm 1.

(173) Let $n = 4$ and $k = 2$. Let the agents' value densities be as shown in Figure 4.7. After the first round we will have

- $\mathrm{Cut}(a_1) = 0.65, \mathrm{Cut}(b_1) = 1, \mathrm{Cut}(a_2) = 0.5, \mathrm{Cut}(b_2) = 1, \mathrm{Cut}(a_3) = 0, \mathrm{Cut}(b_3) = 0.45$,
  $\mathrm{Cut}(a_4) = 0, \mathrm{Cut}(b_4) = 0.4$.

The dividing line between the two subcakes, i.e. c_1, will be $\mathrm{Cut}(a_3) = 0.5$.

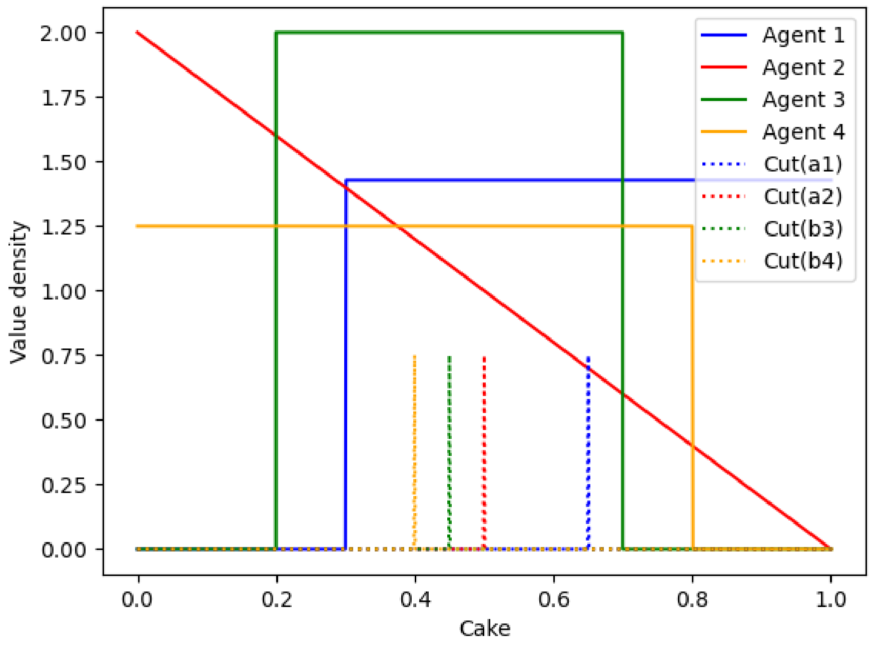


Figure 4.7. A potential value distribution for four agents from Example 1. When running Algorithm 1 on the shown value density functions with $k = 2$, after the first round we will have $a_1 = a_2 = 0.5$, $b_1 = b_2 = 1$, $a_3 = a_4 = 0$, and $b_3 = b_3 = 0.5$. This leads to the Cut values shown. In the second round, Algorithm 1 will recurse on the subcake $[0, \mathrm{Cut}(a_3)) = [0, 0.5]$ with agents 3 and 4 and on the subcake $[0.5, 1]$ with agents 1 and 2.

(174) Let $n = 1000$ and $k = 3$. Algorithm 1 works as follows in each round:

1. Round 1: everyone is asked to mark their $\frac{1}{10}, \frac{2}{10}, \ldots, \frac{9}{10}$ points. These are used to separate the agents into 10 subcakes, each containing 100 agents.

2. Round 2: Everyone is asked to mark their $\frac{1}{10}, \frac{2}{10}, \ldots, \frac{9}{10}$ points within their respective value interval $[a_i, b_i]$. For example, for the second subcake each agent marks their $\frac{11}{100}, \frac{12}{100}, \ldots, \frac{19}{100}$ points. Again these are used to separate each subcake further into 10 subcakes, each containing 10 agents.

3. Round 3: Everyone is asked to mark their $1/10, \ldots, 9/10$ points within their respective value interval. This time when assigning agents to subcakes, the algorithm assigns only 1 to each, so we're done.

Next we prove that the algorithm correctly computes a proportional allocation of the cake in k rounds.

Proof of Proposition 4.6.1. Consider Algorithm 1. We claim that after j rounds each subcake contains at most $n^{1-\mathrm{j}/k}$ agents. In the base case, after 0 rounds, the sole subcake contains all n agents. In the inductive case, we assume that after j rounds each subcake contains at most $n^{1-\mathrm{j}/k}$ agents. Consider an arbitrary subcake containing m agents and an arbitrary ℓ. Then

$$m_\ell = \lceil m \cdot \frac{\mathrm{j}}{\lceil n^{1/k} \rceil} \rceil - \lceil m \cdot \frac{\mathrm{j}-1}{\lceil n^{1/k} \rceil} \rceil \leq m^{1-1/(k-\mathrm{j})} \leq n^{1-\frac{\mathrm{j}+1}{k}} \tag{4.175}$$

This concludes the induction. Then after k rounds each subcake contains at most $n^{1-k/k} = 1$ agents. Thus the algorithm generates an allocation in k rounds.

Next we claim inductively that at the start of every call to Algorithm 1, for all $\mathrm{i} \in N$ we have $x \leq \mathrm{Cut}_\mathrm{i}(a_\mathrm{i})$ and $\mathrm{Cut}_\mathrm{i}(b_\mathrm{i}) \leq y$. In the initial call to Algorithm 1 this is true since $0 \leq \mathrm{Cut}_\mathrm{i}(0)$ and $\mathrm{Cut}_\mathrm{i}(1) \leq 1$. In the recursive call in step 6, consider an arbitrary $\mathrm{j} \in [z]$ and an arbitrary agent $\mathrm{i} \in S_\mathrm{j}$. If $\mathrm{j} = 1$, then $c_{\mathrm{j}-1} = x \leq a_\mathrm{i} = a_\mathrm{i} + a_\mathrm{i} + (b_\mathrm{i} - a_\mathrm{i}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}-1} m_\ell = new(a_\mathrm{i})$ by inductive assumption. If instead $\mathrm{j} > 1$, then because $\mathrm{i} \notin S_{\mathrm{j}-1}$, we know by definition of c_j that

$$c_{\mathrm{j}-1} \leq \mathrm{Cut}_\mathrm{i}\left(a_\mathrm{i} + (b_\mathrm{i} - a_\mathrm{i}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}} m_\ell\right) = \mathrm{Cut}_\mathrm{i}(new(a_\mathrm{i})) \,. \tag{4.176}$$

Either way, in the recursive call we have $x = c_{\mathrm{j}-1} \leq \mathrm{Cut}_{\mathrm{i}}(new(a_{\mathrm{i}}))$. If $\mathrm{j} = z$ then $c_{\mathrm{j}} = y \geq b_{\mathrm{i}} = a_{\mathrm{i}} + (b_{\mathrm{i}} - a_{\mathrm{i}}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}} m_\ell = new(b_{\mathrm{i}})$ by inductive assumption. And if instead $\mathrm{j} < z$, then because $\mathrm{i} \in S_{\mathrm{j}}$, we know by definition of c_{j} that

$$c_{\mathrm{j}} \geq \mathrm{Cut}_{\mathrm{i}}(a_{\mathrm{i}} + (b_{\mathrm{i}} - a_{\mathrm{i}}) \cdot \frac{1}{|A|} \sum_{\ell=1}^{\mathrm{j}} m_\ell) = \mathrm{Cut}_{\mathrm{i}}(new(b_{\mathrm{i}}))\,. \tag{4.177}$$

Therefore $x \leq \mathrm{Cut}_{\mathrm{i}}(a_{\mathrm{i}})$ and $\mathrm{Cut}_{\mathrm{i}}(b_{\mathrm{i}}) \leq y$ at the start of every call to Algorithm 1.

To argue that every agent receives value at least $1/n$, we proceed by induction on k. In particular, we claim that at the start of every call to Algorithm 1, every agent i has $b_{\mathrm{i}} - a_{\mathrm{i}} \geq |A|/n$. In the initial call to Algorithm 1 this is true since $1 - 0 = n/n$. In the recursive call in step 6, consider an arbitrary $\mathrm{j} \in [z]$ and an arbitrary agent i. Because $x \leq \mathrm{Cut}_{\mathrm{i}}(a_{\mathrm{i}})$ and $\mathrm{Cut}_{\mathrm{i}}(b_{\mathrm{i}}) \leq y$ at the start of each call to Algorithm 1, we have that agent i values $[x, y]$ as at least $b_{\mathrm{i}} - a_{\mathrm{i}}$. By definition of $new(a_{\mathrm{i}})$ and $new(b_{\mathrm{i}})$ in step 6, we have by inductive assumption

$$new(b_{\mathrm{i}}) - new(a_{\mathrm{i}}) = (b_{\mathrm{i}} - a_{\mathrm{i}}) \cdot \frac{m_{\mathrm{j}}}{|A|} \geq \frac{m_{\mathrm{j}}}{n} \tag{4.178}$$

This completes the induction. When Algorithm 1 returns, it gives each agent i the interval $[x, y]$. Since $x \leq \mathrm{Cut}_{\mathrm{i}}(a_{\mathrm{i}})$ and $\mathrm{Cut}_{\mathrm{i}}(b_{\mathrm{i}}) \leq y$, this has value at least $b_{\mathrm{i}} - a_{\mathrm{i}} \geq \frac{1}{n}$ to agent i.

To argue the bound on the number of queries, we proceed by induction on k. For $k = 1$, the bound is n^2, which is satisfied since we issue $n - 1$ queries for each of n agents. In the inductive case, in the first round we issue $\lceil n^{1/k} \rceil - 1 \leq n^{1/k}$ queries for every agent, for a total of at most $n^{1+1/k}$. By the inductive assumption, the remaining number of queries is

$$\sum_{\mathrm{j}=1}^{\lceil n^{1/k} \rceil} (k-1) m_{\mathrm{j}}^{1+1/(k-1)} \leq (k-1) \Big(\sum_{\mathrm{j}=1}^{\lceil n^{1/k} \rceil} m_{\mathrm{j}} \Big)^{1+1/k} = (k-1) n^{1+1/k} \tag{4.179}$$

Combining, we get at most $kn^{1+1/k}$ queries in total. □

A key step in connecting cake cutting with sorting will be the following reduction, which reduces sorting a vector of n elements with rank queries to proportional (contiguous) cake cutting with n agents. Rank queries have the form *"How is $rank(x_{\mathrm{j}})$ compared to k?"*, where the answer can be "<", "=", or ">".

Proposition 4.6.2. *There exists a polynomial time reduction from sorting n elements with rank queries to proportional cake cutting with n agents. The reduction holds for any number of rounds.*

The reduction from sorting to cake cutting was essentially done in the work of Woeginger and Sgall [63], but appears implicitly. We formalize the connection to rank queries and note the reduction is round-preserving. The proof of Proposition 4.6.2 is in section 4.6.3.

Proposition 4.6.3. *There is a deterministic sorting algorithm in the rank query model that runs in k rounds and asks a total of $O(kn^{1+1/k})$ queries.*

Proof. By the reduction from sorting to cake cutting in Proposition 4.6.2, the upper bound follows from Proposition 4.6.1.

The sketch of the resulting deterministic sorting algorithm is as follows. In the first round, for each x in the array, query comparing $rank(x)$ to $\lceil n^{1-1/k} \rceil, \lceil 2n^{1-1/k} \rceil, \ldots, \lceil n - n^{1-1/k} \rceil$. This divides the array into $\lceil n^{1/k} \rceil$ blocks of indices of the form $(\lceil (\mathrm{i}-1)n^{1-1/k} \rceil, \lceil \mathrm{i}n^{1-1/k} \rceil)$ for $\mathrm{i} = 1, 2, \ldots, \lceil n^{1/k} \rceil$. Each element either has its exact rank revealed, or is found to belong to a particular block. Then recursively call the sorting algorithm in each block. □

4.6.2 Lower bound

In this section we first show a lower bound for sorting in the rank query model; for deterministic algorithms this bound improves upon the bound in [62] by a constant factor and the proof is simpler (see Section 4.6.4). Deterministic algorithms are relevant specifically for fair division, since some studies find that it is preferable to avoid randomness in the allocations if possible when dealing with human agents.

Proposition 4.6.4. *Let $c(k,n)$ be the minimum total number of queries required to sort n elements in the rank query model by the best deterministic algorithm in k rounds. Then $c(k,n) \geq \frac{k}{2\mathrm{e}} n^{1+1/k} - kn$.*

Alon and Azar [62] show a lower bound of $\Omega\left(kn^{1+1/k}\right)$ for randomized sorting with rank queries, which together with the reduction in Proposition 4.6.2 implies the next corollary.

Corollary 12. *Let $\mathcal{A}$ be an algorithm that runs in k rounds for solving proportional cake cutting with contiguous pieces for n agents. If $\mathcal{A}$ succeeds with constant probability, then it issues $\Omega(kn^{1+1/k})$ queries in expectation.*

The proof of Proposition 4.6.4 is given in section 4.6.4.

4.6.3 Sorting to cake cutting reduction

Here we prove the reduction of sorting to proportional cake cutting where the sorting is not with comparisons, but rather with queries that, given an item p and index i return whether the rank of p is less than, equal to, or greater than i. The bulk of the work has already been done by Woeginger and Sgall [63] through the introduction of a set of cake valuations and an adversary protocol. We present again their valuations and adversary protocol without proving the relevant lemmas; we would refer the reader to their paper for the proofs. Then we perform the last few steps to prove the reduction.

Definition 4.6.1. *[63] Let the α-point of an agent p be the infimum of all numbers x such that $\mu_p([0,x]) = \alpha$. In other words, $Cut_p(\alpha) = x$.*

We fix $0 < \epsilon < 1/n^4$. The choice is not important.

Definition 4.6.2. *[63] For $\mathrm{i} = 1, \ldots, n$ let $X_\mathrm{i} \subset [0,1]$ be the set consisting of the n points $\mathrm{i}/(n+1) + k\epsilon$ with integer $1 \leq k \leq n$. Further let $X = \bigcup_{1 \leq \mathrm{i} \leq n} X_\mathrm{i}$*

By definition every agent's 0-point is at 0. The positions of the i/n-points with $1 \leq \mathrm{i} \leq n$ are fixed by the adversary during the execution of the protocol. In particular, the i/n-points of all agents are distinct elements of X_{i}. Note that this implies that all i/n-points are left of all $(\mathrm{i}+1)/n$ points.

Definition 4.6.3. *[63] Let $\mathcal{I}_{p,\mathrm{i}}$ be a tiny interval of length ϵ centered around the* i/n*-point of agent p.*

We place all the value of each agent p in her $\mathcal{I}_{p,\mathrm{i}}$ for $\mathrm{i} = 0, \dots, n$. More precisely, for $\mathrm{i} = 0, \dots, n$ she has a sharp peak of value $\mathrm{i}/(n^2+n)$ immediately to the left of her i/n point and a sharp peak of value $(n-\mathrm{i})/(n^2+n)$ immediately to the right of her i/n point. Note that the measure between the i/n and $(\mathrm{i}+1)/n$ points is indeed $1/n$. Further note that the value $\mu_p(\mathcal{I}_{p,\mathrm{i}}) = 1/(n+1)$. Also note that the $\mathcal{I}_{p,\mathrm{i}}$ are all disjoint except for the $\mathcal{I}_{p,0}$, which are identical. Finally note that every α-point of an agent p lies inside one of that agent's $\mathcal{I}_{p,\mathrm{i}}$s.

Definition 4.6.4. *[63] If $x \in \mathcal{I}_{p,\mathrm{i}}$, then let $c_p(x)$ be the corresponding* i/n*-point of agent p.*

Definition 4.6.5. *[63] We call a protocol* primitive *iff in all of its cut operations $Cut_p(\alpha)$ the value of α is of the form* i/n *with integer $0 \le \mathrm{i} \le n$.*

Lemma 61. *[63] For every protocol P there exists a primitive protocol P' such that for every cake cutting instance of the restricted form described above,*

1. *P and P' make the same number of cuts.*
2. *if P assigns to agent p a piece $\mathcal{J}$ of measure $\mu_p(\mathcal{J}) \ge 1/n$, then also P' assigns to agent p a piece $\mathcal{J}'$ of measure $\mu_p(\mathcal{J}') \ge 1/n$.*

It is also true that given P, protocol P' can be quickly constructed. This follows directly from Woeginger and Sgall's constructive proof of the above lemma. This implies that we, the adversary, may assume w.l.o.g. that the protocol is primitive. We can now define the adversary's strategy. Fix a permutation $\boldsymbol{\pi}$ on $[n]$. Suppose at some point the protocol asks $\mathrm{Cut}_p(\mathrm{i}/n)$. With multiple queries in the same round, answer the queries in an arbitrary order.

1. If $\pi(p) < \mathrm{i}$, then the adversary assigns the i/n point of agent p to the smallest point in the set X_i that has not been used before.

2. If $\pi(p) > \mathrm{i}$, then the adversary assigns the i/n point of agent p to the largest point in the set X_i that has not been used before.

3. If $\pi(p) = \mathrm{i}$, then the adversary assigns the i/n point of agent p to the ith smallest point in the set X_i.

This strategy immediately precipitates the following lemma.

Lemma 62. *[63] If $\pi(p) \leq \mathrm{i} \leq \pi(q)$ and $p \neq q$, then the i/n point of agent p strictly precedes the i/n point of agent q*

At the end, the protocol must assign intervals to agents. Let $y_0, y_1, \ldots, y_n$ be the boundaries of these slices; i.e. $y_0 = 0$, $y_n = 1$, and all other y_j are cuts performed. Then there is a permutation ϕ of $[n]$ such that for $\mathrm{i} = 1, \ldots, n$ the interval $[y_{\mathrm{i}-1}, y_\mathrm{i})$ goes to agent $\phi(\mathrm{i})$.

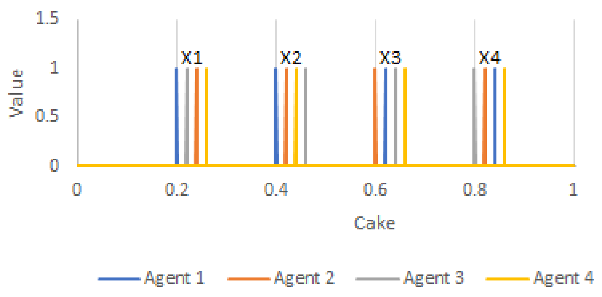


Figure 4.8. A potential value distribution for four agents. Each agent receives a spike in value in each of X_0, X_1, X_2, X_3, X_4 (X_0 is not shown). Each spike has total value 1/5, so to get the required 1/4 value an agent's slice must include parts of multiple X_i. Note that Agent 1 receives the first slot in X_1, Agent 2 receives the second slot in X_2, etc. Further note that slot 1 is allocated to Agent 1 in X_2 and slot 4 is allocated to Agent 4 in X_3. This implies that the slices must be allocated to agents $1, 2, 3, 4$ in order.

Lemma 63. *[63] If the primitive protocol P' is fair, then $y_\mathrm{i} \in X_\mathrm{i}$ for $1 \leq \mathrm{i} \leq n-1$ and the interval $[y_{\mathrm{i}-1}, y_\mathrm{i}]$ contains the $(\mathrm{i}-1)/n$-point and the i/n-point of agent $\phi(\mathrm{i})$.*

Lemma 64. *[63] For any permutation $\sigma \neq \mathrm{id}$ of $[n]$, there exists some i with*

$$\sigma(\mathrm{i}+1) \leq \mathrm{i} \leq \sigma(\mathrm{i})\,.$$

We can now claim that $\phi = \boldsymbol{\pi}^{-1}$. To prove this, suppose for sake of contradiction $\phi \neq \boldsymbol{\pi}^{-1}$; then $\boldsymbol{\pi} \circ \phi \neq id$ and by Lemma 64, there exists an i such that

$$\boldsymbol{\pi}(\phi(\mathrm{i}+1)) \leq \mathrm{i} \leq \boldsymbol{\pi}(\phi(\mathrm{i})) \tag{4.180}$$

Then let $p = \phi(\mathrm{i}+1)$ and $q = \phi(\mathrm{i})$. Further let z_p be the i/n point of agent p and z_q be the i/n point of agent q. By Lemma 64, we have $z_p < z_q$. By Lemma 63, we have $z_p \in [y_{\mathrm{i}}, y_{\mathrm{i}+1}]$ and $z_q \in [y_{\mathrm{i}-1}, y_{\mathrm{i}}]$. But this implies $z_p \geq y_{\mathrm{i}} \geq z_q$, in contradiction with $z_p < z_q$. Therefore $\phi = \boldsymbol{\pi}^{-1}$. With this preliminary work out of the way, we are finally ready to state and prove the reduction.

Proposition 4.6.2 (restated): *There exists a polynomial time reduction from sorting an n element with rank queries to proportional cake cutting with n agents. The reduction holds for any number of rounds.*

Proof. After an evaluation query $\mathrm{Eval}_p(x)$, where $x = \mathrm{Cut}_{p'}(\mathrm{i}/n)$ and $p \neq p'$, there are only two possible answers: $\mathrm{i}/(n+1)$ and $(\mathrm{i}+1)/(n+1)$. This reveals whether the i/n point of p is left or right of that of p'. This only reveals new information if $\boldsymbol{\pi}(p') = \mathrm{i}$. In this case, the information is whether $\boldsymbol{\pi}(p) < \mathrm{i}$ or $\boldsymbol{\pi}(p) > \mathrm{i}$.

After a cut query $\mathrm{Cut}_p(\mathrm{i}/n)$, there are only three answers. These correspond exactly to $\boldsymbol{\pi}(p) < \mathrm{i}$, $\boldsymbol{\pi}(p) = \mathrm{i}$, and $\boldsymbol{\pi}(p) > \mathrm{i}$. Thus w.l.o.g., all queries are cut queries.

Then given a sorting problem with rank queries, we can construct a proportional cake cutting instance such that any solution assigns slices according to the inverse permutation of the unsorted elements of the original sorting problem. The sorting problem can then be solved without any additional queries. Furthermore, each query in the cake cutting instance can be answered using at most one query in the sorting instance. This completes the reduction. Because of the one-to-one correspondence between queries, it immediately follows that the reduction holds for any number of rounds. □

4.6.4 Sorting lower bound

Our approach for the lower bound builds on the work in [75]. To show that this sorting problem is hard, we find a division between two regions of the array such that one must be solved in future rounds while the other still needs to be solved in the current round.

Proposition 4.6.4 (restated): *Let $c(k,n)$ be the minimum total number of queries required to sort n elements in k rounds in the rank query model. Then $c(k,n) \geq \frac{k}{2\mathrm{e}}n^{1+1/k} - kn$.*

Proof. We proceed by induction on k. For $k = 1$, note that if any two items p_{j}, p_k have no query for indices $\mathrm{i}, \mathrm{i}+1$ then the adversary can assign those positions to those items and the solver will be unable to determine their true order. Thus for $\mathrm{i} = 2, 4, \ldots n$ at least $n-1$ queries are necessary, for a total of $\lfloor n/2 \rfloor (n-1)$. Then

$$\lfloor n/2 \rfloor (n-1) \geq (n/2 - 1/2)(n-1) = n^2/2 - n + 1/2 > n^2/(2\mathrm{e}) - n\,.$$

For $k > 1$, assume the claim holds for all pairs (k', n') where either $(k' < k)$ or $(k' = k$ and $n' < n)$. If $n^{1/k} \leq 2\mathrm{e}$, then

$$\frac{n^{1/k}}{2\mathrm{e}} - 1 \leq 0 \iff \frac{k}{2\mathrm{e}}n^{1+1/k} - kn \leq 0$$

so the bound is non-positive, and is thus trivially satisfied. Thus we may assume $n^{1/k} > 2\mathrm{e}$.

If there are no queries in the first round, then we have

$$c(k,n) \geq c(k-1,n) \geq \frac{(k-1)}{2\mathrm{e}}n^{1+\frac{1}{k-1}} - (k-1)n = \frac{k}{2\mathrm{e}}n^{1+1/k}\left[\left(1-\frac{1}{k}\right)n^{\frac{1}{k^2-k}} + \frac{2\mathrm{e}}{kn^{1/k}}\right] - kn$$

From here it suffices to show $(1-\frac{1}{k})n^{1/(k^2-k)} + \frac{2\mathrm{e}}{kn^{1/k}} \geq 1$.

Recall the AMGM inequality $\alpha a + \beta b \geq a^\alpha b^\beta$ with $a, b, \alpha, \beta > 0$ and $\alpha + \beta = 1$. Taking $\alpha = 1 - 1/k$, $\beta = 1/k$, $a = n^{1/k^2 - 1/k}$, and $b = 2\mathrm{e}/n^{1/k}$, we get

$$\left(1 - \frac{1}{k}\right) n^{1/(k^2-k)} + \frac{2\mathrm{e}}{kn^{1/k}} \geq (2\mathrm{e})^{1/k} \geq 1 \tag{4.181}$$

so we may assume there is at least one query in the first round.

Take any k-round algorithm for sorting a set V of n elements using rank queries. Let x be the maximum integer such that there exist x items with no queries in $[1, x]$ but there do not exist $x + 1$ items with no queries in $[1, x + 1]$. Note that since there is at least one query, it follows that $x < n$. Let S be one such set of x items. Then at least $n - x$ items have a query in $[1, x + 1]$. At this point the adversary announces that every element of S precedes every element of $V - S$. The adversary also announces the item at position $x + 1$. We call this item p_{mid}. None of the $n - x$ queries help to sort the items in S since they are either at $x+1$ or for an item not in S, so we also need $c(k-1, x)$ queries to sort S. Additionally, none of the $n - x$ queries help to sort the items in $V - S - \{p_{mid}\}$, so we also need an additional $c(k, n - x - 1)$ queries to sort $V - S - \{p_{mid}\}$. This implies the following inequality.

$$c(k, n) \geq c(k, n - x - 1) + (n - x) + c(k - 1, x) \tag{4.182}$$

We consider two cases.

Case $x \geq k/\ln 2$**.** By the inductive assumption,

$$\begin{aligned} c(k, n) &\geq \frac{k}{2\mathrm{e}}(n - x - 1)^{1+1/k} - k(n - x - 1) + (n - x) + \frac{k-1}{2\mathrm{e}} x^{1+1/(k-1)} - (k - 1)x \\ &= \frac{k}{2\mathrm{e}} n^{1+1/k} \left[\left(1 - \frac{x+1}{n}\right)^{1+1/k} + \left(1 - \frac{1}{k}\right) \frac{x^{1+1/(k-1)}}{n^{1+1/k}} + \frac{2\mathrm{e}}{kn^{1/k}} \right] - kn + k \end{aligned} \tag{4.183}$$

In the AM-GM inequality $\alpha a + \beta b \geq a^\alpha b^\beta$, taking

$$\alpha = 1 - 1/k, \beta = 1/k, a = \frac{x^{1+1/(k-1)}}{n^{1+1/k}}, \text{ and } b = \frac{2\mathrm{e}}{n^{1/k}}, \tag{4.184}$$

we get

$$c(k,n) \geq \frac{k}{2\mathrm{e}} n^{1+1/k} \left[\left(1 - \frac{x+1}{n}\right)^{1+1/k} + \frac{x}{n^{1-1/k^2}} \cdot \frac{(2\mathrm{e})^{1/k}}{n^{1/k^2}} \right] - kn + k \tag{4.185}$$

Now, since $(1 + \frac{1}{k})^k \nearrow \mathrm{e}$, we have $\mathrm{e}^{1/k} > 1 + 1/k$. This yields

$$c(k,n) \geq \frac{k}{2\mathrm{e}} n^{1+1/k} \left[\left(1 - \frac{x+1}{n}\right)^{1+1/k} + 2^{1/k} \frac{x}{n} \left(1 + \frac{1}{k}\right) \right] - kn + k \tag{4.186}$$

Then recall Bernoulli's Inequality: $(1-a)^t \geq 1 - at$ if $t \geq 1$ and $a \leq 1$. This yields

$$\begin{aligned} c(k,n) &\geq \frac{k}{2\mathrm{e}} n^{1+1/k} \left[1 - \frac{x+1}{n}\left(1 + \frac{1}{k}\right) + 2^{1/k} \frac{x}{n} \left(1 + \frac{1}{k}\right) \right] - kn + k \\ &= \left[\frac{k}{2\mathrm{e}} n^{1+1/k} - kn \right] + \frac{k+1}{2\mathrm{e}} n^{1/k} \left(\left(2^{1/k} - 1\right) x - 1 \right) + k \end{aligned} \tag{4.187}$$

Then since by L'Hôpital's rule $k(2^{1/k} - 1) \to \ln 2$ from above, we have

$$c(k,n) \geq \left[\frac{k}{2\mathrm{e}} n^{1+1/k} - kn \right] + \frac{k+1}{2\mathrm{e}} n^{1/k} \left(\frac{x \ln 2}{k} - 1 \right) + k\,. \tag{4.188}$$

Then by invoking our case assumption that $x \geq k/\ln 2$, we get

$$c(k,n) \geq \left[\frac{k}{2\mathrm{e}} n^{1+1/k} - kn \right] + k \geq \frac{k}{2\mathrm{e}} n^{1+1/k} - kn,$$

as required.

Case $x < k/\ln 2$. From inequality (4.182), we get

$$c(k,n) \geq c(k, n-x-1) + (n-x) + c(k-1, x) \geq c(k, n-x-1) + n - \frac{k}{\ln 2}\,.$$

By the inductive hypothesis,

$$c(k,n) \geq \frac{k}{2\mathrm{e}}(n-x-1)^{1+1/k} - nk + n - \frac{k}{\ln 2} = \frac{k}{2\mathrm{e}}n^{1+1/k}\left(1 - \frac{x+1}{n}\right)^{1+1/k} - nk + n - \frac{k}{\ln 2}$$

Using the Bernoulli inequality $(1-a)^t \geq 1 - at$ with $t \geq 1$ and $a \leq 1$, we get

$$\begin{aligned} c(k,n) &\geq \frac{k}{2\mathrm{e}}n^{1+1/k}\Big[1 - \frac{x+1}{n}\left(1+\frac{1}{k}\right)\Big] - nk + n - \frac{k}{\ln 2} \\ &= \left[\frac{k}{2\mathrm{e}}n^{1+1/k} - nk\right] - \frac{k}{2\mathrm{e}}n^{1/k}(x+1)\left(1+\frac{1}{k}\right) + n - \frac{k}{\ln 2} \\ &\geq \left[\frac{k}{2\mathrm{e}}n^{1+1/k} - nk\right] - \frac{k+1}{2\mathrm{e}}n^{1/k}\left(\frac{k}{\ln 2}+1\right) + n - \frac{k}{\ln 2} \qquad (4.189) \end{aligned}$$

At this point we want to show $n > \frac{k+1}{2\mathrm{e}}n^{1/k}(1 + k/\ln 2) + k/\ln 2$. It suffices to show both of the following inequalities

$$\text{(i)} \ \frac{3n}{4} > \frac{k+1}{2\mathrm{e}}n^{1/k}\left(\frac{k}{\ln 2}+1\right) \quad \text{and} \quad \text{(ii)} \ \frac{n}{4} > \frac{k}{\ln 2} \qquad (4.190)$$

Inequality (i) holds if and only if

$$n^{1-1/k} > \frac{2(k+1)}{3\mathrm{e}}\left(\frac{k}{\ln 2}+1\right)$$

Since $n > (2\mathrm{e})^k$, we get that $n^{1-1/k} > (2\mathrm{e})^{k-1}$. For $k \geq 2$, we obtain $(2\mathrm{e})^{k-1} > \frac{2(k+1)}{3\mathrm{e}}\left(\frac{k}{\ln 2}+1\right)$, which concludes (i).

To show (ii), recall that $n > (2\mathrm{e})^k$. Then for $k \geq 2$ we get $(2\mathrm{e})^k > 4k/\ln(2)$, which implies (ii). This concludes the second case and the proof of the theorem.

□

4.7 Folklore lemmas

Here we include a few folklore lemmas that we use, together with their proofs for completeness.

Lemma 65. *Let* $\mathbf{y} = (y_1, \ldots, y_n)$ *with* $y_1 \geq \ldots \geq y_n \geq 0$ *and* $\sum_{\mathrm{i}=1}^{n} y_{\mathrm{i}} = 1$*. Then* $\sum_{\mathrm{j}=1}^{\mathrm{i}} y_{\mathrm{j}} \geq \mathrm{i}/n \; \forall \mathrm{i} \in [n]$.

Proof. Let $\mathrm{i} \in [n]$. Since $\mathbf{y}$ is decreasing, we have $\left(\sum_{\mathrm{j}=1}^{\mathrm{i}} y_{\mathrm{j}}\right)/\mathrm{i} \geq \left(\sum_{\mathrm{j}=\mathrm{i}+1}^{n} y_{\mathrm{j}}\right)/(n-\mathrm{i})$ (†).

Assume by contradiction that $y_1 + \ldots + y_{\mathrm{i}} < \frac{\mathrm{i}}{n}$ (‡). Adding $y_{\mathrm{i}+1} + \ldots + y_n$ to both sides of (‡), we get

$$\begin{aligned} 1 = y_1 + \ldots + y_n &< \frac{\mathrm{i}}{n} + y_{\mathrm{i}+1} + \ldots + y_n \\ &\leq \frac{\mathrm{i}}{n} + \frac{n-\mathrm{i}}{\mathrm{i}} \cdot \left(\sum_{\mathrm{j}=1}^{\mathrm{i}} y_{\mathrm{j}}\right) && \text{(By (†))} \\ &< \frac{\mathrm{i}}{n} + \frac{n-\mathrm{i}}{\mathrm{i}} \cdot \left(\frac{\mathrm{i}}{n}\right) && \text{(By (‡))} \\ &= 1\,. && (4.191) \end{aligned}$$

We obtained $1 < 1$, thus the assumption in (‡) must have been false and the lemma holds. □

Lemma 66. *Let* $x \in \mathbb{R}_{\geq 3}$*. Then* $x^{1+\frac{1}{x}} > x + 1\,.$

Proof. Raising both sides to the power $1/(x+1)$, the inequality is equivalent to $x^{\frac{1}{x}} > (x+1)^{\frac{1}{x+1}}$, or $(1/x)\ln(x) > (1/(x+1))\ln(x+1)$ (†).

Define $g(x) = (\ln x)/x$. Its derivative is $g'(x) = (1-\ln x)/x^2$. Thus g is increasing on $[1, \mathrm{e})$ and decreasing on $[\mathrm{e}, +\infty)$. It follows that (†) holds for $x \geq 3$ and the lemma follows. □

The next lemma shows that if v is an integrable function defined on $[0,1]$, then there is an interval I of length p on the circle where the interval $[0,1]$ is bent such that the point 0 coincides with 1, with the property that $\int_I v(x)\,dx = p$.

Lemma 67. *Let* $v : [0,1] \to \mathbb{R}_{\geq 0}$ *be an integrable function with* $\int_0^1 v(x)\,dx = 1$*. Then there exists* $a \in [0,1]$ *such that one of the following holds:*

- $\int_a^{a+p} v(x)\,dx = p$*, where* $0 \leq a \leq 1-p$*;*

- $\int_0^a v(x)\,dx + \int_{a+1-p}^1 v(x)\,dx = p$, *where* $1-p < a < 1$.

Proof. We define a new function $g : [0,1] \to \mathbb{R}_{\geq 0}$, such that

$$g(x) = \begin{cases} \int_x^{x+p} v(y)\,dy & \text{if } 0 \leq x \leq 1-p\,. \\ \\ \int_x^1 v(y)\,dy + \int_0^{x+p-1} v(y)\,dy & \text{if } 1-p < x \leq 1\,. \end{cases}$$

To prove the lemma it suffices to show that there exists $c \in [0,1]$ such that $g(c) = p$. Indeed, the function g is continuous and so integrable. Let $F : [0,1] \to \mathbb{R}_{\geq 0}$ be $F(x) = \int_0^x v(y)\,dy\,.$ Using this notation, we get:

$$\begin{aligned}
\int_0^1 g(x)\,dx &= \left[\int_0^{1-p}\int_x^{x+p} v(y)\,dy\,dx\right] + \left[\int_{1-p}^1 \left(\int_x^1 v(y)\,dy\right) + \left(\int_0^{x+p-1} v(y)\,dy\right)\,dx\right] \\
&= \int_0^{1-p}\Big(F(x+p) - F(x)\Big)\,dx + \int_{1-p}^1 \Big((F(1)-F(x)) + (F(x+p-1) - F(0))\Big)\,dx \\
&= \int_0^{1-p} F(x+p)\,dx - \int_0^{1-p} F(x)\,dx + \int_{1-p}^1 1\,dx - \int_{1-p}^1 F(x)\,dx + \int_{1-p}^1 F(x+p-1)\,dx \\
&\qquad\qquad (\text{Since } F(1) = 1 \text{ and } F(0) = 0.) \\
&= \int_0^{1-p} F(x+p)\,dx - \int_0^{1-p} F(x)\,dx + p - \int_{1-p}^1 F(x)\,dx + \int_{1-p}^1 F(x+p-1)\,dx\,.
\end{aligned} \tag{4.192}$$

We have

$$\int_{1-p}^1 F(x+p-1)\,dx = \int_0^p F(y)\,dy \qquad \text{and} \qquad \int_0^{1-p} F(x+p)\,dx = \int_p^1 F(z)\,dz\,. \tag{4.193}$$

Using (4.193) in (4.192) yields

$$\int_0^1 g(x)\,dx = \int_p^1 F(z)\,dz - \int_0^{1-p} F(x)\,dx + p - \int_{1-p}^1 F(x)\,dx + \int_0^p F(y)\,dy\,. \tag{4.194}$$

Notice that $\int_p^1 F(z)\,dz + \int_0^p F(y)\,dy = \int_0^1 F(x)\,dx$ and $-\int_0^{1-p} F(x)\,dx - \int_{1-p}^1 F(x)\,dx = -\int_0^1 F(x)\,dx$.

Therefore, the four integrals in (4.194) cancel each other and we get $\int_0^1 g(x)\,dx = p\,.$ (†)

Applying the first mean value theorem for definite integrals in (†), there exists $c \in [0,1]$ with the property that $g(c) = \frac{1}{1-0}\int_0^1 g(x)\,dx = p$, which concludes the proof. □

REFERENCES

[1] A. Ambainis, "Quantum lower bounds by quantum arguments," *Journal of Computer and System Sciences*, vol. 64, no. 4, pp. 750–767, 2002, ISSN: 0022-0000.

[2] S. Zhang, "On the power of ambainis lower bounds," *Theoretical Computer Science*, vol. 339, no. 2, pp. 241–256, 2005, ISSN: 0304-3975.

[3] S. Aaronson, "Lower bounds for local search by quantum arguments," *SIAM J. Comput.*, vol. 35, no. 4, pp. 804–824, 2006. DOI: 10.1137/S0097539704447237. [Online]. Available: https://doi.org/10.1137/S0097539704447237.

[4] A. C. Yao, "Probabilistic computations: Toward a unified measure of complexity," in *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE Computer Society, 1977, pp. 222–227.

[5] S. Brânzei, D. Choo, and N. Recker, "The sharp power law of local search on expanders," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, https://arxiv.org/abs/2305.08269, 2024.

[6] S. Brânzei and N. J. Recker, "Spectral lower bounds for local search," *arXiv preprint arXiv:2403.06248*, 2024.

[7] S. Brânzei, D. Paparas, and N. Recker, "Searching, sorting, and cake cutting in rounds," *arXiv preprint arXiv:2012.00738*, 2020.

[8] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, "How easy is local search?" *J. Comput. Syst. Sci.*, vol. 37, no. 1, pp. 79–100, 1988. DOI: 10.1016/0022-0000(88)90046-3. [Online]. Available: https://doi.org/10.1016/0022-0000(88)90046-3.

[9] D. Aldous, "Minimization algorithms and random walk on the d-cube," *The Annals of Probability*, vol. 11, no. 2, pp. 403–413, 1983.

[10] S. A. Vavasis, "Black-box complexity of local minimization," *SIAM Journal on Optimization*, vol. 3, no. 1, pp. 60–80, 1993.

[11] S. Bonnabel, "Stochastic gradient descent on riemannian manifolds," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013.

[12] S.-i. Amari, "Information geometry and its applications: Survey," in *Geometric Science of Information*, F. Nielsen and F. Barbaresco, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 3–3, ISBN: 978-3-642-40020-9.

[13] N. Boumal, *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023.

[14] J. Böttcher, K. P. Pruessmann, A. Taraz, and A. Würfl, "Bandwidth, expansion, treewidth, separators and universality for bounded-degree graphs," *European Journal of Combinatorics*, vol. 31, no. 5, pp. 1217–1227, 2010, ISSN: 0195-6698. DOI: https://doi.org/10.1016/j.ejc.2009.10.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0195669809002017.

[15] N. Alon and J. H. Spencer, *The Probabilistic Method*, Second. New York: Wiley, 2004, ISBN: 0471370460 9780471370468 0471722154 9780471722151 0471653985 9780471653981.

[16] H. Dinh and A. Russell, "Quantum and randomized lower bounds for local search on vertex-transitive graphs," *Quantum Information & Computation*, vol. 10, no. 7, pp. 636–652, 2010.

[17] M. Santha and M. Szegedy, "Quantum and classical query complexities of local search are polynomially related," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, 2004, pp. 494–501.

[18] D. C. Llewellyn, C. Tovey, and M. Trick, "Local optimization on graphs," *Discrete Applied Mathematics*, vol. 23, no. 2, pp. 157–178, 1989.

[19] A. Z. Broder, A. M. Frieze, and E. Upfal, "Static and dynamic path selection on expander graphs: A random walk approach," *Random Struct. Algorithms*, vol. 14, no. 1, pp. 87–109, 1999.

[20] S. Zhang, "Tight bounds for randomized and quantum local search," *SIAM Journal on Computing*, vol. 39, no. 3, pp. 948–977, 2009.

[21] C. Tovey, *Polynomial local improvement algorithms in combinatorial optimization*, Ph.D. thesis, Stanford University, 1981.

[22] X. Sun and A. C.-C. Yao, "On the quantum query complexity of local search in two and three dimensions," *Algorithmica*, vol. 55, no. 3, pp. 576–600, 2009.

[23] S. Brânzei and J. Li, "The query complexity of local search and brouwer in rounds," in *Conference on Learning Theory*, PMLR, 2022, pp. 5128–5145.

[24] Y. F. Verhoeven, "Enhanced algorithms for local search," *Information Processing Letters*, vol. 97, no. 5, pp. 171–176, 2006, ISSN: 0020-0190. DOI: https://doi.org/10.1016/j.ipl.2005.11.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020019005003091.

[25] Y. Babichenko, S. Dobzinski, and N. Nisan, "The communication complexity of local search," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 650–661.

[26] T. Leighton, S. Rao, and A. Srinivasan, "Multicommodity flow and circuit switching," in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, IEEE, vol. 7, 1998, pp. 459–465.

[27] J. Chuzhoy, "Routing in undirected graphs with constant congestion," *SIAM Journal on Computing*, vol. 45, no. 4, pp. 1490–1532, 2016.

[28] J. Chuzhoy, D. H. Kim, and R. Nimavat, "New hardness results for routing on disjoint paths," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 2017, pp. 86–99.

[29] J. Chuzhoy, D. H. Kim, and R. Nimavat, "Almost polynomial hardness of node-disjoint paths in grids," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018, pp. 1220–1233.

[30] J. Zhang, H. Lin, S. Jegelka, S. Sra, and A. Jadbabaie, "Complexity of finding stationary points of nonconvex nonsmooth functions," in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 11 173–11 182. [Online]. Available: https://proceedings.mlr.press/v119/zhang20p.html.

[31] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, "Lower bounds for finding stationary points I," *Math. Program.*, vol. 184, no. 1, pp. 71–120, 2020. DOI: 10.1007/s10107-019-01406-y. [Online]. Available: https://doi.org/10.1007/s10107-019-01406-y.

[32] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, "Lower bounds for finding stationary points II: first-order methods," *Math. Program.*, vol. 185, no. 1-2, pp. 315–355, 2021. DOI: 10.1007/s10107-019-01431-x. [Online]. Available: https://doi.org/10.1007/s10107-019-01431-x.

[33] S. Bubeck and D. Mikulincer, “How to trap a gradient flow,” in *Conference on Learning Theory*, PMLR, 2020, pp. 940–960.

[34] Y. Drori and O. Shamir, “The complexity of finding stationary points with stochastic gradient descent,” in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 2658–2667. [Online]. Available: https://proceedings.mlr.press/v119/drori20a.html.

[35] M. D. Hirsch, C. H. Papadimitriou, and S. A. Vavasis, “Exponential lower bounds for finding brouwer fix points,” *Journal of Complexity*, vol. 5, no. 4, pp. 379–416, 1989.

[36] C. H. Papadimitriou, “On the complexity of the parity argument and other inefficient proofs of existence,” *J. Comput. Syst. Sci.*, vol. 48, no. 3, pp. 498–532, 1994.

[37] X. Chen and X. Deng, “On the complexity of 2d discrete fixed point problem,” *Theor. Comput. Sci.*, vol. 410, no. 44, pp. 4448–4456, 2009. DOI: 10.1016/j.tcs.2009.07.052. [Online]. Available: https://doi.org/10.1016/j.tcs.2009.07.052.

[38] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, “The complexity of computing a nash equilibrium,” *Communications of the ACM*, vol. 52, no. 2, pp. 89–97, 2009.

[39] X. Chen, X. Deng, and S.-H. Teng, “Settling the complexity of computing two-player nash equilibria,” *Journal of the ACM (JACM)*, vol. 56, no. 3, pp. 1–57, 2009.

[40] V. V. Vazirani and M. Yannakakis, “Market equilibrium under separable, piecewise-linear, concave utilities,” *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–25, 2011.

[41] X. Chen, D. Paparas, and M. Yannakakis, “The complexity of non-monotone markets,” *Journal of the ACM (JACM)*, vol. 64, no. 3, pp. 1–56, 2017.

[42] C. Daskalakis, S. Skoulakis, and M. Zampetakis, “The complexity of constrained min-max optimization,” *CoRR*, vol. abs/2009.09623, 2020. arXiv: 2009.09623. [Online]. Available: https://arxiv.org/abs/2009.09623.

[43] X. Chen and X. Deng, “On algorithms for discrete and approximate brouwer fixed points,” in *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 2005, pp. 323–330.

[44] X. Chen and S.-H. Teng, "Paths beyond local search: A tight bound for randomized fixed-point computation," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, IEEE, 2007, pp. 124–134.

[45] J. Fearnley, P. Goldberg, A. Hollender, and R. Savani, "The complexity of gradient descent: $CLS = PPAD \cap PLS$," *Journal of the ACM*, vol. 70, no. 1, pp. 1–74, 2022.

[46] C. Daskalakis and C. Papadimitriou, "Continuous local search," in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*, SIAM, 2011, pp. 790–804.

[47] P. Hubáek and E. Yogev, "Hardness of continuous local search: Query complexity and cryptographic lower bounds," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2017, pp. 1352–1371.

[48] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *Journal of the ACM (JACM)*, vol. 46, no. 6, pp. 787–832, 1999.

[49] A. C.-C. Yao, "Probabilistic computations: Toward a unified measure of complexity," in *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, 1977, pp. 222–227. DOI: 10.1109/SFCS.1977.24.

[50] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[51] S. Olesker-Taylor and L. Zanetti, "Geometric bounds on the fastest mixing markov chain," *arXiv preprint arXiv:2111.05816*, 2021.

[52] D. A. Levin and Y. Peres, *Markov chains and mixing times*. American Mathematical Society; 2nd Revised edition, 2017.

[53] S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao, "Fastest mixing markov chain on graphs with symmetries," *SIAM J. Optimization*, vol. 20, pp. 792–819, 2 2009.

[54] S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao, "Symmetry analysis of reversible markov chains," *Internet Mathematics*, vol. 2, no. 1, pp. 31–71, 2005.

[55] R. palek and M. Szegedy, “All quantum adversary methods are equivalent,” in *Automata, Languages and Programming*, Springer Berlin Heidelberg, 2005, pp. 1299–1311, ISBN: 978-3-540-31691-6.

[56] S. Brânzei, D. Choo, and N. Recker, “The sharp power law of local search on expanders,” in *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Online available at: https://arxiv.org/abs/2305.08269, SIAM, 2024, pp. 1792–1809.

[57] J. Von Neumann, “Zur theorie der gesellschaftsspiele,” *Math. Ann.*, vol. 100, pp. 295–320, 1928.

[58] N. K. Vereshchagin, “Randomized boolean decision trees: Several remarks,” *Theoretical Computer Science*, vol. 207, no. 2, pp. 329–342, 1998.

[59] E. Kushilevitz and N. Nisan, *Communication Complexity.* Cambridge University Press, 1996.

[60] S. Ben-David, E. Blais, M. Göös, and G. Maystre, “Randomised composition and small-bias minimax,” in *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, IEEE, 2022, pp. 624–635. DOI: 10.1109/FOCS54457.2022.00065. [Online]. Available: https://doi.org/10.1109/FOCS54457.2022.00065.

[61] L. G. Valiant, “Parallelism in comparison problems,” *SIAM Journal on Computing*, vol. 4, no. 3, pp. 348–355, 1975.

[62] N. Alon and Y. Azar, “The average complexity of deterministic and randomized parallel comparison-sorting algorithms,” *SIAM J. Comput.*, vol. 17, no. 6, pp. 1178–1192, 1988.

[63] G. J. Woeginger and J. Sgall, “On the complexity of cake cutting,” *Discrete Optimization*, vol. 4, no. 2, pp. 213–220, 2007.

[64] A. Nemirovski, “On parallel complexity of nonsmooth convex optimization,” *Journal of Complexity*, vol. 10, no. 4, pp. 451–463, 1994.

[65] E. Balkanski and Y. Singer, "The adaptive complexity of maximizing a submodular function," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, Los Angeles, CA, USA, June 25-29, 2018*, I. Diakonikolas, D. Kempe, and M. Henzinger, Eds., ACM, 2018, pp. 1138–1151. DOI: 10.1145/3188745.3188752. [Online]. Available: https://doi.org/10.1145/3188745.3188752.

[66] S. Akl, *Parallel Sorting Algorithms*. Academic Press, 2014.

[67] N. Pippenger, "Sorting and selecting in rounds," *SIAM Journal on Computing*, vol. 16, no. 6, pp. 1032–1038, 1987.

[68] B. Bollobás, "Sorting in rounds," *Discrete Mathematics*, vol. 72, no. 1-3, pp. 21–28, 1988.

[69] N. Alon, Y. Azar, and U. Vishkin, "Tight complexity bounds for parallel comparison sorting," in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, IEEE, 1986, pp. 502–510.

[70] A. Wigderson and D. Zuckerman, "Expanders that beat the eigenvalue bound: Explicit construction and applications," *Combinatorica*, vol. 19, no. 1, pp. 125–138, 1999.

[71] W. I. Gasarch, E. Golub, and C. P. Kruskal, "Constant time parallel sorting: An empirical view," *J. Comput. Syst. Sci.*, vol. 67, no. 1, pp. 63–91, 2003. DOI: 10.1016/S0022-0000(03)00040-0. [Online]. Available: https://doi.org/10.1016/S0022-0000(03)00040-0.

[72] R. Häggkvist and P. Hell, "Parallel sorting with constant time for comparisons," *SIAM Journal on Computing*, vol. 10, no. 3, pp. 465–472, 1981.

[73] B. Bollobás and A. Thomason, "Parallel sorting," *Discrete Applied Mathematics*, vol. 6, no. 1, pp. 1–11, 1983.

[74] N. Alon and Y. Azar, "Sorting, approximate sorting, and searching in rounds," *SIAM Journal on Discrete Mathematics*, vol. 1, no. 3, pp. 269–280, 1988.

[75] N. Alon, Y. Azar, and U. Vishkin, "Tight complexity bounds for parallel comparison sorting," in *27th Annual Symposium on Foundations of Computer Science*, 1986, pp. 502–510.

[76] S. Brânzei and J. Li, “The query complexity of local search and brouwer in rounds,” in *Conference on Learning Theory, 2-5 July 2022, London, UK*, P. Loh and M. Raginsky, Eds., ser. Proceedings of Machine Learning Research, vol. 178, PMLR, 2022, pp. 5128–5145. [Online]. Available: https://proceedings.mlr.press/v178/branzei22a.html.

[77] R. Dorfman, “The detection of defective members of large populations,” *The Annals of Mathematical Statistics*, vol. 14, pp. 436–440, 1943.

[78] A. G. Dyachkov, V. V. Rykov, and A. M. Rashad, “Superimposed distance codes,” *Problems Control Information Theory*, vol. 18, pp. 237–250, 1989.

[79] N. Alon, D. Moshkovitz, and S. Safra, “Algorithmic construction of sets for k-restrictions,” *ACM Transactions on Algorithms*, vol. 2, no. 2, pp. 153–177, 2006.

[80] E. Porat and A. Rothschild, “Explicit non-adaptive combinatorial group testing schemes,” in *Proceedings of the 35th international colloquium on automata, languages and programming*, 2008, pp. 748–759.

[81] P. Indyk, H. Q. Ngo, and A. Rudra, “Efficiently decodable non-adaptive group testing,” in *Proceedings of the 21st annual ACM-SIAM symposium on discrete algorithms*, 2010, pp. 1126–1142.

[82] H. Ngo, E. Porat, and A. Rudra, “Efficiently decodable error-correcting list disjunct matrices and applications,” in *Proceedings of the 38th international colloquium on automata, languages and programming*, 2011, pp. 557–568.

[83] A. De Bonis, L. Gasieniec, and U. Vaccaro, “Optimal two-stage algorithms for group testing problems,” *SIAM Journal on Computing*, vol. 34, pp. 1253–1270, 2005.

[84] D. Eppstein, M. T. Goodrich, and D. S. Hirschberg, “Improved combinatorial group testing algorithms for real-world problem sizes,” *SIAM Journal on Computing*, vol. 36, pp. 1360–1375, 2007.

[85] D. Du and H. Park, “On competitive group testing,” *SIAM Journal on Computing*, vol. 23, no. 5, pp. 1019–1025, 1994.

[86] Y. Cheng, D.-Z. Du, and F. Zheng, “A new strongly competitive group testing algorithm with small sequentiality,” *Annals of Operations Research*, vol. 229, pp. 265–286, 2015.

[87] P. Damaschke, “Combinatorial search in two and more rounds,” *Theoretical Computer Science*, vol. 780, pp. 1–11, 2019.

[88] D. Gerbner and M. Vizer, “Rounds in a combinatorial search problem,” *Discrete Applied Mathematics*, vol. 276, pp. 60–68, 2020.

[89] J. Scarlett, M. Aldridge, and O. Johnson, *Group Testing: An Information Theory Perspective*. Now Foundations and Trends, 2019.

[90] H. Steinhaus, “The problem of fair division,” *Econometrica*, vol. 16, pp. 101–104, 1948.

[91] J. M. Robertson and W. A. Webb, *Cake Cutting Algorithms: Be Fair If You Can*. A. K. Peters, 1998.

[92] S. Brams and A. Taylor, *Fair Division: from cake cutting to dispute resolution*. Cambridge University Press, 1996.

[93] H. Moulin, *Fair Division and Collective Welfare*. The MIT Press, 2003.

[94] A. D. Procaccia, “Cake cutting: Not just child’s play,” *Communications of the ACM*, vol. 56, no. 7, pp. 78–87, 2013.

[95] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, Eds., *Handbook of Computational Social Choice*, 1st ed. Cambridge University Press, 2016.

[96] J. Goldman and A. D. Procaccia, “Spliddit: Unleashing fair division algorithms,” *ACM SIG. Exch*, vol. 13, no. 2, pp. 41–46, 2014.

[97] N. Alon, “Splitting necklaces,” *Advances in Mathematics*, vol. 63, no. 3, pp. 247–253, 1987.

[98] S. Branzei, “Computational fair division,” Ph.D. dissertation, Aarhus University, 2015.

[99] A. D. Procaccia, “Cake cutting algorithms,” in *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, Eds., Cambridge University Press, 2016, pp. 311–330. DOI: 10.1017/CBO9781107446984.014. [Online]. Available: https://doi.org/10.1017/CBO9781107446984.014.

[100] S. Even and A. Paz, "A note on cake cutting," *Discrete Applied Mathematics*, vol. 7, no. 3, pp. 285–296, 1984.

[101] J. Edmonds and K. Pruhs, "Cake cutting really is not a piece of cake," in *SODA*, 2006, pp. 271–278.

[102] A. D. Procaccia, "Thou shalt covet thy neighbors cake," in *IJCAI*, 2009, pp. 239–244.

[103] H. Aziz and S. Mackenzie, "A discrete and bounded envy-free cake cutting protocol for any number of agents," in *FOCS*, 2016, pp. 416–427.

[104] G. Amanatidis, G. Christodoulou, J. Fearnley, E. Markakis, C. Psomas, and E. Vakaliou, "An improved envy-free cake cutting protocol for four agents," in *Proceedings of the 11th International Symposium on Algorithmic Game Theory SAGT*, 2018, pp. 87–99.

[105] K. Cechlarova, J. Dobos, and E. Pillarova, "On the existence of equitable cake divisions," *J. Inf. Sci.*, vol. 228, pp. 239–245, 2013.

[106] A. D. Procaccia and J. Wang, "A lower bound for equitable cake cutting," in *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2017, pp. 479–495.

[107] S. Brânzei and N. Nisan, "The query complexity of cake cutting," in *NeurIPS*, 2022. [Online]. Available: http://papers.nips.cc/paper%5C_files/paper/2022/hash/f7a7bb369e48f10e85fce85b67d8c516-Abstract-Conference.html.

[108] S. Brânzei and N. Nisan, "Communication complexity of cake cutting," in *Proceedings of the 2019 ACM Conference on Economics and Computation (EC)*, 2019, p. 525.

[109] E. Segal-Halevi, "Cake-cutting with different entitlements: How many cuts are needed?" *CoRR*, vol. abs/1803.05470, 2018. arXiv: 1803.05470. [Online]. Available: http://arxiv.org/abs/1803.05470.

[110] A. Deligkas, A. Filos-Ratsikas, and A. Hollender, "Two's Company, Three's a Crowd: Consensus-Halving for a Constant Number of Agents," *arXiv e-prints*, arXiv:2007.15125, arXiv:2007.15125, Jul. 2020. arXiv: 2007.15125 [cs.GT].

[111] E. Balkanski, S. Brânzei, D. Kurokawa, and A. D. Procaccia, “Simultaneous cake cutting,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, C. E. Brodley and P. Stone, Eds., AAAI Press, 2014, pp. 566–572. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8330.

[112] A. Deligkas, A. Filos-Ratsikas, and A. Hollender, “Two’s company, three’s a crowd: Consensus-halving for a constant number of agents,” in *EC ’21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, P. Biró, S. Chawla, and F. Echenique, Eds., ACM, 2021, pp. 347–368. DOI: 10.1145/3465456.3467625. [Online]. Available: https://doi.org/10.1145/3465456.3467625.

[113] P. Goldberg, A. Hollender, and W. Suksompong, “Contiguous cake cutting: Hardness results and approximation algorithms,” *J. Artif. Intell. Res.*, vol. 69, pp. 109–141, 2020. DOI: 10.1613/jair.1.12222. [Online]. Available: https://doi.org/10.1613/jair.1.12222.

[114] G. Cheze, *Envy-free cake cutting: A polynomial number of queries with high probability*, 2020.

[115] P. W. Goldberg, A. Hollender, A. Igarashi, P. Manurangsi, and W. Suksompong, “Consensus halving for sets of items,” in *Web and Internet Economics - 16th International Conference, WINE 2020, Beijing, China, December 7-11, 2020, Proceedings*, X. Chen, N. Gravin, M. Hoefer, and R. Mehta, Eds., ser. Lecture Notes in Computer Science, vol. 12495, Springer, 2020, pp. 384–397. DOI: 10.1007/978-3-030-64946-3_27. [Online]. Available: https://doi.org/10.1007/978-3-030-64946-3%5C_27.

[116] A. Filos-Ratsikas, A. Hollender, K. Sotiraki, and M. Zampetakis, “Consensus-halving: Does it ever get easier?” In *Proceedings of the 21st ACM Conference on Economics and Computation*, ser. EC ’20, Virtual Event, Hungary: Association for Computing Machinery, 2020, pp. 381–399, ISBN: 9781450379755. DOI: 10.1145/3391403.3399527. [Online]. Available: https://doi.org/10.1145/3391403.3399527.

[117] A. Filos-Ratsikas, K. A. Hansen, K. Hogh, and A. Hollender, “Fixp-membership via convex optimization: Games, cakes, and markets,” in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, 2022, pp. 827–838. DOI: 10.1109/FOCS52979.2021.00085.

[118] N. Alon and A. Graur, *Efficient splitting of measures and necklaces*, 2020. DOI: 10.48550/ARXIV.2006.16613. [Online]. Available: https://arxiv.org/abs/2006.16613.

[119] B. Plaut and T. Roughgarden, "Communication complexity of discrete fair division," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, T. M. Chan, Ed., SIAM, 2019, pp. 2014–2033. DOI: 10.1137/1.9781611975482.122. [Online]. Available: https://doi.org/10.1137/1.9781611975482.122.

[120] V. Bilò *et al.*, "Almost envy-free allocations with connected bundles," in *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, A. Blum, Ed., ser. LIPIcs, vol. 124, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 14:1–14:21. DOI: 10.4230/LIPIcs.ITCS.2019.14. [Online]. Available: https://doi.org/10.4230/LIPIcs.ITCS.2019.14.

[121] A. Deligkas, E. Eiben, R. Ganian, T. Hamm, and S. Ordyniak, "The complexity of envy-free graph cutting," in *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 2022, pp. 237–243.

[122] H. Oh, A. D. Procaccia, and W. Suksompong, "Fairly allocating many goods with few queries," *SIAM Journal on Discrete Mathematics*, vol. 35, no. 2, pp. 788–813, 2021. DOI: 10.1137/20M1313349. eprint: https://doi.org/10.1137/20M1313349. [Online]. Available: https://doi.org/10.1137/20M1313349.

[123] P. Manurangsi and W. Suksompong, "Closing gaps in asymptotic fair division," *SIAM Journal on Discrete Mathematics*, vol. 35, no. 2, pp. 668–706, 2021. DOI: 10.1137/20M1353381. eprint: https://doi.org/10.1137/20M1353381. [Online]. Available: https://doi.org/10.1137/20M1353381.

[124] B. R. Chaudhury, T. Kavitha, K. Mehlhorn, and A. Sgouritsa, "A little charity guarantees almost envy-freeness," *SIAM J. Comput.*, vol. 50, no. 4, pp. 1336–1358, 2021. DOI: 10.1137/20M1359134. [Online]. Available: https://doi.org/10.1137/20M1359134.

[125] E. Elkind, E. Segal-Halevi, and W. Suksompong, "Mind the gap: Cake cutting with separation," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 5330–5338. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16672.

[126] S. Bouveret, K. Cechlárová, E. Elkind, A. Igarashi, and D. Peters, "Fair division of a graph," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ser. IJCAI'17, Melbourne, Australia: AAAI Press, 2017, pp. 135–141, ISBN: 9780999241103.

[127] X. Bei and W. Suksompong, "Dividing a graphical cake," in *AAAI Conference on Artificial Intelligence*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:207757688.

[128] A. Igarashi and F. Meunier, "Envy-free division of multi-layered cakes," in *Web and Internet Economics - 17th International Conference, WINE 2021, Potsdam, Germany, December 14-17, 2021, Proceedings*, M. Feldman, H. Fu, and I. Talgam-Cohen, Eds., ser. Lecture Notes in Computer Science, vol. 13112, Springer, 2021, pp. 504–521. DOI: 10.1007/978-3-030-94676-0_28. [Online]. Available: https://doi.org/10.1007/978-3-030-94676-0%5C_28.

[129] M. Kyropoulou, J. Ortega, and E. Segal-Halevi, "Fair cake-cutting in practice," *Games Econ. Behav.*, vol. 133, pp. 28–49, 2022. DOI: 10.1016/j.geb.2022.01.027. [Online]. Available: https://doi.org/10.1016/j.geb.2022.01.027.

[130] E. Segal-Halevi, "Fairly dividing a cake after some parts were burnt in the oven," in *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems*, ser. AAMAS '18, Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 1276–1284.

[131] A. Farhadi and M. Hajiaghayi, "On the complexity of chore division," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, International Joint Conferences on Artificial Intelligence Organization, Jul. 2018, pp. 226–232. DOI: 10.24963/ijcai.2018/31. [Online]. Available: https://doi.org/10.24963/ijcai.2018/31.

[132] P. W. Goldberg and I. Iaru, *Equivalence of models of cake-cutting protocols*, 2021. DOI: 10.48550/ARXIV.2108.03641. [Online]. Available: https://arxiv.org/abs/2108.03641.

[133] S. Brânzei, I. Caragiannis, D. Kurokawa, and A. D. Procaccia, "An algorithmic framework for strategic fair division," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, D. Schuurmans and M. P. Wellman, Eds., AAAI Press, 2016, pp. 418–424. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12294.

[134] E. Mossel and O. Tamuz, "Truthful fair division," in *Algorithmic Game Theory*, S. Kontogiannis, E. Koutsoupias, and P. G. Spirakis, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 288–299, ISBN: 978-3-642-16170-4.

[135] S. Branzei and P. B. Miltersen, "A dictatorship theorem for cake cutting," in *IJCAI*, 2015, pp. 482–488.

[136] Y. Chen, J. K. Lai, D. C. Parkes, and A. D. Procaccia, “Truth, justice, and cake cutting,” *Games and Economic Behavior*, vol. 77, no. 1, pp. 284–297, 2013. [Online]. Available: https://EconPapers.repec.org/RePEc:eee:gamebe:v:77:y:2013:i:1:p:284-297.

[137] X. Bu, J. Song, and B. Tao, “On existence of truthful fair cake cutting mechanisms,” *Artificial Intelligence*, vol. 319, p. 103 904, 2023, ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2023.103904. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370223000504.

[138] X. Bei, X. Lu, and W. Suksompong, “Truthful cake sharing,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 4809–4817.

[139] B. Tao, “On existence of truthful fair cake cutting mechanisms,” in *Proceedings of the 23rd ACM Conference on Economics and Computation*, ser. EC ’22, Boulder, CO, USA: Association for Computing Machinery, 2022, pp. 404–434, ISBN: 9781450391504. DOI: 10.1145/3490486.3538321. [Online]. Available: https://doi.org/10.1145/3490486.3538321.

[140] *Wolfram mathematica*. [Online]. Available: https://www.wolframalpha.com.

[141] *Useful inequalities*, https://sites.math.washington.edu/~morrow/335_17/ineq.pdf.

[142] Y. Manabe and T. Okamoto, “A cryptographic moving-knife cake-cutting protocol,” in *Proc. of 2nd IWIGP*, 2012, pp. 15–23.