OPT2023: 15th Annual Workshop on Optimization for Machine Learning

# The Sharp Power Law of Local Search on Expanders

**Simina Brânzei** SIMINA.BRANZEI@GMAIL.COM
*Purdue University*

**Davin Choo** DAVIN@U.NUS.EDU
*National University of Singapore*

**Nicholas Recker** NRECKER@PURDUE.EDU
*Purdue University*

## Abstract

Local search is a powerful heuristic in optimization and computer science, the complexity of which has been studied in the white box and black box models. In the black box model, we are given a graph $G = (V, E)$ and oracle access to a function $f : V \to \mathbb{R}$. The local search problem is to find a vertex $v$ that is a local minimum, i.e. with $f(v) \leq f(u)$ for all $(u, v) \in E$, using as few queries to the oracle as possible. The query complexity is well understood on the grid and the hypercube, but much less is known beyond.

We show that the query complexity of local search on $d$-regular expanders with constant degree is $\Omega\left(\frac{\sqrt{n}}{\log n}\right)$, where $n$ is the number of vertices of the graph. This matches within a logarithmic factor the upper bound of $O(\sqrt{n})$ for constant degree graphs from [2], implying that steepest descent with a warm start is essentially an optimal algorithm for expanders.

We obtain this result by considering a broader framework of graph features such as vertex congestion and separation number. We show that for each graph, the randomized query complexity of local search is $\Omega\left(\frac{n^{1.5}}{g}\right)$, where $g$ is the vertex congestion of the graph; and $\Omega\left(\sqrt[4]{\frac{s}{\Delta}}\right)$, where $s$ is the separation number and $\Delta$ is the maximum degree. For separation number the previous bound was $\Omega\left(\sqrt[8]{\frac{s}{\Delta}}/\log n\right)$, given by [17] for quantum and randomized algorithms. To prove these results, we design a variant of the relational adversary method from [1]. Our variant is asymptotically at least as strong as the version in [1] for all randomized algorithms, as well as strictly stronger on some problems and easier to apply in our setting.

**Keywords:** local search, local minimum, graph theory, query complexity, congestion, separation number, expansion, expanders, relational adversary

## Acknowledgments

This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing. This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/2021-08-013). This research is supported by the US National Science Foundation under grant CCF-2238372.

## 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., [14]) and the black box model (see, e.g., [2]). 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., [20]). 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 [6], which adapts stochastic gradient descent to work on Riemannian manifolds. See [4] and [7] 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.

## 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 $e \in E$ 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 $\max_{e \in E} \sum_{Q \in \mathcal{P}} c_e^Q$.

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 $G$ is [1]: $s = \max_{H \subseteq V} \min_{\substack{A \subseteq H: \\ |H|/4 \leq |A| \leq 3|H|/4}} |\delta(A)|$.

See, e.g., [10] for a survey of graph features.

**$d$-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 $\beta$-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. [3]). 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. Our contributions

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

### 3.1. Our variant of the relational adversary method

Our first contribution is to design a new variant of the relational adversary method of [1]. While [1] 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.

**Theorem 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*

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

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

$$\min_{\mathcal{Z} \subseteq \mathcal{X} : q(\mathcal{Z}) > 0} \frac{M(\mathcal{Z})}{100 \cdot q(\mathcal{Z})} \, . \tag{2}$$

1. For example, the separation number of a barbell graph (i.e., two cliques of size $n/2$ connected by a single edge) is $n/8$.
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.

The proof is included in the full version of the paper, where 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.

**Proposition 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 [1] gives a lower bound of $T \geq \Lambda$ for some $\Lambda > 0$, then Theorem 1 gives a lower bound of $T \geq \Lambda/40$.*

### 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.

**Theorem 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 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 ([2]). Theorem 2 gives meaningful results precisely when one can construct an all-pairs set of paths with vertex congestion $g = o(n^{1.5})$.

Theorem 2 also implies a lower bound of $\Omega\left(\frac{n^{1.5}}{g_e \cdot \Delta}\right)$ on any graph $G$, where $g_e$ is the edge congestion and $\Delta$ the maximum degree of $G$.

### 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] , which 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 is non-trivial however.

**Theorem 3** *Let $G = (V, E)$ be a connected undirected graph with $n$ vertices, maximum degree $\Delta$, 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 [16]. It is an interesting open question whether the current upper and lower bounds can be improved.

### 3.4. Corollary for expanders

Since $d$-regular $\beta$-expanders with constant $d$ and $\beta$ admit an all-pairs set of paths with congestion $O(n \cdot \log n)$ (e.g., see [9]), we get the next lower bound for constant degree expanders.

**Corollary 1** *Let $G = (V, E)$ be an undirected $d$-regular $\beta$-expander with $n$ vertices, where $d$ and $\beta$ 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 ([2]) 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

4. That is, the vertex $x^*$ is defined as: $x^* = x_j$, where $j = \text{argmin}_{i=1}^{t} f(x_i)$.

improvement can be made, returning the final vertex reached. When $t = \sqrt{n\Delta}$, where $\Delta$ 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.

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 $\beta$-expander with $n$ vertices and maximum degree $\Delta$. Then the randomized query complexity of local search on $G$ is $\Omega\left(\frac{\beta\sqrt{n}}{\Delta \log^2 n}\right)$.*

## 4. Related work

The query complexity of local search was first studied experimentally by [19]. The first breakthrough in the theoretical analysis of local search was obtained in the work of Aldous [2], which 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 $\{0,1\}^n$. This almost matches the query complexity of steepest descent with a warm start, which was also analyzed in [2] 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 $\Delta$.

Aldous' lower bound for the hypercube was later improved in [1], which showed a bound of $\Omega(2^{n/2}/n^2)$ via a variant of the relational adversary method from quantum computing. [21] 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, [16] developed a deterministic divide-and-conquer approach to solving local search that is theoretically optimal over all graphs in the deterministic context. On the hypercube, their method yields a lower bound of $\Omega(2^n/\sqrt{n})$ and an upper bound of $O(2^n \log(n)/\sqrt{n})$.

Another commonly studied graph for local search is the $d$-dimensional grid $[n]^d$. [1] used his relational adversary method there to show a randomized lower bound of $\Omega(n^{d/2-1}/\log n)$ for every constant $d \geq 3$. [21] 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 [18] 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 [8].

More general results are few and far between. On many graphs, the simple bound from [2] of $\Omega(\Delta)$ queries is the best known lower bound: hiding the local minimum in one of the $\Delta$ 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 [16]. [5] 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., [9, 11, 15]). 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., [12, 13]).

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.

Works like [6] 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., [4].

## References


[1] Scott Aaronson. Lower bounds for local search by quantum arguments. *SIAM J. Comput.*, 35 (4):804–824, 2006. doi: 10.1137/S0097539704447237. URL `https://doi.org/10.1137/S0097539704447237`.

[2] David Aldous. Minimization algorithms and random walk on the $d$-cube. *The Annals of Probability*, 11(2):403–413, 1983.

[3] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley, New York, second edition, 2004. ISBN 0471370460 9780471370468 0471722154 9780471722151 0471653985 9780471653981.

[4] Shun-ichi Amari. Information geometry and its applications: Survey. In Frank Nielsen and Frédéric Barbaresco, editors, *Geometric Science of Information*, pages 3–3, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40020-9.

[5] Yakov Babichenko, Shahar Dobzinski, and Noam Nisan. The communication complexity of local search. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 650–661, 2019.

[6] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[7] Nicolas Boumal. *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023.

[8] Simina Brânzei and Jiawei Li. The query complexity of local search and brouwer in rounds. In *Conference on Learning Theory*, pages 5128–5145. PMLR, 2022.

[9] Andrei Z. Broder, Alan M. Frieze, and Eli Upfal. Static and dynamic path selection on expander graphs: A random walk approach. *Random Struct. Algorithms*, 14(1):87–109, 1999.

[10] Julia Böttcher, Klaas P. Pruessmann, Anusch Taraz, and Andreas Würfl. Bandwidth, expansion, treewidth, separators and universality for bounded-degree graphs. *European Journal of Combinatorics*, 31(5):1217–1227, 2010. ISSN 0195-6698. doi: https://doi.org/10.1016/j.ejc.2009.10.010. URL `https://www.sciencedirect.com/science/article/pii/S0195669809002017`.

[11] Julia Chuzhoy. Routing in undirected graphs with constant congestion. *SIAM Journal on Computing*, 45(4):1490–1532, 2016.

[12] Julia Chuzhoy, David HK Kim, and Rachit Nimavat. New hardness results for routing on disjoint paths. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 86–99, 2017.

[13] Julia Chuzhoy, David HK Kim, and Rachit Nimavat. Almost polynomial hardness of node-disjoint paths in grids. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1220–1233, 2018.

[14] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988. doi: 10.1016/0022-0000(88)90046-3. URL https://doi.org/10.1016/0022-0000(88)90046-3.

[15] Tom Leighton, Satish Rao, and Aravind Srinivasan. Multicommodity flow and circuit switching. In *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, volume 7, pages 459–465. IEEE, 1998.

[16] Donna Crystal Llewellyn, Craig Tovey, and Michael Trick. Local optimization on graphs. *Discrete Applied Mathematics*, 23(2):157–178, 1989.

[17] Miklos Santha and Mario 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*, pages 494–501, 2004.

[18] Xiaoming Sun and Andrew Chi-Chih Yao. On the quantum query complexity of local search in two and three dimensions. *Algorithmica*, 55(3):576–600, 2009.

[19] Craig Tovey. Polynomial local improvement algorithms in combinatorial optimization, 1981. Ph.D. thesis, Stanford University.

[20] Stephen A Vavasis. Black-box complexity of local minimization. *SIAM Journal on Optimization*, 3 (1):60–80, 1993.

[21] Shengyu Zhang. Tight bounds for randomized and quantum local search. *SIAM Journal on Computing*, 39(3):948–977, 2009.