SIAG on Optimization Views and News

A Forum for the SIAM Activity Group on Optimization

Volume 33 Number 1 December 2025

Contents

Articles

Open Problems in Optimization

Jelena Diakonikolas
Department of Computer Sciences,
University of Wisconsin-Madison,
Madison, WI 53706, USA
jelena@cs.wisc.edu
https://www.jelena-diakonikolas.com/

The aim of this article is to initiate a discussion about core open problems in optimization and how they should be shared and discussed in our community.

1 Introduction

Optimization as a field was not driven by open problems. Instead, visionaries like Tyrrell Rockafellar, Boris Polyak, Arkadi Nemirovski and Yurii Nesterov, among many others, shaped the field by exploring questions that today are at the core of optimization theory and algorithms. Later generations (including the author's) look to their work with admiration and when seeking inspiration.

With the recent explosion of research in optimization, particularly driven by optimization's profound impact on modern machine learning, the field has expanded to the extent that it is hard to imagine engineering and computer science students going through their college education without learning about (stochastic) gradient descent.

As the field evolves in different directions, there are often rather simple-looking problems that appear well-understood to an untrained eye, yet many questions around them remain. This article is an effort to begin the conversation around what open problems are at the core of our field and are worth exploring, and to motivate other optimization researchers to contribute their own problems that they would like to see solved. The provided list of questions is necessarily biased by my own taste and experience. For those readers who disagree that the community should work on them, it is an invitation to provide their own open questions in talks, expository articles and research papers.

2 A Case for Recognized Open Problems

There is something to be said about the existence of recognized open problems and how they may impact both the development of our field and the recognition that work addressing them receives. For the former, there is a recent example that I believe illustrates well my point. Consider the very basic problem of zero-sum (simplex-simplex constrained) matrix games; namely, $\min_{\boldsymbol{x}\in\Delta_n}\max_{\boldsymbol{y}\in\Delta_m}\boldsymbol{y}^\top\boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ and $\Delta_i=\{\boldsymbol{z}\in\mathbb{R}^i:\boldsymbol{z}\geq\boldsymbol{0},\boldsymbol{1}^\top\boldsymbol{z}=1\}$, $i\in\{m,n\}$, is the probability simplex. Up until recently, if you had asked me[1] what is the oracle complexity, as measured by matrix-vector products, to find a solution to such a matrix game with primal-dual gap at most $\epsilon>0$, I would have said $1/\epsilon$, without a second thought.

The belief that the answer should be $1/\epsilon$ is not without a good reason; in fact, there is more than one good reason. By now classical papers on upper bounds for solving general min-max optimization and/or variational inequality problems [75, 79] include zero-sum matrix games as one of their main examples and have oracle complexity scaling with $1/\epsilon$. On the lower bound side, under the ℓ_2 geometry (note that zero-sum games correspond to the ℓ_1 geometry), known lower bounds scale with $1/\epsilon$. Further, in other related problem settings like smooth convex optimization, oracle complexity of problems in the standard ℓ_1 and ℓ_2 geometries are nearly[2] the same in their dependence on ϵ [45, 54], so it seems reasonable that the same phenomenon would transfer to min-max settings, including the aforementioned bilinear problems. Additionally, for zero-sum games in particular, there is a lower bound of the order $1/\epsilon$ that rules out a specific class of "strongly uncoupled" algorithms motivated by their use in the online learning community [25]. All this evidence would thus suggest that $1/\epsilon$ should be the correct answer.

As one could guess based on the prelude to this discussion, surprisingly, this is not true—the oracle complexity for zero-sum games, measured by the matrix-vector products, is lower than $1/\epsilon$! This was shown in a very recent (posted online in September 2025) paper [53]. More to the point, the authors would likely not have worked on this problem were the question of problem complexity not raised in a paper studying lower bounds for this problem, posted just nine months earlier [59]. This then raises the question of how many problems in optimization remain open not because they are too challenging to solve, but because people with the "right" ideas and techniques are not aware of them.

The second point here is about the recognition that work resolving open problems receives. Experts will, of course, appreciate seeing solutions to problems that they had known to be open for a while. Experts will appreciate seeing them even more so if they themselves had tried solving them. However, not knowing that a problem had been open for quite some time and not knowing that many people in our community had tried solving it can lead to insufficient appreciation from future readers and students, especially if the solution turns out to be "simple" (as many solutions are, in retrospect). Here are some examples.

Obtaining a fully parameter-free method for smooth strongly convex optimization while retaining optimal oracle complexity had been open for quite some time, and not for a lack of trying. How to relax the knowledge of the Lipschitz constant of the gradient is reasonably clear from standard analyses of accelerated gradient descent and its variants (e.g., [78, 41, 35, 23, 93, 1])—primarily because it is used in a specific inequality involving two subsequent algorithm iterates, making it possible to evaluate and adjust the Lipschitz constant estimate. However, how to remove the knowledge of the strong convexity modulus is much less clear. The reason is that the standard analysis applies inequalities derived from the definition of strong convexity involving the function minimizer—which is, of course, unknown—and so the inequality cannot be verified to adjust the modulus estimate. We still, in fact, do not have "direct" methods for solving smooth strongly convex optimization problems that are parameter-free and provably oracle complexity-optimal. There is some evidence that such "direct" methods may be impossible, at least for a specific class of algorithms that includes Nesterov's accelerated method [78] and its variants; see, for instance, [3, Section 4.1].

Instead, an alternative approach is to develop an "indirect" parameter-free oracle-complexity optimal method for minimizing the gradient norm of a smooth convex function and restart it each time the gradient norm is halved. It is a simple exercise (that I often give to students in my optimization classes) to show that this is sufficient. However, until recently, there was no oracle complexity-optimal method for minimizing the gradient norm of a smooth convex function. A natural approach using (a fixed) regularization is off by a logarithmic factor [80], and for a while it was open whether there existed an oracle complexity-optimal method in this context, parameter-free or not. The first oracle complexity-optimal method, due to [56], was not directly constructed by a human, but computer-assisted. Moreover, even today it is not clear whether this method could be made any-time (i.e., convergent without fixing the number of iterations or the target accuracy) or parameter-free.

Intriguingly, the regularization-based approach [80] can in fact lead to an oracle complexity-optimal, parameter-free algorithm for gradient norm minimization, but more care is needed in constructing this method and appropriately choosing and adjusting the amount of regularization to remove the extraneous logarithmic factor. Such a result was obtained only recently, for the Euclidean norm setting [60]. It seems plausible that the result should similarly be extendible to other ℓ_p norms, leveraging the results in [32], but this question is still open.

Given all the history, surely, a lot of credit should go to [60]. While experts understand this point (especially those of us who had thought about this problem), the fact that the problem did not exist as a "recognized open problem" obscures the reality that many researchers in our field made unsuccessful attempts at this problem, over many years.

On a related note, the existence of explicit, recognized open problems sidesteps the criticisms arising from solutions

[1] or almost any of my colleagues who had worked on oracle complexity of optimization; I have personally asked some of them!

[2] i.e., up to a logarithmic factor.

that turn out to be, often counter-intuitively, rather simple-looking. In my own work, some examples are a positive result for parallelizing a class of convex optimization problems (going against 30 years of research establishing impossibility of such results for broad classes of problems) [34] and efficient algorithms for solving certain classes of fixed-point equation problems with *expansive* operators [30].

3 Some Open Questions in Optimization

To make the discussion more concrete and "put my money where my mouth is," this section is devoted to the discussion of some open problems in optimization, necessarily biased by my own taste and interests, as previously disclosed.

3.1 Better understanding of computational complexity of optimization

The study of information-based complexity of optimization was initiated in the late 1970s and early 1980s [77, 92]. In the considered model, the underlying question is how much information about a problem needs to be inferred before the problem can be solved to a target error. Most optimization algorithms are oracle-based—that is, they can be fully described by a sequence of updates in which the information about the problem the algorithm attempts to solve is gained by querying an "oracle" that reveals information based on specific queries (e.g., value of the function, its gradient, or a higher-order derivative at a query point). Thus, the study of complexity has centered around "oracle complexity". In the oracle complexity model, one specifies a problem class, an oracle type, and a notion of error set to some target $\epsilon > 0$ ($\epsilon = 0$ would mean the problem is solved exactly). The oracle complexity of a problem is then defined as the minimum number of queries any algorithm must make on a worst-case instance in the class to output a solution with error at most ϵ. Observe here the min-max nature of the oracle complexity model: we take a minimum over all possible algorithms and we take a maximum over problem instances.

For standard classes of problems (e.g., convex minimization in standard ℓ_p-norm settings with weakly smooth objective functions) oracle complexity of optimization is considered well-understood, with matching upper and lower oracle complexity bounds; see, for instance [76, 77, 45]. While oracle complexity has been highly valuable in understanding the complexity of problems and algorithms, even within the classes of problems whose oracle complexity is well-understood, there remain questions that I believe are pertinent to their computational complexity and empirical performance, as discussed below.

Between the two oracle complexities in nonsmooth convex optimization. Consider the class of nonsmooth convex optimization problems

$$\min_{\boldsymbol{x}\in\mathcal{B}} f(\boldsymbol{x}), \tag{1}$$

where $\mathcal{B} := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq 1\}$ is the unit Euclidean ball centered at zero and f is a convex 1-Lipschitz function (w.r.t. the ℓ_2 norm). We discuss this problem in terms of its first-order oracle complexity (it suffices to think about the number of subgradient queries needed to solve this problem to error $\epsilon > 0$), though there are more general lower bounds that apply to any local[3] oracle.

Oracle complexity of the optimization problem in this context, where the notion of error is the standard optimality gap $\epsilon = f(\boldsymbol{x}) - \min_{\boldsymbol{y}\in\mathcal{B}} f(\boldsymbol{y})$, is well understood: up to absolute constant factors, the oracle complexity is $\min\{d\log(1/\epsilon),\, 1/\epsilon^2\}$ [77, 45, 54]. This means that there are both information theoretic lower bounds and algorithms that match this complexity, up to constant factors. However, the "low dimension/high accuracy" bound $d\log(1/\epsilon)$ and the "high dimension/low accuracy" bound $1/\epsilon^2$ are achieved by quite different algorithms.

The low-dimension bound $d\log(1/\epsilon)$ is attained by cutting planes or ellipsoid-style algorithms[4], whose implementations involve computationally expensive iterations; for existing methods [55, 81, 49, 62], the overall runtime is polynomial, but high, of order at least $d^3\log(1/\epsilon)$). On the other hand, the high-dimension $1/\epsilon^2$ bound is attained by a simple algorithm with cheap iterations: projected subgradient method (PsGD). The per-iteration complexity of PsGD for simple feasible sets like $\mathcal{B}$ amounts to the computational effort needed to compute (or approximate) the objective's subgradient.

Interestingly, we do not have a way to interpolate meaningfully between these two bounds using simple, subgradient-style algorithms.[5] What is known is that demonstrating complexity like $d\log(1/\epsilon)$ for PsGD is impossible. This is due to recently demonstrated memory-computation lower bounds [70, 11, 20], which establish (among other results) that any algorithm with oracle complexity $d\log(1/\epsilon)$ must require order-d^2 memory in the worst case; PsGD only requires order-d memory. What these lower bounds do not rule out is a simple algorithm of "PsGD style" with oracle complexity of the order, say, d/ϵ. It is also worth noting here that this line of work was, in fact, initiated by an open problem published at the Conference on Learning Theory in 2019 [97]. This leads us to ask the question:

Is such a result possible?

Obtaining such a fast algorithm of subgradient-type would be interesting for applications, for at least two reasons. First, subgradient-type methods are easy to implement and test. Second, for moderate-size problems and/or for a target error

[3] Here, a "local" oracle can be any oracle that, when queried at any point $\boldsymbol{x} \in \mathcal{B}$, returns only information about f pertaining to the point $\boldsymbol{x}$ (e.g., function value, subgradient, or any-order Taylor approximation, provided it exists).

[4] In fact, the $d\log(1/\epsilon)$ oracle complexity bound is attained up to constants by the center of mass method [77], which is not a polynomial-time algorithm. Polynomial-time methods for this problem, which are based on cutting planes or the inscribed ellipsoid, typically have slightly higher (e.g., by a log or a poly-log factor) oracle complexity; a point which we will ignore in the rest of this article.

[5] On the other hand, there are results that interpolate between these bounds using the more computationally intensive cutting plane-style methods [10].

$\epsilon > 0$ that is not too small compared to $1/d$, the computational complexity of such an algorithm could be lower than the complexity of either ellipsoid-style methods or the vanilla PsGD.

Parallelism as a bridge between memory and computation. Another related question here is how much can really be ruled out regarding algorithms with $d\,\log(1/\epsilon)$ oracle complexity. As mentioned earlier, existing memory-computation lower bounds rule out algorithms with memory that is lower than order-d^2. But could there exist algorithms that cleverly utilize this "high" memory, but are still reasonably "simple"? For instance, could there be a parallel algorithm, with order-d (or even poly(d)) many threads, each of which is "simple" (meaning, PsGD-style), that exhibits $d\,\log(1/\epsilon)$ *depth* (number of parallel rounds of computation)?

There is some evidence that this idea may not be entirely outlandish. In particular, such an approach is the premise of *survey descent* [46], which shows quite promising numerical performance, exhibiting empirically linear (i.e., geometrically fast) convergence. Nevertheless, rigorous oracle complexity has not been established[6] in the sense of this article, even when we specialize to classes of well-behaved functions like max-of-affine.

Relevant to this discussion are also lower bounds for parallelizing convex optimization. In the model introduced by Nemirovski [74], an algorithm can make poly$(d, 1/\epsilon)$ queries *in parallel* in each round. By in parallel, we mean that the queries asked in the same round can depend on previous queries and answers, but not on each other's answers. The oracle complexity in this context is then measured by the required *depth* of algorithms—the number of sequential rounds needed to construct a solution with target error $\epsilon > 0$. Information theoretic lower bounds for this problem rule out reducing oracle complexity (or depth) below $\min\{d^{1/3}, 1/\epsilon^2\}$ [74, 98, 33]. In other words, they do not rule out results like the aforementioned one concerning $d\log(1/\epsilon)$ depth within a method like survey descent. On the other hand, algorithms that take advantage of parallelism to reduce the number of sequential rounds all have polynomial dependence on $1/\epsilon$ (i.e., they do not converge linearly) [14, 38, 34]. Thus, none of these results lead to $d\,\log(1/\epsilon)$ depth for a subgradient-like algorithm. Of course, it would be even more interesting if the complexity could be reduced to some $d^\alpha\,\log(1/\epsilon)$, $\alpha \in [1/3,\,1)$ with poly$(d, 1/\epsilon)$ queries per round; say, to $d^{1/3}\,\log(1/\epsilon)$ (which would effectively match the lower bound). Obtaining such a result *with any method*—not just a subgradient-like method—or as conjectured in [14], ruling such a result out entirely, appears fundamental to our understanding of the complexity of this problem in the parallel model of oracle complexity.

3.2 When and why do cyclic algorithms work?

Cyclic block coordinate methods rely on a simple idea: partition the set of coordinates into blocks, and perform updates over one block at a time, traversing all sets in the partition over a period called a cycle. In the simplest form, the partition is into single-coordinate sets, and we refer to this as, simply, a cyclic coordinate method. There are different strategies for ordering the coordinates (blocks) in a cyclic coordinate method. One can employ (any) deterministic ordering (typically just $\{1, 2, 3, \dots, d\}$). Alternatively, one can employ a fixed random permutation generated at the initialization of the method, or an independent random permutation generated at the beginning of each cycle.

The most basic cyclic algorithms perform updates that fully minimize the objective function over the selected (block of) coordinate(s) in each cycle, keeping all other coordinates fixed. We will refer to such a strategy as the *exact cyclic (block) coordinate descent*. For some problems, the exact coordinate minimization updates are easy to perform, often in closed form, which is likely how those algorithms came to be. Some classical (and widely used) examples are the method of Kaczmarz for solving linear systems (which can be interpreted as exact cyclic coordinate descent on the dual problem) [52] and Osborne's method for a type of preconditioning procedure called matrix balancing [83].

In addition to being simple, cyclic coordinate descent (and its closely related variants) are highly effective in practice and preferred over alternative methods. For instance, cyclic coordinate methods are the default solvers in software packages for large-scale statistical learning like GLMNet [40] and SparseNet [73], while the method of Osborne [83] is the default preconditioning method for eigenvalue computation of non-symmetric matrices in all major software packages including Python, R, Julia, MATLAB, LAPACK, and EISPACK [82, 51, 72, 2, 88, 37, 71].

Despite their wide use, the theory of cyclic methods is still largely lacking. The first global complexity results for any nontrivial class of problems and cyclic methods were only obtained in the past decade, starting with [7]. These results—which are for cyclic gradient descent and its accelerated variant—are largely unsatisfying, as their worst-case complexity is worse than the worst case complexity of vanilla (single block) gradient descent and its acclerated version by poly(m) factors, where m is the number of blocks. Note that $m = d$ for cyclic coordinate descent! Moreover, such dependence appears unavoidable, due to lower bounds in [91], which apply even to simple convex quadratics. This dimensional dependence was shown to be fixable for at least some quadratics (but not more generally), using random permutation of the cyclic ordering [43, 61, 100]. This largely left open the question of how much worse cyclic methods can be than their corresponding "vanilla" baselines, in the worst case sense.

The work by my group [89, 18, 64, 17] brought a new per-

[6] It is relevant to mention here that [46] does provide *some* linear convergence results, but not in the strong global sense discussed here. Instead, the guarantees are *local* (asymptotic, with no quantitative bound on rate), they apply to a class of objectives expressible as the maximum of smooth strongly convex functions, and the linear rate scales with the minimum Lagrange multiplier associated with the problem, which may be quite close to zero. Thus, there is no formal guarantee that the depth of the algorithm would be even poly(d) $\log(1/\epsilon)$, even in the setting considered in [46].

spective on the convergence rate of cyclic methods, by replacing traditional (coordinate) Lipschitz conditions with a more "geometric" Lipschitz condition that captures how changes across different coordinates impact one another. This geometric Lipschitz condition (a full description is omitted here for brevity) loses no generality: it is implied by the more traditional Lipschitz assumptions, but it can give a more fine-grained picture of a considered problem instance. An additional insight from this work, beyond generalizing the Lipschitz conditions, is that for cyclic methods it appears important to look at the progress of algorithms across a *full cycle* of updates, in the aggregate, anchoring progress to either the beginning or the end of the cycle. This is in contrast to randomized block coordinate methods or traditional full-vector-update methods, which normally track progress across a single iteration/update.

The results from this line of work yielded the first cyclic method that provably addressed the broad class of variational inequalities with monotone Lipschitz operators [89]. This line of work further improved the worst-case scaling with the number of blocks m compared to baseline methods for convex optimization: specifically, the improvement was by a factor $\sqrt{m}$ for unaccelerated methods like gradient descent [89], and by a factor $m^{1/4}$ for accelerated methods [64]. This line of work further gave the first global complexity results for cyclic gradient descent applied to smooth nonconvex minimization problems (and its proximal operator-based generalization for composite problems) [18]. It also led to more fine-grained complexity characterizations for related incremental gradient methods [17].

Common to all these results is that the worst-case complexity scales polynomially with the dimension d (or, more generally, the number of coordinate blocks), and they depend crucially on *Euclidean* arguments. There are only two exceptions that I am aware of. The first one is [19], which gave a primal-dual cyclic block coordinate algorithm for extensive-form matrix games; the worst-case complexity is shown to be *no worse* than the complexity of the mirror-prox baseline (but not necessarily better). The second is the very recent result for the complexity of classical Osborne's algorithm [16]—a method that resisted any meaningful complexity analysis for decades—for which we proved it runs in near-linear time in the input size, which is effectively the best we could hope for. Osborne's algorithm can be interpreted as an exact cyclic coordinate descent on a certain log-sum-exp-type function.

What is interesting about these two examples is that both correspond to problems with natural extreme geometries: ℓ_1 and ℓ_∞. On the other hand, for the ℓ_2 geometry and our most basic problem—convex quadratics—we know there are "bad" worst-case examples for which cyclic algorithms (or, at least, cyclic gradient descent) do not work well. This raises the question:

Do cyclic methods have an edge in naturally extreme geometries like ℓ_1 and ℓ_∞?

Note that, if true, this would also explain, at least partially, the success of cyclic methods in statistical learning tasks solved by SparseNET and GLMNet, since the induced geometry in those tasks (through regularization like LASSO) is typically ℓ_1.

Another point here is that beyond the aforementioned result regarding Osborne's algorithm or the special case with two blocks (alternating minimization) [6, 36], we effectively do not have any theory for convergence of exact cyclic coordinate descent. Note that it is usually exact cyclic coordinate descent—not cyclic gradient descent or any of its variants—that is used in practice and is empirically effective. Exact cyclic coordinate descent was also the original motivation for the initial results in [7]. Hence another generally open question here is:

How does taking exact minimization steps in exact cyclic descent benefit convergence?

Here, it is particularly interesting to explain how/why *exact* minimization updates are generally more beneficial than coordinate *gradient* updates. Observe immediately here that if a problem is coordinate-separable, then one full cycle of the algorithm solves the problem to zero error. Thus, it appears that any reasonable analysis should depend on how the coordinates are coupled to each other and give a bound on the number of iterations equal to one in the coordinate-separable case (i.e., when there is no coupling).

3.3 Complexity of min-max optimization, root-finding, and fixed-point equations

Consider the following three families of problems:

1. (Unconstrained) Min-max optimization:
$$\min_{\boldsymbol{x}\in\mathcal{X}}\max_{\boldsymbol{y}\in\mathcal{Y}} f(\boldsymbol{x},\boldsymbol{y}), \tag{2}$$
where $\mathcal{X}\subset\mathbb{R}^n, \mathcal{Y}\subset\mathbb{R}^m$ are convex compact nonempty sets and f is smooth (gradient-Lipschitz).

2. Root finding: given a compact convex nonempty set $\mathcal{Z}\subset\mathbb{R}^d$ and oracle access to a Lipschitz continuous operator $\boldsymbol{F}:\mathcal{Z}\to\mathbb{R}^d$ such that $\boldsymbol{z}-\boldsymbol{F}(\boldsymbol{z})\in\mathcal{Z}$ for all $\boldsymbol{z}\in\mathcal{Z}$, find $\boldsymbol{z}^*\in\mathcal{Z}$ such that $\boldsymbol{F}(\boldsymbol{z}^*)=\boldsymbol{0}$; and

3. Fixed-point equation: given a compact convex nonempty set $\mathcal{Z}\subset\mathbb{R}^d$ and oracle access to a Lipschitz continuous operator $\boldsymbol{T}:\mathcal{Z}\to\mathcal{Z}$, find $\boldsymbol{z}^*\in\mathcal{Z}$ such that $\boldsymbol{T}(\boldsymbol{z}^*)=\boldsymbol{z}^*$.

The above problems can be stated (and have been extensively studied) in more general forms (e.g., unconstrained versions, non-Euclidean spaces), but for the present discussion the above formulations are convenient, particularly when mapping between the different problem classes. The rationale for the compactness assumption for the sets $\mathcal{X},\mathcal{Y},\mathcal{Z}$ is that, under the stated assumptions, it ensures existence of solutions (via Brouwer's fixed-point theorem [13], and due to relationships between these problems we will discuss momentarily), so we do not need to worry about these problems being potentially ill-posed.

Consider the following notion of ϵ-approximation for the min-max optimization problem in (2), where $\epsilon > 0$. Define first $\boldsymbol{G}_{\mathcal{X}}(\boldsymbol{x},\boldsymbol{y}) := \boldsymbol{x} - \Pi_{\mathcal{X}}(\boldsymbol{x} - \nabla_{\boldsymbol{x}} f(\boldsymbol{x},\boldsymbol{y}))$, $\boldsymbol{G}_{\mathcal{Y}}(\boldsymbol{x},\boldsymbol{y}) := \boldsymbol{y} - \Pi_{\mathcal{Y}}(\boldsymbol{y} + \nabla_{\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y}))$, where Π denotes the orthogonal projection operator and $\nabla_{\boldsymbol{x}}$, $\nabla_{\boldsymbol{y}}$ are the gradients w.r.t. the $\boldsymbol{x}$-argument and the $\boldsymbol{y}$-argument, respectively. We say that $(\boldsymbol{x}_\epsilon,\ \boldsymbol{y}_\epsilon) \in \mathcal{X} \times \mathcal{Y}$ is an ϵ-approximate solution to (2) if

$$r(\boldsymbol{x},\boldsymbol{y}) := \sqrt{\|\boldsymbol{G}_{\mathcal{X}}(\boldsymbol{x}_\epsilon,\boldsymbol{y}_\epsilon)\|_2^2 + \|\boldsymbol{G}_{\mathcal{Y}}(\boldsymbol{x}_\epsilon,\boldsymbol{y}_\epsilon)\|_2^2} \le \epsilon.$$

The quantity $r(\boldsymbol{x},\boldsymbol{y})$ is also sometimes referred to as the *natural residual* [39]. The condition $r(\boldsymbol{x},\boldsymbol{y}) \le \epsilon$ can be interpreted as (or, more precisely, it directly implies the standard notion of) ϵ-approximate *stationarity*, which for convex-concave problems further implies other standard notions of error like the primal-dual gap; see, e.g., [29] for a relevant discussion on these implications. Approximate versions of root-finding problems and fixed point equations are defined in a natural way for $\epsilon > 0$ as such points $\boldsymbol{z}_\epsilon \in \mathcal{Z}$ for which $\|\boldsymbol{F}(\boldsymbol{z}_\epsilon)\|_2 \le \epsilon$ and $\|\boldsymbol{T}(\boldsymbol{z}_\epsilon) - \boldsymbol{z}_\epsilon\|_2 \le \epsilon$, respectively.

Now let us discuss the relationship between the three stated problems. First, min-max optimization is captured by root-finding problems, because we can stack $\boldsymbol{x},\boldsymbol{y}$ into a $d = m + n$ dimensional vector $\boldsymbol{z}$ and define $\boldsymbol{F}(\boldsymbol{z})$ as the stacked vector comprised of $\boldsymbol{G}_{\mathcal{X}}(\boldsymbol{x},\boldsymbol{y})$ and $\boldsymbol{G}_{\mathcal{Y}}(\boldsymbol{x},\boldsymbol{y})$. Clearly, in this case, $\|\boldsymbol{F}(\boldsymbol{z})\|_2 = r(\boldsymbol{x},\boldsymbol{y})$ and so $\|\boldsymbol{F}(\boldsymbol{z})\|_2 \le \epsilon$ if and only if $r(\boldsymbol{x},\boldsymbol{y}) \le \epsilon$, for $\boldsymbol{z} = (\boldsymbol{x}^\top,\boldsymbol{y}^\top)^\top$. Second, for these problems as stated, root finding is equivalent to solving fixed-point equations. To see this, given either $\boldsymbol{F}$ or $\boldsymbol{T}$ for root finding or fixed-point equations respectively, consider $\boldsymbol{T}(\boldsymbol{z}) = \boldsymbol{z} - \boldsymbol{F}(\boldsymbol{z})$. Then, clearly, for any $\epsilon \ge 0$ and any $\boldsymbol{z} \in \mathcal{Z}$, $\|\boldsymbol{F}(\boldsymbol{x})\|_2 \le \epsilon$ if and only if $\|\boldsymbol{T}(\boldsymbol{z}) - \boldsymbol{z}\|_2 \le \epsilon$. Since $\boldsymbol{T}$ is Lipschitz-continuous and self-maps a convex compact set, Brouwer's fixed-point theorem [13] ensures the existence of a fixed point ($\boldsymbol{z}^* \in \mathcal{Z}$ such that $\boldsymbol{T}(\boldsymbol{z}^*) = \boldsymbol{z}^*$), which in turn, by equivalence, ensures the existence of a root of $\boldsymbol{F}$, and which by specialization ensures the existence of a solution to (2).

When additional assumptions on the problems are imposed (e.g., if we assume the objective in (2) is convex-concave, or if we assume the operator $\boldsymbol{F}$ is monotone, or if we assume $\boldsymbol{T}$ is 1-Lipschitz—a.k.a. nonexpansive), these problems are known to be *tractable*, that is solvable in time polynomial in both $1/\epsilon$ and the parameters describing the problem, and their complexity is fairly well understood. On the other hand, without further assumptions, these problems become (or, at least, are conjectured to become) computationally intractable.

In particular, consider our previously introduced class of fixed-point equations, and let D be the diameter of $\mathcal{Z}$ and let γ be the Lipschitz constant of $\boldsymbol{T}$. There is a recently demonstrated lower bound result establishing, for sufficiently large but constant $\gamma > 1$, a worst-case oracle complexity exponential in dimension for solving the associated fixed-point equation problem to error $\epsilon > 0$ [4][7]. This lower bound applies even under the *smoothed analysis framework* [90], meaning that even if the instance space is slightly perturbed and we look at the suitable "average case" under this perturbation, the problem remains exponentially hard.

On the other hand, my recent results [30] showed that as long as $\gamma < 1+\epsilon/D$ (and not trivially close to this bound), the oracle complexity of finding a $\boldsymbol{z} \in \mathcal{Z}$ with $\|\boldsymbol{T}(\boldsymbol{z}) - \boldsymbol{z}\|_2 \le \epsilon$ is polynomial. Further, this result extends to other norms and even some geodesic metric spaces. An interesting question that arises here is

Is there a phase transition in oracle complexity of solving fixed-point equations to error $\epsilon > 0$ occurring at $\gamma = 1 + \epsilon/D$?

The same paper [30] introduced a class of α-gradually expansive operators, defined as operators that satisfy, for all $\boldsymbol{z},\boldsymbol{w} \in \mathcal{Z}$:

$$\begin{aligned}&\|\boldsymbol{T}(\boldsymbol{z}) - \boldsymbol{T}(\boldsymbol{w})\| \\ &\le \Big(1 + \frac{\alpha \min\{\|\boldsymbol{T}(\boldsymbol{z}) - \boldsymbol{z}\|, \|\boldsymbol{T}(\boldsymbol{w}) - \boldsymbol{w}\|\}}{D}\Big)\|\boldsymbol{z} - \boldsymbol{w}\|.\end{aligned} \tag{3}$$

Such operators are $(1+\alpha)$-Lipschitz continuous and their Lipschitz constant can reach $\gamma = 1 + \alpha$. Interestingly, as shown in [30], the associated (ϵ-approximate) fixed-point equation problems remain polynomial time solvable for $\alpha < \sqrt{2} - 1$, bringing the Lipschitz constant of $\boldsymbol{T}$ up to anything smaller than (but not trivially close to) $\sqrt{2}$. Given how quickly and how computationally hard the problem becomes for $\gamma > 1$, it seems reasonable to ask

For what other classes of fixed-point equations with $\gamma > 1$ is polynomial oracle complexity attainable?

Answers to such questions have immediate implications for root-finding problems, at least as defined above, due to the discussed equivalence. The root-finding problems were defined above in relation to fixed-point equations to ensure existence of solutions. But this is not necessary, and it may be possible to guarantee existence (so the problem is well-posed) even without imposing compactness or boundedness assumptions on $\mathcal{Z}$. This whole area appears interesting in its own right due to further connections to equilibrium problems like variational inequalities. An example of a condition that ensures tractability in this case is the weak Minty condition [31], but *it is not known whether further (strict) generalizations* (i.e., not just extending the value of the associated parameter, but defining a strictly larger problem class) of such a condition while retaining tractability *are possible.* This is yet another possible direction for future research.

Finally, the complexity of min-max optimization (2) with general, possibly nonconvex-nonconcave smooth objectives, is not yet well understood. One would guess that the hardness stemming from the exponential oracle complexity of fixed-point equations would transfer to this setting, but we do not yet have such lower bounds that are specific to standard min-max optimization problems as stated in (2). This

[7] There is a much older lower bound for the ℓ_∞ case of this problem due to [47] that similarly established worst-case exponential oracle complexity that kicks in already at $\gamma \ge 1 + c\epsilon/D$ for a universal constant $c > 0$.

is because the class of fixed-point equation problems is larger than the class of min-max optimization problems, so not every fixed-point equation can be mapped back to a problem as stated in (2). Instead, existing exponential oracle complexity lower bounds for min-max optimization like [8, 26] only apply to problems in which the feasible set for $(\boldsymbol{x}, \boldsymbol{y})$ is *not* a product space—instead, the constraints for the primal variables $\boldsymbol{x}$ and the dual variables $\boldsymbol{y}$ are *coupled.* Thus, one of the basic questions to address here and that seems within reach is

Is oracle complexity of (2) *exponential in dimension?*[8]

Finally, given the centrality of (2) to problems in game theory and their more recent relevance to machine learning (particularly within the realm of adversarial training) [24], the general exploration of structural properties that could make such problems tractable would be of interest. It is worthwhile to mention here some existing examples, such as finding second-order stationary points for unconstrained min-max problems with bounded objectives [69], finding first-order stationary points for one-side tractable problems (e.g., where the primal/dual is convex/concave or satisfies dominance conditions like e.g., Polyak-Łojasiewicz condition) or under "interaction-dominant" conditions—see the recent work [22] (and references therein) for an overview—and approximating the primal-dual gap for distributionally robust optimization problems arising from generalized linear models [63].

3.4 Local error bounds in learning problems

The study of local error bounds is central to optimization theory. Broadly speaking, local error bounds are conditions pertaining to optimization problems that establish boundedness of distance to target solutions (e.g., optima) as a function of some problem residual that can typically be tracked or bounded in the analysis of optimization algorithms (e.g., optimality gap, gradient norm or natural residual). For instance, if r is a residual function (a nonnegative-valued function that is equal to zero if and only if $\boldsymbol{x}$ is a target solution) and $\mathcal{S}$ denotes the set of target solutions, a typical local error bound would look like

$$\mu \operatorname{dist}(\boldsymbol{x}, \mathcal{Z})^{\nu} \leq r(\boldsymbol{x}), \tag{4}$$

where $\mu > 0, \nu > 0$ are the local error bound parameters (e.g., $\nu = 2$ for locally "quadratic growth").

The upshot is that such conditions can ensure that once a suitable problem residual is driven to zero by an optimization algorithm, the iterates themselves converge to the set of optima. They also play a central role in establishing linear convergence of algorithms; at least, this is true for specific local error bounds like "sharpness" and a related Polyak-Łojasiewicz condition–see, e.g., [87] and references therein.

The study of local error bounds dates back to the work of Hoffman on solving systems of linear equations from the 1950s [48] and featured prominently in optimization theory research over the subsequent decades [5, 66, 65, 12, 68, 67, 85, 15, 86, 84]. What is common to these results is a focus on proving, for a large class of problems (e.g., quadratic problems, linear programs, complementarity problems), that for some fixed ν (e.g., $\nu = 1$ or $\nu = 2$), μ is *strictly positive.*

While technically this ensures convergence of the iterates to optima, it does not tell us much about how fast this happens, particularly if μ can be arbitrarily close to zero. In fact, for most of the problems for which general local error bounds have been established, μ can be exponentially small in the problem dimension. For this reason, to my knowledge, many optimization researchers studying global complexity properties of optimization algorithms did not consider local error bounds to be particularly useful in the study of complexity.

In the more recent literature, different types of local error bounds started resurfacing in the study of machine learning problems. For instance, a type of local error bound that applies to problems in matrix completion, matrix sensing, and related problems, is the main reason such problems are solvable despite their inherent nonconvexity [50, 9, 99].

In my own work (with students and collaborators), proving that local error bounds hold for different stochastic, nonconvex optimization problems arising in learning theory (specifically, in learning generalized linear models and single-index models) has proven crucial to establishing their solvability under certain distributional assumptions [94, 96, 102, 95, 101]. On the other hand, such problems are known to be computationally intractable if no distributional assumptions are imposed [28, 42, 27]. Interestingly, we showed how to solve many such problems by solving them on the Euclidean sphere [95, 101, 96, 102]—which, itself, is a nonconvex set. Thus, a general path forward in addressing optimization problems arising in machine learning may be addressed by asking and answering:

If a problem appears solvable (in a formal learning sense, and at least based on empirical evidence), does it satisfy a local error bound?

It is my hope that the optimization community will once again pick up on this general research area, as optimization theorists are uniquely qualified to lead the charge on establishing such local error bound results.

4 Where to go from here?

This article collected a few open questions that I have been thinking about and genuinely care about. More broadly, I believe our community would benefit from being more intentional—and more open—about sharing the open problems that can move the field forward. Beyond formulating and discussing open questions within research papers, here are a few concrete suggestions for how we might proceed:

1. I would like to see more senior and widely recognized members of the optimization community write expository pieces articulating what they view as the central open questions in the field—particularly questions that have remained unresolved despite years of sustained effort.

[8] This is a known open problem, also stated in [8].

2. Major optimization meetings that draw broad participation, such as SIAM-OP, ISMP, and ICCOPT, could include dedicated sessions on open problems. These could take the form of panels (with speakers explicitly prompted to discuss key open directions) or research-style talks focused on presenting and contextualizing compelling open questions.

3. It would also be valuable to create dedicated venues for short expository articles centered on open problems. For example, in theoretical computer science, Conference on Learning Theory (COLT) has an open-problems track (which, incidentally, has also featured optimization-related questions; see, e.g., [21, 57, 44, 58, 97]). Perhaps flagship optimization journals such as SIOPT and Mathematical Programming could consider introducing similar tracks.

Acknowledgments. The author acknowledges partial support from the Air Force Office of Scientific Research under award number FA9550-24-1-0076, the NSF CAREER Award CCF-2440563, and the NSF MFAI Award DMS-2502282. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Defense.

References

[1] Zeyuan Allen-Zhu and Lorenzo Orecchia. “Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent”. In: *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2017, pp. 3–1.

[2] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, et al. *LAPACK users' guide*. SIAM, 1999.

[3] Yossi Arjevani and Ohad Shamir. “On the iteration complexity of oblivious first-order optimization algorithms”. In: *International Conference on Machine Learning*. PMLR. 2016, pp. 908–916.

[4] Idan Attias, Yuval Dagan, Constantinos Daskalakis, Rui Yao, and Manolis Zampetakis. “Fixed Point Computation: Beating Brute Force with Smoothed Analysis”. In: *arXiv preprint arXiv:2501.10884* (2025).

[5] AA Auslender and J-P Crouzeix. “Global regularity theorems”. In: *Mathematics of Operations Research* 13.2 (1988), pp. 243–253.

[6] Amir Beck. “On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes”. In: *SIAM Journal on Optimization* 25.1 (2015), pp. 185–209.

[7] Amir Beck and Luba Tetruashvili. “On the convergence of block coordinate descent type methods”. In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2037–2060.

[8] Martino Bernasconi, Matteo Castiglioni, Andrea Celli, and Gabriele Farina. “On the role of constraints in the complexity of min-max optimization”. In: *arXiv preprint arXiv:2411.03248* (2024).

[9] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. “Global optimality of local search for low rank matrix recovery”. In: *Advances in Neural Information Processing Systems* 29 (2016).

[10] Moïse Blanchard, Junhui Zhang, and Patrick Jaillet. “Memory-constrained algorithms for convex optimization”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 6156–6189.

[11] Moïse Blanchard, Junhui Zhang, and Patrick Jaillet. “Quadratic memory is necessary for optimal query complexity in convex optimization: Center-of-mass is Pareto-optimal”. In: *The Thirty Sixth Annual Conference on Learning Theory*. PMLR. 2023, pp. 4696–4736.

[12] Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. “The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems”. In: *SIAM Journal on Optimization* 17.4 (2007), pp. 1205–1223.

[13] Luitzen Egbertus Jan Brouwer. “Über abbildung von mannigfaltigkeiten”. In: *Mathematische annalen* 71.1 (1911), pp. 97–115.

[14] Sébastien Bubeck, Qijia Jiang, Yin-Tat Lee, Yuanzhi Li, and Aaron Sidford. “Complexity of highly parallel non-smooth convex optimization”. In: *Advances in neural information processing systems* 32 (2019).

[15] James V Burke and Michael C Ferris. “Weak sharp minima in mathematical programming”. In: *SIAM Journal on Control and Optimization* 31.5 (1993), pp. 1340–1359.

[16] Xufeng Cai, Jason Altschuler, and Jelena Diakonikolas. “Near-Linear Runtime for a Classical Matrix Preconditioning Algorithm”. In: *Journal of the ACM* (2025).

[17] Xufeng Cai, Cheuk Yin Lin, and Jelena Diakonikolas. “Tighter convergence bounds for shuffled SGD via primal-dual perspective”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 72475–72524.

[18] Xufeng Cai, Chaobing Song, Stephen Wright, and Jelena Diakonikolas. “Cyclic block coordinate descent with variance reduction for composite nonconvex optimization”. In: *International Conference on Machine Learning*. 2023, pp. 3469–3494.

[19] Darshan Chakrabarti, Jelena Diakonikolas, and Christian Kroer. “Block-coordinate methods and restarting for solving extensive-form games”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 10692–10720.

[20] Xi Chen and Binghui Peng. “Memory-query tradeoffs for randomized convex optimization”. In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2023, pp. 1400–1413.

[21] Xinyi Chen and Elad Hazan. “Open problem: Black-box reductions & adaptive gradient methods”. In: *Proceedings of Machine Learning Research vol* 196 (2024), pp. 1–8.

[22] Xiuyuan Cheng, Yao Xie, Linglingzhi Zhu, and Yunqin Zhu. “Worst-case generation via minimax optimization in Wasserstein space”. In: *arXiv preprint arXiv:2512.08176* (2025).

[23] Michael Cohen, Jelena Diakonikolas, and Lorenzo Orecchia. “On acceleration with noise-corrupted gradients”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1019–1028.

[24] Constantinos Daskalakis. “Non-concave games: A challenge for game theory’s next 100 years”. In: *Cowles Preprints* (2022).

[25] Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. “Near-optimal no-regret algorithms for zero-sum games”. In: *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2011, pp. 235–254.

[26] Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. “The complexity of constrained min-max optimization”. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 1466–1478.

[27] Ilias Diakonikolas, Daniel Kane, and Lisheng Ren. “Near-optimal cryptographic hardness of agnostically learning halfspaces and ReLU regression under Gaussian marginals”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 7922–7938.

[28] Ilias Diakonikolas, Daniel Kane, and Nikos Zarifis. “Near-optimal SQ lower bounds for agnostically learning halfspaces and ReLUs under Gaussian marginals”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 13586–13596.

[29] Jelena Diakonikolas. “Halpern Iteration for Near-Optimal and Parameter-Free Monotone Inclusion and Strong Solutions to Variational Inequalities”. In: *Conference on Learning Theory*. PMLR. 2020, pp. 1428–1451.

[30] Jelena Diakonikolas. “Pushing the Complexity Boundaries of Fixed-Point Equations: Adaptation to Contraction and Controlled Expansion”. In: *arXiv preprint arXiv:2506.17698* (2025).

[31] Jelena Diakonikolas, Constantinos Daskalakis, and Michael I Jordan. “Efficient methods for structured nonconvex-nonconcave min-max optimization”. In: *International Conference on Artificial Intelligence and Statistics*. 2021, pp. 2746–2754.

[32] Jelena Diakonikolas and Cristóbal Guzmán. “Complementary composite minimization, small gradients in general norms, and applications”. In: *Mathematical Programming* 208.1 (2024), pp. 319–363.

[33] Jelena Diakonikolas and Cristóbal Guzmán. “Lower bounds for parallel and randomized convex optimization”. In: *Journal of Machine Learning Research* 21.5 (2020), pp. 1–31.

[34] Jelena Diakonikolas and Cristóbal Guzmán. “Optimization on a finer scale: Bounded local subgradient variation perspective”. In: *arXiv preprint arXiv:2403.16317* (2024). To appear in SIAM Journal on Optimization.

[35] Jelena Diakonikolas and Lorenzo Orecchia. “Accelerated Extra-Gradient Descent: A Novel Accelerated First-Order Method”. In: *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2018, pp. 23–1.

[36] Jelena Diakonikolas and Lorenzo Orecchia. “Alternating randomized block coordinate descent”. In: *International Conference on Machine Learning*. 2018.

[37] R Documentation. *Balance a square matrix via LAPACK’s dgebal*. https://www.rdocumentation.org/packages/expm/versions/0.99-1.1/topics/balance.

[38] John C Duchi, Peter L Bartlett, and Martin J Wainwright. “Randomized smoothing for stochastic optimization”. In: *SIAM Journal on Optimization* 22.2 (2012), pp. 674–701.

[39] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2003.

[40] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. “Regularization paths for generalized linear models via coordinate descent”. In: *Journal of statistical software* 33.1 (2010), p. 1.

[41] Alexander Vladimirovich Gasnikov and Yurii E Nesterov. “Universal method for stochastic composite optimization problems”. In: *Computational Mathematics and Mathematical Physics* 58.1 (2018), pp. 48–64.

[42] Surbhi Goel, Aravind Gollakota, and Adam Klivans. “Statistical-query lower bounds via functional gradients”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 2147–2158.

[43] Mert Gürbüzbalaban, Asuman Ozdaglar, Pablo A Parrilo, and N Denizcan Vanli. “When cyclic coordinate descent outperforms randomized coordinate descent”. In: *Advances in Neural Information Processing Systems*. 2017.

[44] Cristóbal Guzmán. “Open problem: The oracle complexity of smooth convex optimization in nonstandard settings”. In: *Conference on Learning Theory*. PMLR. 2015, pp. 1761–1763.

[45] Cristóbal Guzmán and Arkadi Nemirovski. “On lower complexity bounds for large-scale smooth convex optimization”. In: *Journal of Complexity* 31.1 (2015), pp. 1–14.

[46] XY Han and Adrian S Lewis. “Survey descent: A multipoint generalization of gradient descent for nonsmooth optimization”. In: *SIAM Journal on Optimization* 33.1 (2023), pp. 36–62.

[47] Michael D Hirsch, Christos H Papadimitriou, and Stephen A Vavasis. “Exponential lower bounds for finding Brouwer fixed points”. In: *Journal of Complexity* 5.4 (1989), pp. 379–416.

[48] Alan J Hoffman. “On Approximate Solutions of Systems of Linear Inequalities”. In: *Journal of Research of the National Bureau of Standards* 49.4 (1952).

[49] Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. “An improved cutting plane method for convex optimization, convex-concave games, and its applications”. In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 2020, pp. 944–953.

[50] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. “How to escape saddle points efficiently”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1724–1732.

[51] Julia. *LinearAlgebra.eigen: Compute eigenvalues and eigenvectors*. https://docs.julialang.org/en/v1/stdlib/LinearAlgebra/#LinearAlgebra.eigen.

[52] Stefan Karczmarz. “Angenaherte auflosung von systemen linearer glei-chungen”. In: *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.* (1937), pp. 355–357.

[53] Ishani Karmarkar, Liam O’Carroll, and Aaron Sidford. “Solving Zero-Sum Games with Fewer Matrix-Vector Products”. In: *The 66th Annual Symposium on Foundations of Computer Science (FOCS 2025)*. 2025.

[54] Leonid Khachiyan, Arkadi Nemirovski, and Yurii Nesterov. “Optimal methods for the solution of large-scale convex programming problems”. In: *Modern Mathematical Methods in Optimization, Academie Verlag, Berlin* (1993).

[55] Leonid G Khachiyan, Sergei Pavlovich Tarasov, and II Erlikh. “The method of inscribed ellipsoids”. In: *Soviet Mathematics-Doklady*. Vol. 37. 1. 1988, pp. 226–230.

[56] Donghwan Kim and Jeffrey A Fessler. “Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions”. In: *Journal of optimization theory and applications* 188.1 (2021), pp. 192–219.

[57] Tomer Koren and Shahar Segal. “Open problem: Tight convergence of sgd in constant dimension”. In: *Conference on Learning Theory*. PMLR. 2020, pp. 3847–3851.

[58] Guy Kornowski and Ohad Shamir. “Open problem: Anytime convergence rate of gradient descent”. In: *The Thirty Seventh Annual Conference on Learning Theory*. PMLR. 2024, pp. 5335–5339.

[59] Guy Kornowski and Ohad Shamir. “The oracle complexity of simplex-based matrix games: Linear separability and Nash equilibria”. In: *The 38th Annual Conference on Learning Theory (COLT 2025)*. 2025.

[60] Guanghui Lan, Yuyuan Ouyang, and Zhe Zhang. “Optimal and parameter-free gradient minimization methods for convex and nonconvex optimization”. In: *arXiv preprint arXiv:2310.12139* (2023).

[61] Ching-Pei Lee and Stephen J Wright. “Random permutations fix a worst case for cyclic coordinate descent”. In: *IMA Journal of Numerical Analysis* 39.3 (2019), pp. 1246–1275.

[62] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. “A faster cutting plane method and its implications for combinatorial and convex optimization”. In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE. 2015, pp. 1049–1065.

[63] Shuyao Li, Sushrut Karmalkar, Ilias Diakonikolas, and Jelena Diakonikolas. “Learning a single neuron robustly to distributional shifts and adversarial label noise”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 67383–67421.

[64] Cheuk Yin Lin, Chaobing Song, and Jelena Diakonikolas. “Accelerated cyclic coordinate dual averaging with extrapolation for composite convex optimization”. In: *International Conference on Machine Learning*. 2023, pp. 21101–21126.

[65] Stanisław Łojasiewicz. “Sur la géométrie semi-et sous-analytique”. In: *Annales de l’institut Fourier*. Vol. 43. 5. 1993, pp. 1575–1595.

[66] Stanisław Łojasiewicz. “Une propriété topologique des sous-ensembles analytiques réels”. In: *Les équations aux dérivées partielles* 117.87-89 (1963).

[67] Olvi L Mangasarian. “A condition number for differentiable convex inequalities”. In: *Mathematics of Operations Research* 10.2 (1985), pp. 175–179.

[68] Olvi L Mangasarian. “Error bounds for nondegenerate monotone linear complementarity problems”. In: *Mathematical Programming* 48.1 (1990), pp. 437–445.

[69] Oren Mangoubi and Nisheeth K Vishnoi. “Greedy adversarial equilibrium: an efficient alternative to nonconvex-nonconcave min-max optimization”. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 896–909.

[70] Annie Marsden, Vatsal Sharan, Aaron Sidford, and Gregory Valiant. “Efficient convex optimization requires superlinear memory”. In: *Conference on Learning Theory*. PMLR. 2022, pp. 2390–2430.

[71] MathWorks. *Balance – diagonal scaling to improve eigenvalue accuracy*. https://www.mathworks.com/help/matlab/ref/balance.html.

[72] MathWorks. *Eig: eigenvalues and eigenvectors*. https://www.mathworks.com/help/matlab/ref/eig.html.

[73] Rahul Mazumder, Jerome H Friedman, and Trevor Hastie. “Sparsenet: Coordinate descent with nonconvex penalties”. In: *Journal of the American Statistical Association* 106.495 (2011), pp. 1125–1138.

[74] Arkadi Nemirovski. “On parallel complexity of nonsmooth convex optimization”. In: *Journal of Complexity* 10.4 (1994), pp. 451–463.

[75] Arkadi Nemirovski. “Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems”. In: *SIAM Journal on Optimization* 15.1 (2004), pp. 229–251.

[76] Arkadi S Nemirovski and Yurii Nesterov. “Optimal methods of smooth convex minimization”. In: *USSR Computational Mathematics and Mathematical Physics* 25.2 (1985), pp. 21–30.

[77] Arkadi S. Nemirovski and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. 1983.

[78] Yurii Nesterov. “A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$”. In: *Doklady AN USSR*. Vol. 269. 1983, pp. 543–547.

[79] Yurii Nesterov. “Dual extrapolation and its applications to solving variational inequalities and related problems”. In: *Mathematical Programming* 109.2 (2007), pp. 319–344.

[80] Yurii Nesterov. “How to make the gradients small”. In: *Optima. Mathematical Optimization Society Newsletter* 88 (2012), pp. 10–11.

[81] Yurii E. Nesterov. “Self-Concordant Functions and Polynomial-Time Methods in Convex Programming”. In: *Report, Central Economic and Mathematic Institute, USSR Acad. Sci* (1989). URL: https://cir.nii.ac.jp/crid/1571980074518897024.

[82] NumPy. *numpy.linalg.eig: Compute the eigenvalues and right eigenvectors of a square array*. https://numpy.org/doc/stable/reference/generated/numpy.linalg.eig.html.

[83] EE Osborne. “On pre-conditioning of matrices”. In: *Journal of the ACM (JACM)* 7.4 (1960), pp. 338–345.

[84] Jong-Shi Pang. “Error bounds in mathematical programming”. In: *Mathematical Programming* 79.1 (1997), pp. 299–332.

[85] Boris T Polyak. “Sharp minima, Institute of Control Sciences Lecture Notes, Moscow, USSR (1979)”. In: *IIASA Workshop on Generalized Lagrangians and Their Applications, IIASA, Laxenburg, Austria*. 1979.

[86] Stephen M Robinson. “An application of error bounds for convex programming in a linear space”. In: *SIAM Journal on Control* 13.2 (1975), pp. 271–273.

[87] Vincent Roulet and Alexandre d’Aspremont. “Sharpness, restart and acceleration”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 1119–1129.

[88] Brian T Smith, James M. Boyle, BS Garbow, Y Ikebe, VC Klema, and CB Moler. *Matrix eigensystem routines-EISPACK guide*. Vol. 6. Springer, 2013.

[89] Chaobing Song and Jelena Diakonikolas. “Cyclic coordinate dual averaging with extrapolation”. In: *SIAM Journal on Optimization* 33.4 (2023), pp. 2935–2961.

[90] Daniel A Spielman and Shang-Hua Teng. “Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time”. In: *Journal of the ACM (JACM)* 51.3 (2004), pp. 385–463.

[91] Ruoyu Sun and Yinyu Ye. “Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version”. In: *Mathematical Programming* (2019), pp. 1–34.

[92] Joseph F. Traub, Grzegorz W. Wasilkowski, and Henryk Woźniakowski. *Information-based complexity*. Academic Press Professional, Inc., 1988. ISBN: 978-0-12-697545-1.

[93] Paul Tseng. “On accelerated proximal gradient methods for convex-concave optimization”. In: *submitted to SIAM Journal on Optimization* 2.3 (2008).

[94] Puqian Wang, Nikos Zarifis, Ilias Diakonikolas, and Jelena Diakonikolas. “Robustly Learning a Single Neuron via Sharpness”. In: *40th International Conference on Machine Learning*. 2023.

[95] Puqian Wang, Nikos Zarifis, Ilias Diakonikolas, and Jelena Diakonikolas. “Robustly Learning Monotone Single-Index Models”. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025.

[96] Puqian Wang, Nikos Zarifis, Ilias Diakonikolas, and Jelena Diakonikolas. “Sample and Computationally Efficient Robust Learning of Gaussian Single-Index Models”. In: *The Thirty-Eighth Annual Conference on Neural Information Processing Systems* (2024).

[97] Blake Woodworth and Nathan Srebro. “Open problem: The oracle complexity of convex optimization with limited memory”. In: *Conference on Learning Theory*. PMLR. 2019, pp. 3202–3210.

[98] Blake E Woodworth, Jialei Wang, Adam Smith, Brendan McMahan, and Nati Srebro. “Graph oracle models, lower bounds, and gaps for parallel stochastic optimization”. In: *Advances in neural information processing systems* 31 (2018).

[99] John Wright and Yi Ma. *High-dimensional data analysis with low-dimensional models: Principles, computation, and applications*. Cambridge University Press, 2022.

[100] Stephen Wright and Ching-pei Lee. “Analyzing random permutations for cyclic coordinate descent”. In: *Mathematics of Computation* 89.325 (2020), pp. 2217–2248.

[101] Nikos Zarifis, Puqian Wang, Ilias Diakonikolas, and Jelena Diakonikolas. “Robustly Learning Monotone Generalized Linear Models via Data Augmentation”. In: *Conference on Learning Theory*. 2025.

[102] Nikos Zarifis, Puqian Wang, Ilias Diakonikolas, and Jelena Diakonikolas. “Robustly Learning Single-Index Models via Alignment Sharpness”. In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. 2024, pp. 58197–58243.

Bulletin

Email items to siagoptnews@lists.mcs.anl.gov *for consideration in the bulletin of forthcoming issues.*

Event Announcements

SIAM Conference on Optimization
2-5 June 2026
Edinburgh, United Kingdom

This is the conference of the SIAM Activity Group on Optimization. The SIAM Conference on Optimization showcases the latest research in the theory, algorithms, software, and applications of optimization.

It brings together mathematicians, operations researchers, computer scientists, engineers, software developers, and practitioners, fostering an ideal environment for exchanging new ideas and addressing significant challenges.

The conference serves as a platform for specialists and users of optimization in academia, government, and industry to collaborate and share insights.

Important dates:

13 November 2025: Submit Now: Minisymposium Proposal Submission Deadline Extended

13 November 2025: Submit Now: Contributed Lecture, and Minisymposium Presentation Abstract Submission Deadline

2 March 2026: Apply Now: Travel Support Application Deadline

5 May 2026: Early Registration Deadline

URL: https://www.siam.org/conferences-events/siam-conferences/op26

IFORS 2026
12-17 July 2026
Vienna, Austria

The 24th Conference of the International Federation of Operational Research Societies will take place in Vienna, Austria from 12 to 17 July 2026.

The IFORS 2026 will take place in the main building of the University of Vienna, Universitätsring 1, 1010 Vienna

Important dates

1 December 2025: Opening of abstract submissions

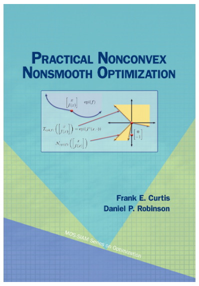
15 March 2026: Deadline for abstract submissions

25 April 2026: Early bird registration deadline

25 April 2026: Final registration deadline for authors

URL: https://www.ifors2026.at/home

Books



Practical Nonconvex Nonsmooth Optimization

by Frank E. Curtis and Daniel P. Robinson

Publisher: SIAM

ISBN: 978-1-61197-858-2

Published: 2025

Series: MOS-SIAM Series on Optimization

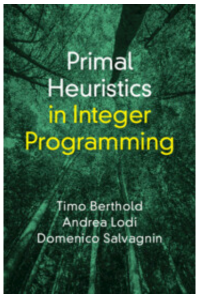
About the book: This book provides a clear and accessible introduction to an important class of problems in mathematical optimization: those involving continuous functions that may be nonconvex, nonsmooth, or both. The authors begin with an intuitive treatment of theoretical foundations, including properties of nonconvex and nonsmooth functions and conditions for optimality. They then offer a broad overview of the most effective and efficient algorithms for solving such problems, with a focus on practical applications in areas such as control systems, signal processing, and data science.

The book focuses on problems defined over finite-dimensional real-vector spaces, requiring no extensive background in functional analysis.

It begins with nonconvex smooth optimization rather than convex optimization, making the material more approachable for readers without extensive prior knowledge of convex analysis and optimization.

The book employs a conversational tone and places lengthy technical proofs at the end of each chapter, helping readers grasp the main ideas before diving into the details of technical proofs.

Audience: This book is intended for advanced undergraduates and graduate students who are familiar with basic optimization concepts and are ready to explore more complex problems. A background in calculus, real analysis, linear algebra, and probability is recommended. It is appropriate for an introductory graduate-level course in continuous optimization. Practitioners and early career researchers will also find the book useful.



Primal Heuristics in Integer Programming

by Timo Berthold, Andrea Lodi, and Domenico Salvagnin

Publisher: Cambridge University Press

eISBN: 9781009574792

Published: 2025

DOI: 10.1017/9781009574792

About the book: Primal heuristics guarantee that feasible, high-quality solutions are provided at an early stage of the solving process, and thus are essential to the success of mixed-integer programming (MIP). By helping prove optimality faster, they allow MIP technology to extend to a wide variety of applications in discrete optimization.

This first comprehensive guide to the development and use of primal heuristics within MIP technology and solvers is ideal for computational mathematics graduate students and industry practitioners. Through a unified viewpoint, it gives a unique perspective on how state-of-the-art results are integrated within the branch-and-bound approach at the core of the MIP technology.

It accomplishes this by highlighting all the required knowledge needed to push the heuristic side of MIP solvers to their limit and pointing out what is left to do to improve them, thus presenting heuristic approaches for MIP as part of the MIP solving process.

Audience: This book is of interest to students, researchers, and practitioners in the field of Mixed Integer and Discrete Optimization as it provides a systematic view on the structure and dynamics of heuristics used in modern solvers, both commercial and open-source.

Chair's Column

The past three years have been a period of continued progress and consolidation for our SIAM Activity Group on Optimization (SIAG/OPT). Optimization continues to evolve through major research developments—ranging from scalable stochastic methods to enhanced capabilities in mixed-integer nonlinear programming, and early progress at the interface of optimization and quantum computing. These innovations reinforce optimization's foundational role in scientific computing, data science, and machine learning.

Membership.

SIAG/OPT remains one of SIAM's largest and most active activity groups. Although membership decreased after the pandemic, it rebounded around the 2023 SIAM Conference on Optimization, and we anticipate significant further growth in 2026. The group continues to strengthen its international reach and maintain strong links with related SIAGs, particularly Computational Science and Engineering, Data Science, Applied and Computational Discrete Algorithms (ACDA), Imaging Science, and Mathematics of Planet Earth.

Sponsored Events.

Over the 2023–2025 period, we co-sponsored the MOPTA conference at Lehigh University each year. MOPTA has drawn roughly 140 participants annually and offers the community a valuable platform for interaction, particularly because it is not affiliated with other professional societies. Additionally, the 2023 Gene Golub SIAM Summer School, also at Lehigh, emphasized optimization with a focus on quantum computing, further supporting the training of early-career researchers.

Prizes.

A major accomplishment of this board term was successfully establishing a permanent endowment for the SIAG on Optimization Test of Time Award. Through a $25,000 fundraising effort supported by numerous institutions and individuals, we ensured the long-term sustainability of this important recognition for impactful research contributions in optimization.

We are delighted to share excellent news regarding our prize nomination cycle. We have received an exceptionally strong set of nominations for all three 2026 SIAG/OPT prizes:

- SIAG on Optimization Best Paper Prize: 23 nominations
- SIAG on Optimization Early Career Prize: 14 nominations
- SIAG on Optimization Test of Time Award: 20 nominations

This is a truly phenomenal response and a testament to the engagement and commitment of our community. Thank you to everyone who took the time to identify deserving colleagues and submit nominations. Your efforts ensure that our SIAG continues to recognize and celebrate excellence across the spectrum of optimization research and impact. We look forward to celebrating the awardees together in Edinburgh!

2026 SIAM Conference on Optimization.

Following an open international call, the University of Edinburgh was selected as host for the 2026 SIAM Conference on Optimization, which will occur during June 2-5, 2026. The event will feature a diverse and distinguished lineup of plenary speakers and two mini-tutorials, with strong attention to inclusiveness across research areas and origins of speakers.

The conference will take place in the Old Town Campus which lies in the heart of the University quarter and is the location of the iconic McEwan Hall where the plenaries will be held:

https://www.uoecollection.com/conferences-events/venue-hubs/old-town-campus,

https://www.uoecollection.com/conferences-events/venue-hubs/old-town-campus/mcewan-hall

The campus is easily accessible by foot from Waverley Train Station and it is equally easy to move by foot between all the spaces used by the conference. It is also close to most of the attractions Edinburgh has to offer and to large green spaces such as Holyrood Park (Arthur's Seat) and the Meadows. We are getting close to 1,500 talks!

Acknowledgements.

Our group remains committed to delivering high-quality core activities: the triennial conference, our prizes, the newsletter, and the SIAM Journal on Optimization. During this period, we ensured that the SIAG website and newsletter remained up to date, serving as key channels for communication and engagement within our community.

I would like to thank my fellow officers, Coralia Cartis (Vice Chair), Gabriele Eichfelder (Program Director), and Juliane Mueller (Secretary) for their strong commitment and engagement—it was a great pleasure working with you! I also extend my gratitude to Miguel Anjos, Gabriele Eichfelder, and Lars Schewe for their outstanding work in leading the 2026 SIAM Conference on Optimization, as well as to all committee and local organizing committee members and mini-symposia organizers for their dedicated efforts.

Finally, we wish the new elected officers of the SIAG on Optimization a successful 2026–2028 term!

Luis Nunes Vicente (Chair, SIAG on Optimization)
Timothy J. Wilmott Endowed Chair Professor and Department Chair
Department of Industrial and Systems Engineering, Lehigh University

Comments from the Editors

Happy 2026, SIAG on Optimization! We are pleased to present this issue of Views and News. In this issue, we spotlight Jelena Diakonikolas, who comments on the role that open questions play in optimization research. While this article is intended to encourage discussion among all practitioners of optimization, Jelena leads with examples of open questions motivated by her own research, highlighting glaring gaps in knowledge in 1) the oracle complexity of algorithms for nonsmooth optimization, 2) the convergence properties of cyclic coordinate descent, 3) the oracle complexity of a family of problems subsuming min-max optimization, root-finding and fixed-point equations and 4) the connection between solvability of nonconvex problems in machine learning and the satisfaction of local error bounds.

We are excited for the 2026 SIAM Conference on Optimization, to be held June 2-5 in Edinburgh, and we look forward to seeing many members there.

All issues of *Views and News* are available online at https://siagoptimization.github.io/ViewsAndNews.

The SIAG on Optimization Views and News mailing list, where editors can be reached for feedback, is siagoptnews@lists.mcs.anl.gov. Suggestions for new issues, comments, and papers are always welcome.

Pietro Belotti
DEIB, Politecnico di Milano
Email: pietro.belotti@polimi.it
Web: https://belotti.faculty.polimi.it

Dmitriy Drusvyatskiy
Mathematics Department, University of Washington
Email: ddrusv@uw.edu
Web: https://sites.math.washington.edu/~ddrusv

Matt Menickelly
Argonne National Laboratory
Email: mmenickelly@anl.gov
Web: https://www.mcs.anl.gov/~menickmj