Information complexity of mixed-integer convex optimization

Phillip Kerger
Work with Amitabh Basu, Hongyi Jiang, and Marco Molinaro

JOHNS HOPKINS UNIVERSITY

June 2023, IPCO

Outline

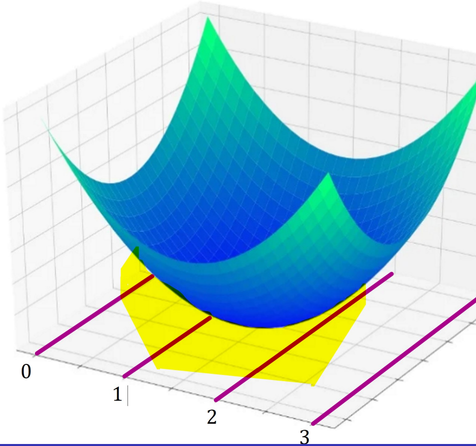
Outline

Setting

Mixed-integer convex optimization:

Min $f(x, y) : \mathbb{R}^{n+d} \to \mathbb{R}$ with

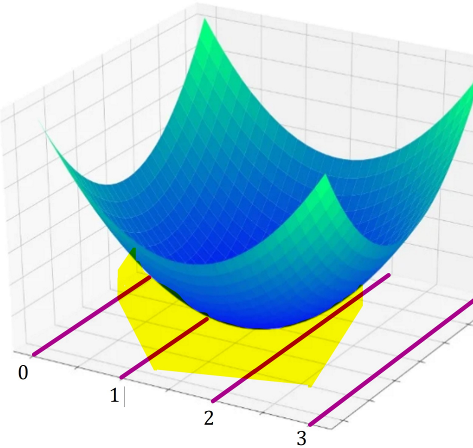
- f convex (possibly nonsmooth)
- $x \in \mathbb{Z}^n, y \in \mathbb{R}^d$
- $(x, y) \in C \subseteq \mathbb{R}^{n+d}$,
  C a convex set



Mixed-integer convex optimization:

Can't solve this exactly!

) Only ask for ″-solution:
Feasible point
$(x; y) : f(x; y) \quad OPT_f \quad ″$.



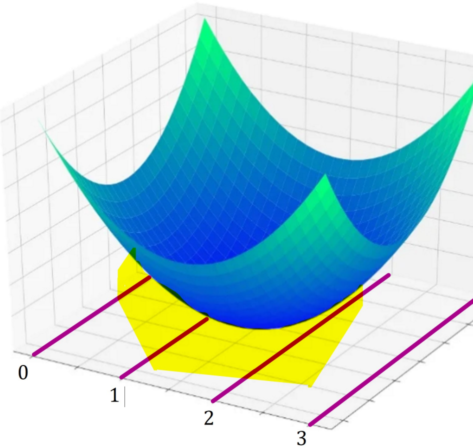
Setting

Mixed-integer convex optimization:

Can't solve this exactly!

) Only ask for "-solution:
Feasible point
$(x; y) : f(x; y) \quad OPT_f \quad$ ".

But if C is tiny...
there is also a problem



Setting

Parameterize the instances:

Definition

$I_{n;d;R;\ ;M}$ is the set of all MICO instances such that

(i) C is a subset of the box $[\ \ R;R]^{n+d}$.

(ii) Fiber containing optimum isn't degenerate, but a ball of size in it is feasible.

(iii) f is M-Lipschitz

Information Complexity:

How much information is needed to solve a problem?

Setting

Information Complexity:
How much information is needed to solve a problem?

Classical First-Order:
How many (exact) evaluations of function value and gradient or separating hyperplanes?

Setting

Information Complexity:
How much information is needed to solve a problem?

Classical First-Order:
How many (exact) evaluations of function value and gradient or separating hyperplanes?

Notice: Gradient length grows with dimension...

Information Complexity:
How much information is needed to solve a problem?

Classical First-Order:
How many (exact) evaluations of function value and gradient or separating hyperplanes?

Notice: Gradient length grows with dimension...
Gives **more information with dimension**!
Can we refine this in a meaningful way?

Setting

Information Complexity:
How much information is needed to solve a problem?

Classical First-Order:
How many (exact) evaluations of function value and gradient or separating hyperplanes?

Notice: Gradient length grows with dimension...
Gives **more information with dimension**!
Can we refine this in a meaningful way?

Goals of this work:

- Study information complexity beyond exact first-order model
- Tighten and generalize existing bounds

Oracles Using First-Order Information

What kinds of oracles?

Oracles Using First-Order Information

What kinds of oracles?

Definition

An *oracle using rst-order information* consists of two parts:

What kinds of oracles?

Definition

An *oracle using first-order information* consists of two parts:

1. For each point $\mathbf{z} \in \mathbb{R}^{n+d}$, a map $g_{\mathbf{z}}$ that maps instances to first-order information at $\mathbf{z}$:
    - Function value and gradient (objective)
    - Feasibility flag and separating hyperplane (feasibility)
    - A collection of these for all $\mathbf{z}$: "First order chart" G

Oracles Using First-Order Information

What kinds of oracles?

Definition

An *oracle using rst-order information* consists of two parts:

1. For each point $\mathbf{z}\ 2\ \mathrm{R}^{n+d}$, a map $g_{\mathbf{z}}$ that maps instances to first-order information at $\mathbf{z}$:
    - Function value and gradient (objective)
    - Feasibility flag and separating hyperplane (feasibility)
    - A collection of these for all $\mathbf{z}$: "First order chart" G

Oracles Using First-Order Information

What kinds of oracles?

Definition

An *oracle using rst-order information* consists of two parts:

1. For each point $\mathbf{z} \; 2 \; \mathrm{R}^{n+d}$, a map $g_{\mathbf{z}}$ that maps instances to first-order information at $\mathbf{z}$:
    - Function value and gradient (objective)
    - Feasibility flag and separating hyperplane (feasibility)
    - A collection of these for all $\mathbf{z}$: "First order chart" G
2. A set of *permissible queries* H
   $h \; 2 \; H$ functions taking first order information as input.

Oracles Using First-Order Information

What kinds of oracles?

Definition

An *oracle using rst-order information* consists of two parts:

1. For each point $\mathbf{z}\ 2\ \mathrm{R}^{n+d}$, a map $g_{\mathbf{z}}$ that maps instances to first-order information at $\mathbf{z}$:
   - Function value and gradient (objective)
   - Feasibility flag and separating hyperplane (feasibility)
   - A collection of these for all $\mathbf{z}$: "First order chart" G
2. A set of *permissible queries* H

   $h\ 2\ H$ functions taking first order information as input.

Oracles Using First-Order Information

What kinds of oracles?

Definition

An *oracle using rst-order information* consists of two parts:

1. For each point $\mathbf{z}\ 2\ \mathrm{R}^{n+d}$, a map $g_{\mathbf{z}}$ that maps instances to first-order information at $\mathbf{z}$:
   - Function value and gradient (objective)
   - Feasibility flag and separating hyperplane (feasibility)
   - A collection of these for all $\mathbf{z}$: "First order chart" G
2. A set of *permissible queries* H
   $h\ 2\ H$ functions taking first order information as input.

For query h at $\mathbf{z}$, for instance J, oracle answers $h(g_{\mathbf{z}}(J))$

! Pair $(G; H)$ defines an oracle!

Oracles Using First-Order Information

Oracles Using First-Order Information

Large class of oracles with answers using only rst-order info

Examples: Oracles Using First-Order Information

What kinds of oracles can we get?

Standard rst-order oracle : H has only the identity

Examples: Oracles Using First-Order Information

What kinds of oracles can we get?

Standard rst-order oracle : H has only the identity

Bit Queries : H output some designated bits of the function value/gradient/sep. hyperplane.

! How many bits of information does one need?

Examples: Oracles Using First-Order Information

What kinds of oracles can we get?

Standard first-order oracle: $\mathcal{H}$ has only the identity

Bit Queries: $\mathcal{H}$ output some designated bits of the function value/gradient/sep. hyperplane.
$\rightarrow$ How many bits of information does one need?

Directional Queries: Elements of $\mathcal{H}$ give the sign of the inner product of the gradient/sep. hyperplane with a vector $h_v = \text{sign}\langle v, g_z \rangle$.

Examples: Oracles Using First-Order Information

What kinds of oracles can we get?

- **Standard first-order oracle**: H has only the identity
- **Bit Queries**: H output some designated bits of the function value/gradient/sep. hyperplane.
  → *How many* **bits** *of information does one need?*
- **Directional Queries**: Elements of H give the sign of the inner product of the gradient/sep. hyperplane with a vector $h_v = \mathrm{sign}\langle v, g_z \rangle$.
- **General Binary Queries**: Let H contain all binary queries

Outline

1. Introduction
   - Setting and Goals
   - Oracles Using First-Order Information
2. Selected Results and Proof Ideas
   - Lower Bounds
   - Upper Bounds

Lower Bounds

Theorem (Lower-Bound for Standard Oracle)

$$\text{icomp} = \Omega\left(2^n \; d \log\left(\frac{MR}{\varepsilon}\right)\right).$$

[redacted]

$$\text{icomp} = \quad 2^n \; d \log \frac{MR}{"} \quad :$$

Improvement on existing best bound (by incorporating M; ").

Lower Bounds

[redacted]

$$\text{icomp} = 2^n \, d \log \frac{MR}{\varepsilon} :$$

- Improvement on existing best bound (by incorporating M, ε).
- Presents a 2^n "transfer" of continuous case LB to MICO

Lower Bounds

How to think about these lower-bounds...

Lower Bounds

How to think about these lower-bounds...

Some continuous instance on each integer ber:

Lower Bounds

How to think about these lower-bounds...
Some continuous instance on each integer fiber:

In unit cube, 2^n ` means you need to solve all of them.

Lower Bounds

Theorem (Lower-Bound for Standard Oracle)

$$\text{icomp} = \Omega\left(2^n \left(d \log\left(\frac{MR}{\varepsilon}\right)\right)\right).$$

- Improvement on existing best bound by incorporating M, ε.
- Presents a 2^n "transfer" of continuous case LB to MICO.

Lower Bounds

Theorem (Lower-Bound for Standard Oracle)

$$\text{icomp} = \Omega\left(2^n \, d \log\left(\frac{MR}{\varepsilon}\right)\right).$$

- Improvement on existing best bound by incorporating M, ε.
- Presents a 2^n "transfer" of continuous case LB to MICO.

Conjecture (Transferring Lower-Bounds from Continuous to Mixed)

- *Oracle (G, H) using first-order info*
- *A family of instances with a lower bound $\ell(d, R, \varepsilon, M)$ for the continuous case with this oracle*

Then there exist a family of instances with lower-bound $2^n \cdot \ell(d, R, \varepsilon, M)$ for the general MICO case.

Conjecture proven for pure optimization case!

Lower Bounds

Proof idea for optimization case of conjecture:

- "Place" one continuous instance on each integer fiber $\mathbf{x} \times \mathbb{R}^d$, $\mathbf{x} \in \{0,1\}^n$.

Lower Bounds

Proof idea for optimization case of conjecture:

- "Place" one continuous instance on each integer fiber $\mathbf{x} \quad R^d$, $\mathbf{x} \in \{0,1\}^n$.
- Want: No query in the full space to give more information than a query on one of the fiber functions

Lower Bounds

Proof idea for optimization case of conjecture:

- "Place" one continuous instance on each integer fiber $\mathbf{x} \times \mathbb{R}^d$, $\mathbf{x} \in \{0,1\}^n$.
- Want: No query in the full space to give more information than a query on one of the fiber functions
  $\to$ Extend the fiber functions to $\mathbb{R}^{n+d}$ such that any query not on a fiber can be simulated by a query on a fiber

Lower Bounds

Proof idea for optimization case of conjecture:

- "Place" one continuous instance on each integer fiber $\mathbf{x} \times \mathbb{R}^d$, $\mathbf{x} \in \{0,1\}^n$.
- Want: No query in the full space to give more information than a query on one of the fiber functions
  $\rightarrow$ Extend the fiber functions to $\mathbb{R}^{n+d}$ such that any query not on a fiber can be simulated by a query on a fiber

Construction:

Lower Bounds

Theorem

If H contains only binary functions, with access to oracle $(G; H)$

$$\text{icomp} = \tilde{\Omega}\left(d^{\frac{8}{7}}\right).$$

Lower Bounds

Theorem

If H contains only binary functions, with access to oracle $(G; H)$

$$\text{icomp} = \tilde{\Omega} \quad d^{\frac{8}{7}} \quad :$$

Uses ideas from recent results on memory-constrained algorithms.

Upper Bounds ($n = 0$ case)

Theorem

- *Instances with function values in* $[\ \ U; U]$
- *Permissible queries* H *are the bit or directional sign queries*

$$\text{icomp} = O \quad d^2 \log^2 \quad \frac{dMR}{''} \qquad \log \frac{U}{''}$$

Upper Bounds ($n = 0$ case)

Theorem

- *Instances with function values in $[-U; U]$*
- *Permissible queries H are the bit or directional sign queries*

$$\text{icomp} = O\left(d^2 \log^2\left(\frac{dMR}{\epsilon}\right) \log \frac{U}{\epsilon}\right)$$

Roughly LB under *exact* oracle times $d \log(\frac{dMR}{\epsilon})$
ϵ-approximate gradient contains roughly $d \log(\frac{1}{\epsilon})$ bits

Conjecture ("The above upper-bound is good")

When permissible queries H are binary functions,

$$\text{icomp} = \Omega\left(d^2 \log\left(\frac{MR}{\epsilon}\right)^2\right)$$

.

Upper Bounds

Theorem (Similar result in the general MICO case)

For the general MICO case with binary permissible queries H,

$$\mathrm{icomp} = O \quad 2^n d\,(n+d)^2 \log^2 \quad \frac{dMR}{\min f \quad ; 1g''}$$

Theorem

- **Finitely many** *instances* $I \quad I_{n;d;R;\ ;M}$
- *H for the oracle consisting of all binary functions*

$$icomp = O \quad \log j I j + d \log \quad \frac{MR}{''}$$

Compare to:

$\tilde{\Omega} \quad d^{\frac{8}{7}}$ lower-bound

$\Omega \quad d^2 \log \quad \frac{MR}{''} \quad ^2$ conjectured lower-bound

Proof idea for feasibility case:
Maintain a family $U \subseteq I$ of the instances that are possible, and a polyhedron P containing C.

Start with $U = I$ and $P = [-R, R]^d$.
We will be able to either **reduce** $|U|$ **or** $\mathrm{vol}(P)$ **by a constant fraction** with each query.

Upper Bounds

Proof idea for feasibility case:
Maintain a family $U \subseteq I$ of the instances that are possible, and a polyhedron P containing C.

Start with $U = I$ and $P = [-R, R]^d$.
We will be able to either reduce $|U|$ or $\mathrm{vol}(P)$ by a constant fraction with each query.

While $|U| > 1$ do the following:

- Set p equal to be the centroid of P. If the separation oracle at p reports that $p \in C$, then we return p.

- Otherwise...

Upper Bounds

Case 1: For all v, no more than half the instances $C' \in U$ give the answer $g_p^{sep}(C') = v$.

Upper Bounds

Case 1: For all v, no more than half the instances $C' \in U$ give the answer $g_p^{sep}(C') = v$.

$\Rightarrow$ there is a set of answers $V \subseteq \mathbb{R}^d$ such that between $\frac{1}{4}|U|$ and $\frac{3}{4}|U|$ of the sets give an answer in V.

Upper Bounds

Case 1: For all v, no more than half the instances $C' \in U$ give the answer $g_p^{sep}(C') = v$.

$\Rightarrow$ there is a set of answers $V \subseteq \mathbb{R}^d$ such that between $\frac{1}{4}|U|$ and $\frac{3}{4}|U|$ of the sets give an answer in V.

Query whether the true instance has $g_p^{sep}(C) \in V$.

$\Rightarrow$ Size of U decreases by at least $1/4$.

Upper Bounds

Case 1: For all v, no more than half the instances $C' \in U$ give the answer $g_p^{sep}(C') = v$.

$\Rightarrow$ there is a set of answers $V \subseteq \mathbb{R}^d$ such that between $\frac{1}{4}|U|$ and $\frac{3}{4}|U|$ of the sets give an answer in V.
Query whether the true instance has $g_p^{sep}(C) \in V$.
$\Rightarrow$ Size of U decreases by at least $1/4$.

Case 2: There exists $v \in \mathbb{R}^d$ such that more than half of the instances have $g_p^{sep}(C') = v$.

Upper Bounds

Case 1: For all v, no more than half the instances $C' \in U$ give the answer $g_p^{sep}(C') = v$.

$\Rightarrow$ there is a set of answers $V \subseteq \mathbb{R}^d$ such that between $\frac{1}{4}|U|$ and $\frac{3}{4}|U|$ of the sets give an answer in V.
Query whether the true instance has $g_p^{sep}(C) \in V$.
$\Rightarrow$ Size of U decreases by at least $1/4$.

Case 2: There exists $v \in \mathbb{R}^d$ such that more than half of the instances have $g_p^{sep}(C') = v$.
Query whether the true instance has $g_p^{sep}(C) = v$.
$\Rightarrow$ Either size of U decreases by at least half, or we get an exact separating hyperplane for the true instance to reduce the volume of P by at least $1/e$ (Grünbaum's theorem).

Case 1: For all v, no more than half the instances $C' \in U$ give the answer $g_p^{sep}(C') = v$.

$\Rightarrow$ there is a set of answers $V \subset \mathbb{R}^d$ such that between $\frac{1}{4}|U|$ and $\frac{3}{4}|U|$ of the sets give an answer in V.
Query whether the true instance has $g_p^{sep}(C) \in V$.
$\Rightarrow$ Size of U decreases by at least $1/4$.

Case 2: There exists $v \in \mathbb{R}^d$ such that more than half of the instances have $g_p^{sep}(C') = v$.
Query whether the true instance has $g_p^{sep}(C) = v$.
$\Rightarrow$ Either size of U decreases by at least half, or we get an exact separating hyperplane for the true instance to reduce the volume of P by at least $1/e$ (Grünbaum's theorem).

Reducing U can only happen $\log(|U|)$ times, and reducing the volume of P can only happen $d\log(\frac{MR}{\min\{\varepsilon, 1\}})$ times.

Thank you for your attention!

Questions?