BnB-PEP: A Unified Methodology for Constructing Optimal Optimization Methods

Shuvomoy Das Gupta
MIT Operations Research Center, *sdgupta@mit.edu*
joint work with Bart Van Parys & Ernest K. Ryu

Take home messages

- First-order methods are very popular to solve large-scale optimization + machine learning problems
- We want to find the optimal≡fastest first-order methods
- BnB-PEP (Branch-and-Bound Performance Estimation Programming) is a unified methodology to find the optimal first-order methods
  - applicable convex/nonconvex smooth/nonsmooth problems

What is BnB-PEP?

- Shows that optimal first-order methods are themselves solutions to optimization problems
- Proposes a custom branch-and-bound algorithm to solve these optimization to certifiable global optimality and in a practical time scale
- For all the applications considered, finds first-order methods that outperforms state-of-the-art results
- Provides a way to systematically generate mathematical convergence proofs

Setup

- *Optimization problem.* We want to solve minimize $f(x)$, where $x \in \mathbf{R}^d$ is the decision variable, f lies in some function class $\mathcal{F}$, and $d \gg N$
- *Function class.* $\mathcal{F}$ can be smooth/nonsmooth convex/nonconvex etc
- *Class of fixed-step first-order methods $\mathcal{M}_N$.* Any method $M \in \mathcal{M}_N$ can be described as:

$$\begin{aligned} &\text{pick initial point } x_0 \\ &x_1 = x_0 - h_{1,0} f'(x_0) \\ &x_2 = x_1 - h_{2,0} f'(x_0) - h_{2,1} f'(x_1) \\ &\vdots \\ &x_N = x_{N-1} - \sum_{i=0}^{N-1} h_{N,i} f'(x_i) \end{aligned}$$

for some stepsizes $\{h_{i,j}\}$

- *Performance measure $\mathcal{E}$.* $\mathcal{E}$ measures the performance of method $M \in \mathcal{M}_N$ on $f \in \mathcal{F}$, can be $f(x_N) - f(x_\star)$, $\|x_N - x_\star\|^2$, $\|\nabla f(x_N)\|^2$, and so on

Optimal method

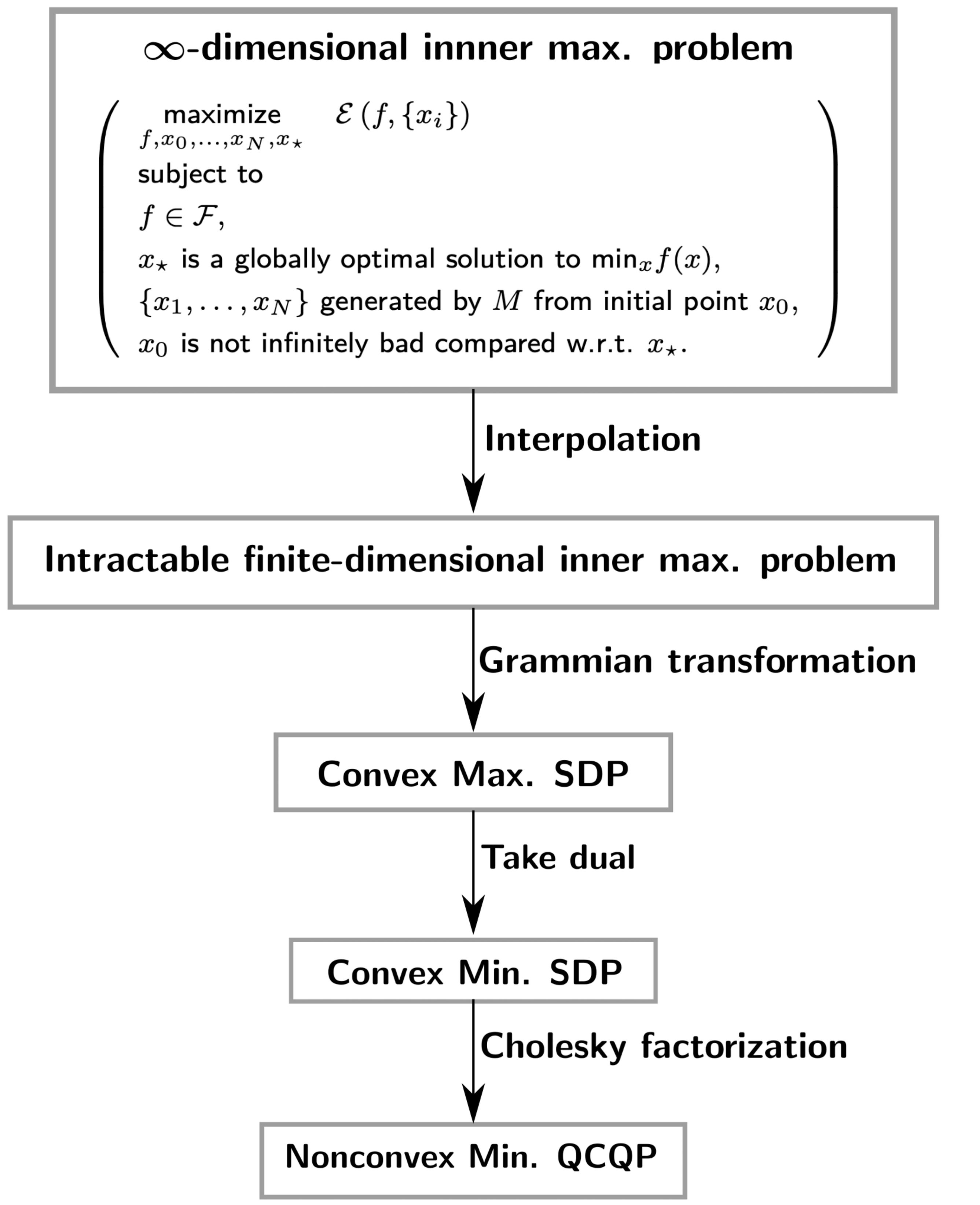
- *Definition of the optimal method.* The optimal method $M_\star \in \mathcal{M}_N$ can be constructed by solving the min-max problem:

$$\left[\underset{M \in \mathcal{M}_N}{\text{minimize}} \left(\begin{array}{l} \underset{f, x_0, \ldots, x_N, x_\star}{\text{maximize}} \quad \mathcal{E}(f, \{x_i\}) \\ \text{subject to} \\ f \in \mathcal{F}, \\ x_\star \text{ is a globally optimal solution to } \min_x f(x), \\ \{x_1, \ldots, x_N\} \text{ generated by } M \text{ from initial point } x_0, \\ x_0 \text{ is not infinitely bad compared w.r.t. } x_\star. \end{array} \right) \right] \quad (\mathcal{P})$$

- $(\mathcal{P})$ is an infinite-dimensional and intractable problem, BnB-PEP transforms it into an equivalent nonconvex but practically tractable QCQP

Problem transformation

- *Inner problem transformation.* Transform the inner maximization problem



- *Outer problem transformation.* Transform the min-max problem $(\mathcal{P})$ into a nonconvex QCQP

BnB-PEP algorithm

- We propose the BnB-PEP algorithm to construct the optimal method by solving the nonconvex QCQP to global optimality and it has three stages
- **Stage 1** finds a feasible solution by solving a convex SDP
- **Stage 2** warms-starts with the stage 1 solution and finds a locally optimal solution using a nonlinear interior-point method
- **Stage 3** warm-starts with the stage 2 solution and finds a globally optimal solution using a customized spatial branch-and-bound algorithm

Customized branch-and-bound algorithm

- Computes tighter valid bounds on the variables using SDP-based relaxation
- Finds an improved lower-bound on the objective value via lazy constraints
- Implements custom-heuristics by exploiting structure at each node of the BnB tree

Runtime comparison

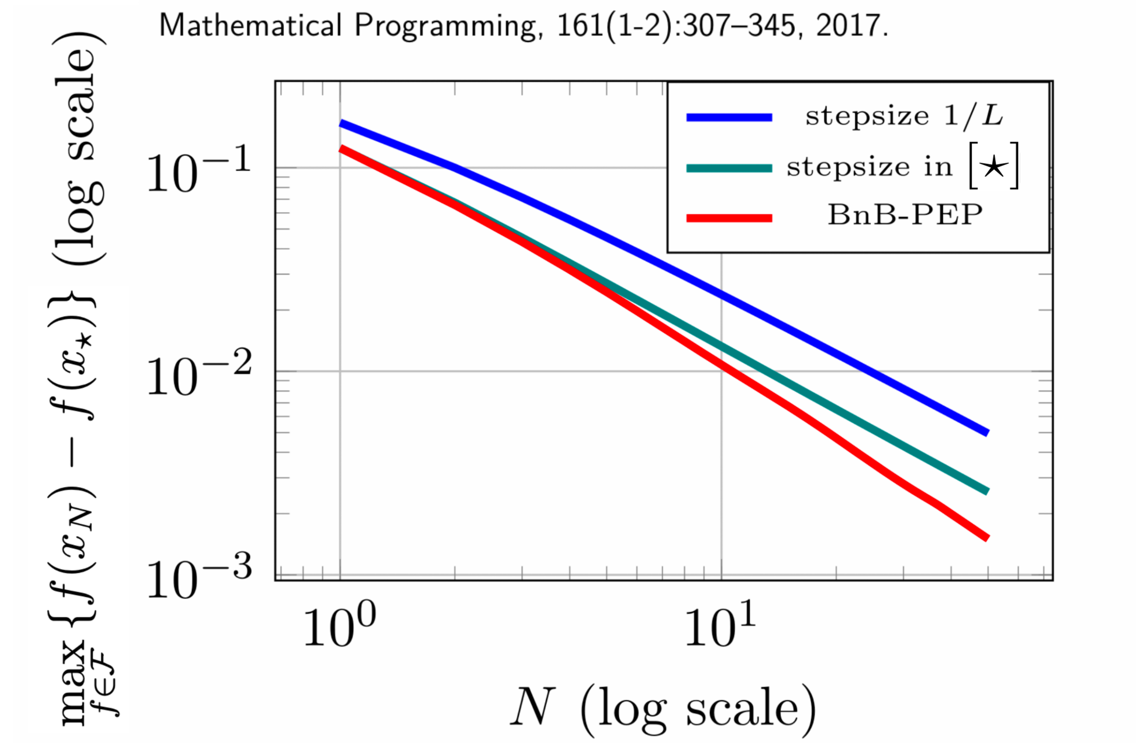
Table 1: Runtime comparison between the BnB-PEP Algorithm and off-the-shelf spatial branch-and-bound algorithm of `Gurobi` executed on MIT Supercloud.

Algorithm	BnB-PEP Algorithm on a core-i7 16 GB laptop	Default `Gurobi` on `MIT Supercloud`
$N = 1$	0.343 s	7 h 35 m
$N = 2$	0.493 s	1 d 8 h
$N = 3$	1.864 s	5 d 19 h
$N = 4$	9.146 s	More than a week

Constructing optimal algorithms numerically

Example. Optimal momentum-less GD $x_i = x_{i-1} - \frac{h_{i-1}}{L} \nabla f(x_{i-1})$ for minimizing L-smooth convex f

[★] A. B. Taylor, J. M. Hendrickx, and F. Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. Mathematical Programming, 161(1-2):307–345, 2017.



Generating proofs using BnB-PEP

- The potential function approach is used widely to construct proofs in optimization
- BnB-PEP can construct the best algorithm w.r.t. to a given potential function and systematically generate analytical convergence proof as a by-product
- *Example theorem where proof is generated by BnB-PEP.* Let f be a 1-weakly convex function with L-bounded subgradients. Assume the initial condition $f(y_0) - f(x_\star) + \|x_0 - x_\star\|^2 \leq R^2$. Then,

$$x_{k+1} = x_k - \frac{\sqrt{1 + 4R^2(N+1)/L^2}}{2(N+1)} f'(x_k),$$

exhibits the rate

$$\frac{1}{N+1} \sum_{i=0}^{N} \|\nabla f_{1/2}(x_i)\|^2 \leq \frac{L^2 \left(2\sqrt{1 + 4R^2(N+1)/L^2} - 1\right)}{N+1},$$

which improves upon the prior state-of-the-art rate