Smooth Constraint Convex Minimization via Conditional Gradients

Sebastian Pokutta

Zuse Institute Berlin
Technische Universität Berlin

Nice, 09/2019

ZUSE INSTITUTE BERLIN

twitter: @spokutta

Joint Work with… (in random order)

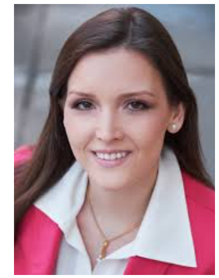
Alexandre D'Aspremont

Thomas Kerdreux

Gábor Braun

Cyrille Combettes

Swati Gupta

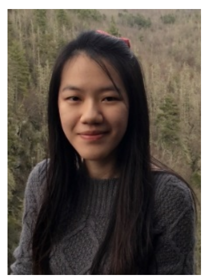


Jelena Diakonikolas

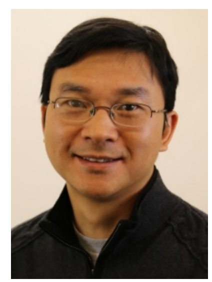
Alejandro Carderera

Robert Hildebrand

Stephen Wright



Yi Zhou



George Lan

Dan Tu

Daniel Zink

ZUSE INSTITUTE BERLIN

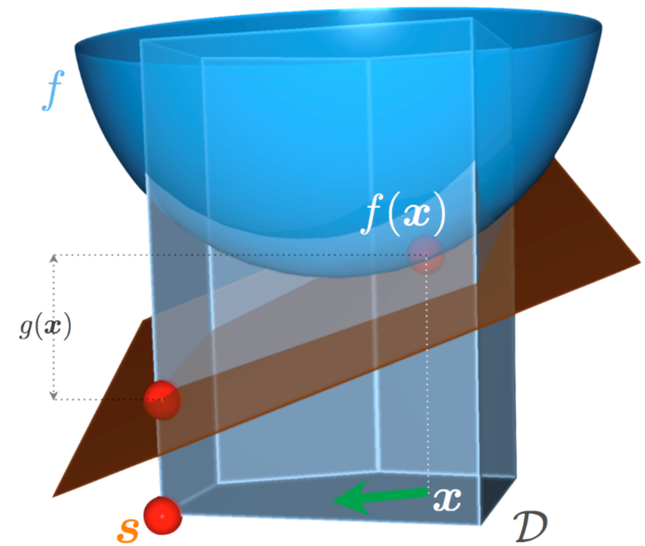
(Constraint) Convex Optimization

Convex Optimization:

Given a feasible region P solve the optimization problem:

$$\min_{x \in P} \; f(x),$$

where f is a convex function (+ extra properties).



Source: [Jaggi 2013]

Our setup.

1. Access to P. Linear Optimization (LO) oracle: Given linear objective c
   $x \leftarrow \arg\min_{v \in P} c^T v$
2. Access to f. First-Order (FO) oracle: Given x return
   $\nabla f(x)$ and $f(x)$

=> Complexity of convex optimization relative to LO/FO oracle

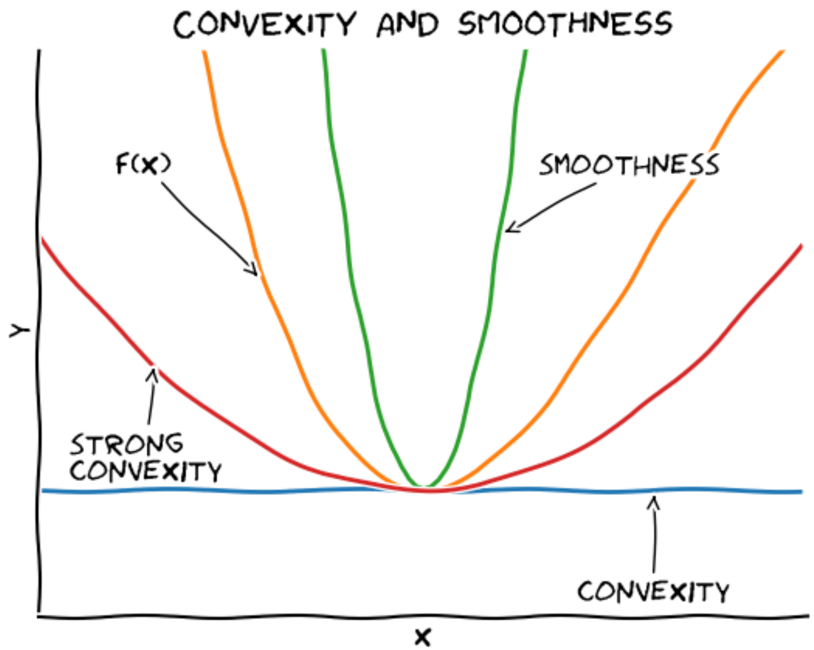
Why would you care for constraint convex optimization?

Setup captures various problems in Machine Learning, e.g.:

1. OCR (Structured SVM Training)
    1. Marginal polytope over chain graph of letters of word and quadratic loss
2. Video Co-Localization
    1. Flow polytope and quadratic loss
3. Lasso
    1. Scaled ℓ_1 -ball and quadratic loss (regression)
4. Regression over structured objects
    1. Regression over convex hull of combinatorial atoms
5. Approximation of distributions
    1. Bayesian inference, sequential kernel herding, …

Smooth Convex Optimization 101

Basic notions



Let $f: R^n \rightarrow R$ be a function. We will use the following basic concepts:

Smoothness. $f(y) \leq f(x) + \nabla f(x)^T (y-x) + L/2 \; ||x-y||^2$

Convexity. $f(y) \geq f(x) + \nabla f(x)^T (y-x)$

Strong Convexity. $f(y) \geq f(x) + \nabla f(x)^T (y-x) + \mu/2 \; ||x-y||^2$

=> Use unclear. Next step: Operationalize notions!

Measures of Progress: Smoothness and Idealized Gradient Descent

Consider an iterative algorithm of the form:

$$x_{t+1} \leftarrow x_t - \eta_t d_t$$

By definition of smoothness. $f(x_t) - f(x_{t+1}) \geq \eta_t \nabla f(x_t)^T d_t - \eta_t^2 \frac{L}{2} \|d_t\|^2$

Smoothness induces primal progress. Optimizing right-hand side:

$$f(x_t) - f(x_{t+1}) \geq \frac{(\nabla f(x_t)^T d_t)^2}{2L\|d_t\|^2} \quad \text{for} \quad \eta_t^* = \frac{\nabla f(x_t)^T d_t}{L\|d_t\|^2}$$

Idealized Gradient Descent (IGD). Choose $d_t \leftarrow x_t - x^*$ (non-det!)

$$f(x_t) - f(x_{t+1}) \geq \frac{(\nabla f(x_t)^T (x_t - x^*))^2}{2L\|x_t - x^*\|^2} \quad \text{for} \quad \eta_t^* = \frac{\nabla f(x_t)^T (x_t - x^*)}{L\|x_t - x^*\|}$$

Measures of Optimality: Convexity

Recall convexity: $f(y) \geq f(x) + \nabla f(x)^T (y-x)$

Primal bound from Convexity. $x \leftarrow x_t$ and $y \leftarrow x^* \in \arg\min_{x \in P} f(x)$:

$$h_t := f(x_t) - f(x^*) \leq \nabla f(x_t)^T (x_t - x^*)$$

Plugging this into the progress from IGD and $\|x_t - x^*\| \leq \|x_0 - x^*\|$.

$$f(x_t) - f(x_{t+1}) \geq \frac{(\nabla f(x_t)^T (x_t - x^*))^2}{2L\|x_t - x^*\|^2} \geq \frac{h_t^2}{2L\|x_0 - x^*\|^2}$$

Rearranging provides contraction and convergence rate.

$$h_{t+1} \leq h_t \cdot \left(1 - \frac{h_t}{2L\|x_0 - x^*\|^2}\right) \quad \Rightarrow \quad h_T \leq \frac{2L\|x_0 - x^*\|^2}{T+4}$$

Measures of Optimality: Strong Convexity

Recall strong convexity: $f(y) \geq f(x) + \nabla f(x)^T (y-x) + \frac{\mu}{2} \|x-y\|^2$

Primal bound from Strong Convexity. $x \leftarrow x_t$ and $y \leftarrow x_t - \gamma(x_t - x^*)$

$$h_t := f(x_t) - f(x^*) \leq \frac{(\nabla f(x_t)^T (x_t - x^*))^2}{2\mu \|x_t - x^*\|^2}$$

Plugging this into the progress from IGD.

$$f(x_t) - f(x_{t+1}) \geq \frac{(\nabla f(x_t)^T (x_t - x^*))^2}{2L \|x_t - x^*\|^2} \geq \frac{\mu}{L} h_t$$

Rearranging provides contraction and convergence rate.

$$h_{t+1} \leq h_t \cdot \left(1 - \frac{\mu}{L}\right) \quad \Rightarrow \quad h_T \leq e^{-\frac{\mu}{L} T} \cdot h_0$$

From IGD to actual algorithms

Consider an algorithm of the form:

$$x_{t+1} \leftarrow x_t - \eta_t d_t$$

Scaling condition (Scaling). Show there exist α_t with

$$\nabla f(x_t)^T d_t / ||d_t|| \geq \alpha_t \nabla f(x_t)^T (x_t - x^*) / ||x_t - x^*||$$

=> Lose an α_t^2 factor in iteration t. Bounds and rates follow.

Example. (Vanilla) Gradient Descent with $d_t \leftarrow \nabla f(x_t)$

$$\nabla f(x_t)^T d_t / ||d_t|| = ||\nabla f(x_t)||^2 \geq 1 \cdot \nabla f(x_t)^T (x_t - x^*) / || x_t - x^* ||$$

=> TODAY: No convergences proofs. Just establishing (Scaling).

Conditional Gradients
(a.k.a. Frank-Wolfe Algorithm)

Conditional Gradients a.k.a. Frank-Wolfe Algorithm

Algorithm 1 Frank-Wolfe Algorithm [Frank and Wolfe, 1956]

Input: smooth convex f function with curvature C, $x_1 \in P$

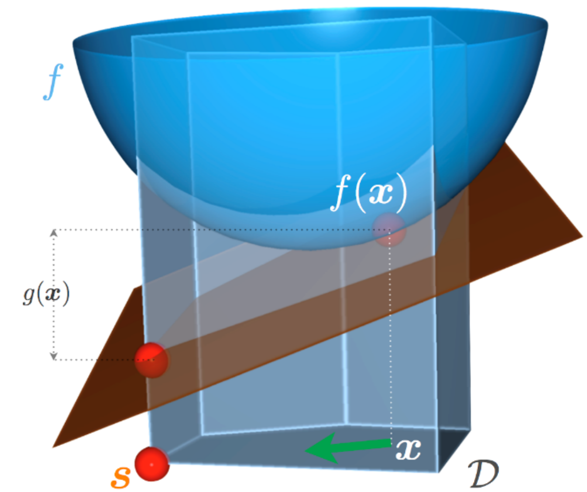
Output: x_t points in P

1: **for** $t = 1$ **to** $T - 1$ **do**

2: $\quad v_t \leftarrow \mathrm{LP}_P(\nabla f(x_t))$

3: $\quad x_{t+1} \leftarrow (1 - \gamma_t)x_t + \gamma_t v_t$ with $\gamma_t := \frac{2}{t+2}$

4: **end for**

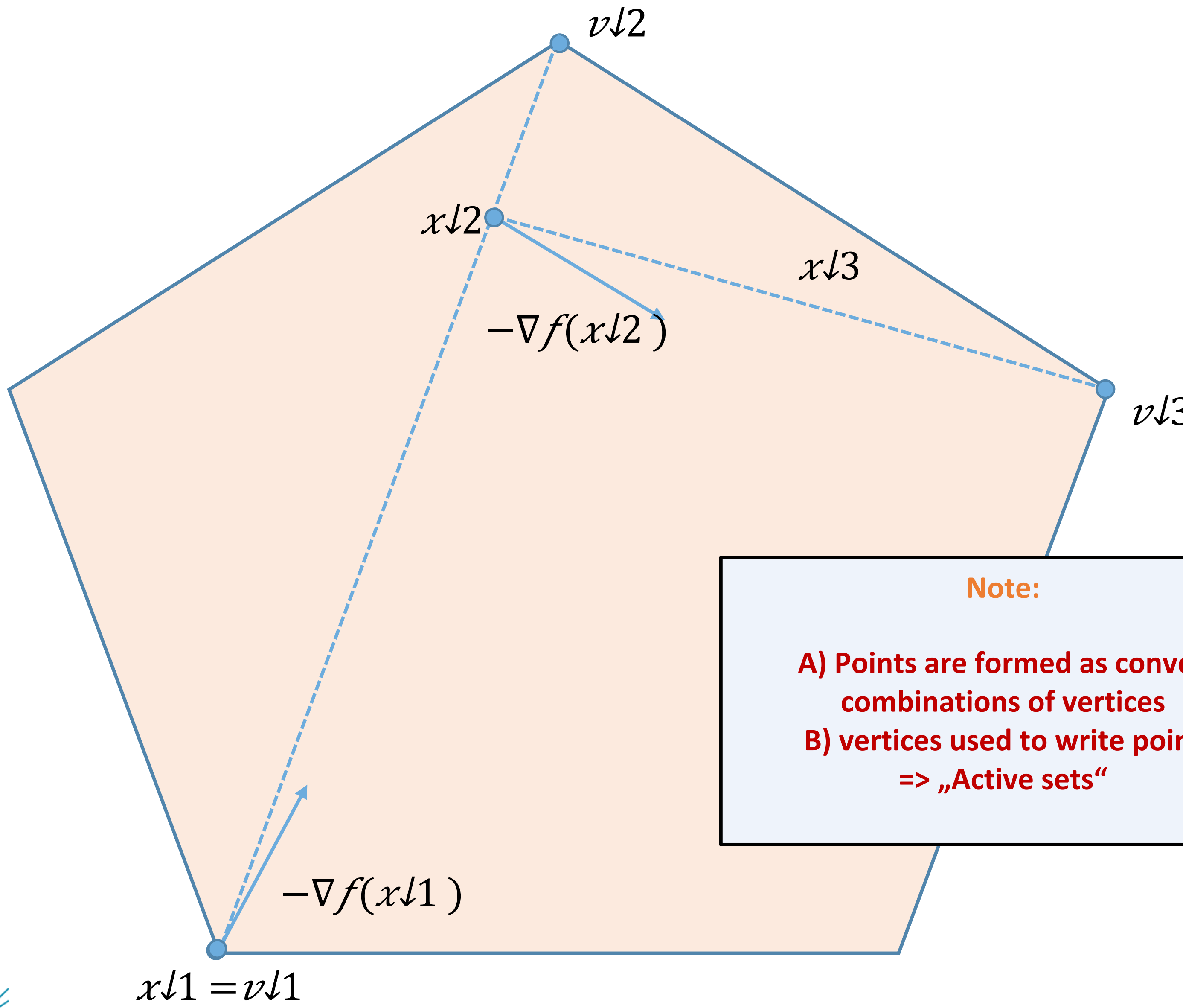


Source: [Jaggi 2013]

1. Advantages
   1. Extremely simple and robust: no complicated data structures to maintain
   2. Easy to implement: requires only a linear optimization oracle (first order method)
   3. Projection-free: feasibility via linear optimization oracle
   4. Sparse distributions over vertices: optimal solution is convex comb. (enables sampling)
2. Disadvantages
   1. Suboptimal convergence rate of $O(1/T)$ in the worst-case

=> Despite suboptimal rate often used because of simplicity

Conditional Gradients a.k.a. Frank-Wolfe Algorithm



Note:

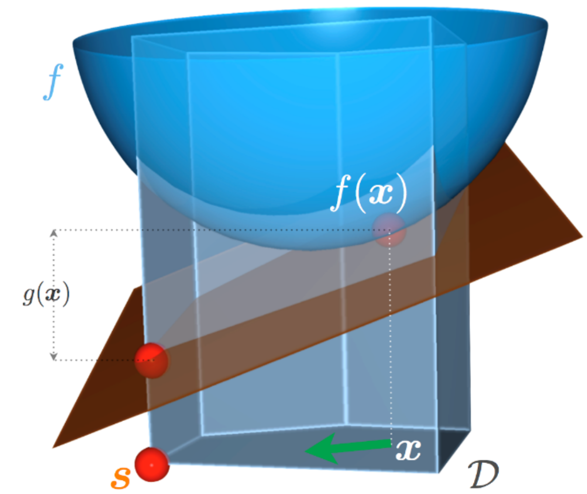
A) Points are formed as convex combinations of vertices
B) vertices used to write point => „Active sets“

Conditional Gradients a.k.a. Frank-Wolfe Algorithm

Algorithm 1 Frank-Wolfe Algorithm [Frank and Wolfe, 1956]

Input: smooth convex f function with curvature C, $x_1 \in P$
Output: x_t points in P

1: **for** $t = 1$ **to** $T - 1$ **do**
2: $\quad v_t \leftarrow \mathrm{LP}_P(\nabla f(x_t))$
3: $\quad x_{t+1} \leftarrow (1 - \gamma_t)x_t + \gamma_t v_t$ with $\gamma_t := \frac{2}{t+2}$
4: **end for**



Source: [Jaggi 2013]

Establishing (Scaling).

FW algorithm takes direction $d_t = x_t - v_t$. Observe

$$\nabla f(x)^T (x_t - v_t) \geq \nabla f(x)^T (x_t - x^*)$$

Hence with $\alpha_t = ||x_t - x^*||/D$ with D diameter of P:

$$\nabla f(x)^T (x_t - v_t)/||x_t - v_t|| \geq ||x_t - x^*||/D \cdot \nabla f(x)^T (x_t - x^*)/||x_t - x^*||$$

=> This α_t is sufficient for $O(1/t)$ convergence but better??

The strongly convex case
Linear convergence in special cases

If f is strongly convex we would expect a linear rate of convergence.

Obstacle.

$$\nabla f(x)^T (x_t - v_t)/\|x_t - v_t\| \geq \|x_t - x^*\|/D \cdot \nabla f(x)^T (x_t - x^*)/\|x_t - x^*\|$$

Special case $x^* \in \mathrm{rel.int}(P)$, Say $B(x^*, 2r) \subseteq P$. Then:

Theorem [Marcotte, Guélat '86]. After a few iterations

$$\nabla f(x)^T (x_t - v_t)/\|x_t - v_t\| \geq r/D \cdot \nabla f(x)^T (x_t - x^*)/\|x_t - x^*\|$$

and linear convergence follows via (Scaling).

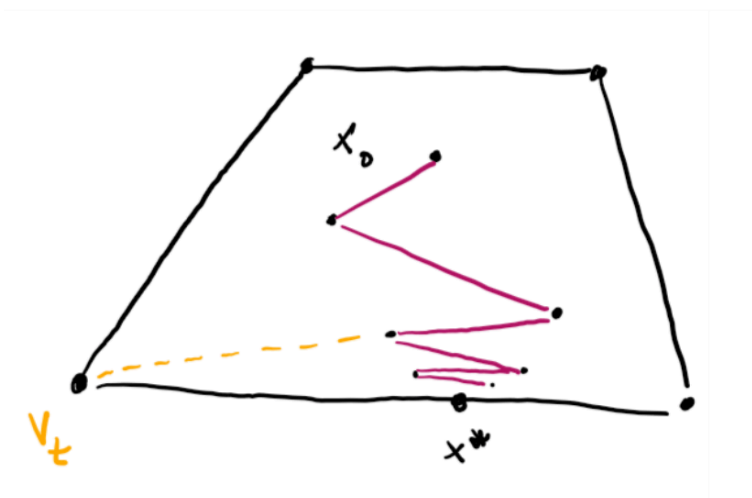
The strongly convex case
Is linear convergence in general possible?

(Vanilla) Frank-Wolfe cannot achieve linear convergence in general:

Theorem [Wolfe '70]. x^* on boundary of *P*. For any $\delta > 0$ for infinitely many *t*:

$$f(x_t) - f(x^*) \geq 1/t^{1+\delta}$$

Issue: zig-zagging (b/c first order opt)



[Wolfe '70] proposed Away Steps

The strongly convex case
Linear convergence in general

First linear convergence result (in general) [Garber, Hazan '13]

1. Simulating (theoretically efficiently) a stronger oracle rather using Away Steps
2. Involved constants are *extremely large* => algorithm unimplementable

Linear convergence for implementable variants [Lacoste-Julien, Jaggi '15]

1. (Dominating) Away-steps are enough
2. Includes most known variants: Away-Step FW, Pairwise CG, Fully-Corrective FW, Wolfe's algorithm, …
3. Key ingredient: There exists $w(P)$ (depending on polytope P (only!)) s.t.

$$\nabla f(x)^T (a_t - v_t) \geq w(P) \nabla f(x)^T (x_t - x^*) / \| x_t - x^* \|$$

($d_t = a_t - v_t$ is basically the direction that either variant dominates)

=> Linear convergence via (Scaling)

ZUSE INSTITUTE BERLIN

TU berlin

Many more variants and results…

Recently there has been a lot of work on Conditional Gradients, e.g.,

1. Linear convergence for conditional gradient sliding [Lan, Zhou ‘14]
2. Linear convergence for (some) non-strongly convex functions [Beck, Shtern ‘17]
3. Online FW [Hazan, Kale ‘12, Chen et al ‘18]
4. Stochastic FW [Reddi et al ‘16] and Variance-Reduced Stochastic FW [Hazan, Luo ’16, Chen et al ‘18]
5. In-face directions [Freund, Grigas ‘15]

… and *many more!!*

=> Very competitive and versatile in real-world applications

Revisiting Conditional Gradients
Lazification

Bottleneck 1: Cost of Linear Optimization
Drawbacks in the context of hard feasible regions

Basic assumption of conditional gradient methods:

Linear Optimization is cheap

As such accounted for as $\mathcal{O}(1)$. This assumption is not warranted if:

1. Linear Program of feasible region is huge
   1. Large shortest path problems
   2. Large scheduling problems
   3. Large-scale learning problems

2. Optimization over feasible region is NP-hard
   1. TSP tours
   2. Packing problems
   3. Virtually *every* real-world combinatorial optimization problem

Rethinking CG in the context of expensive oracle calls

Basic assumption for us:

*Linear Optimization is **not** cheap*

(Think: hard IP can easily require an hour to be solved => one call/it unrealistic)

1. Questions:
    1. Is it necessary to call the oracle in each iteration?
    2. Is it necessary to compute *(approximately) optimal* solutions?
    3. Can we reuse information?

2. Theoretical requirements
    1. Achieve identical convergence rates, otherwise any speedup will be washed out

3. Practical requirements
    1. Make as few oracle calls as possible

Lazification approach using weaker oracle

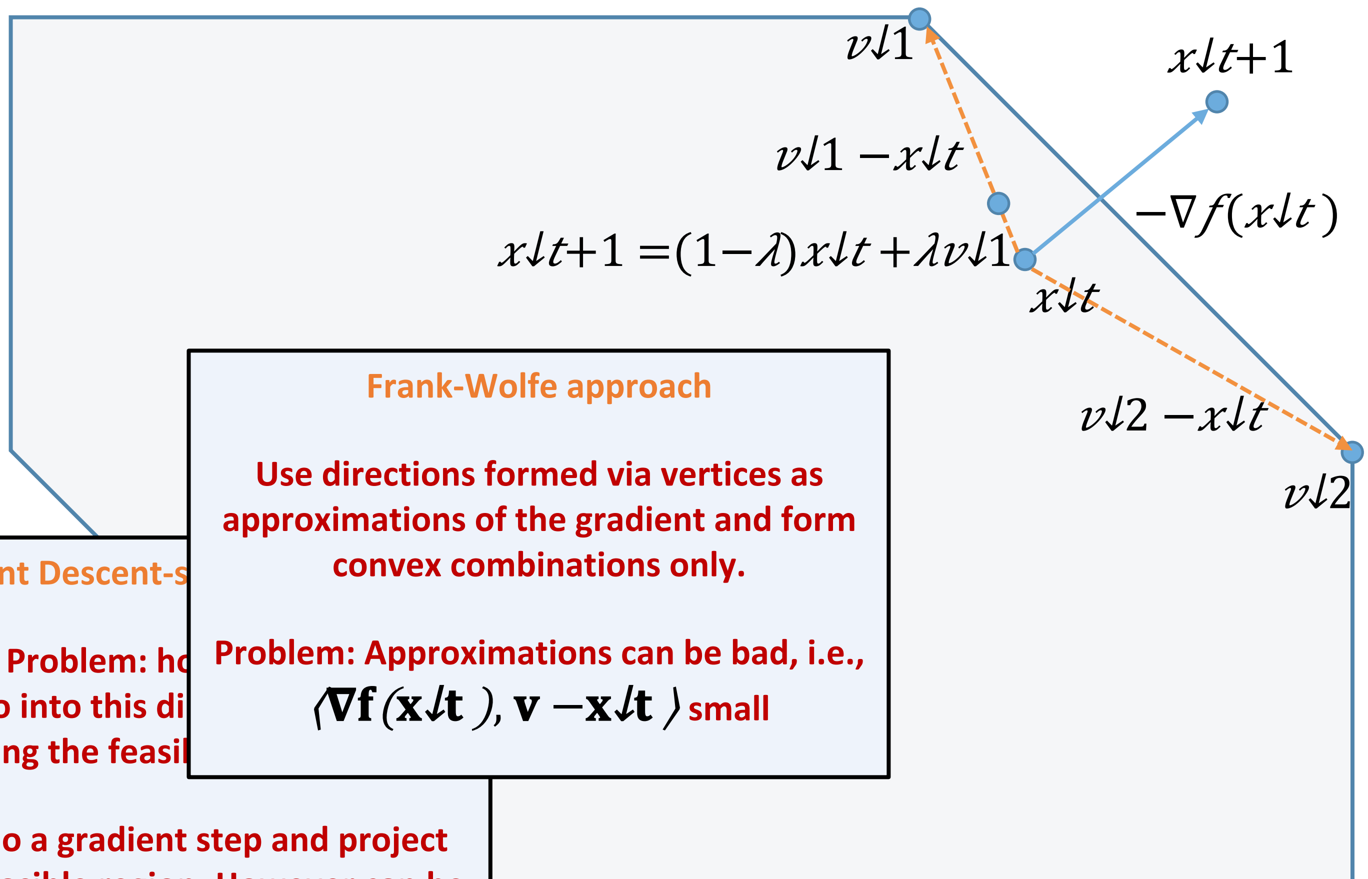
Oracle 1 Weak Separation Oracle $\mathrm{LPsep}_P(c, x, \Phi, K)$

Require: $c \in \mathbb{R}^n$ linear objective, $x \in P$ point, $K \geq 1$ accuracy, $\Phi > 0$ objective value;
Ensure: Either (1) $y \in P$ vertex with $c(x - y) > \Phi/K$, or (2) **false**: $c(x - z) \leq \Phi$ for all $z \in P$.

1. Interpretation of Weak Separation Oracle: *Discrete Gradient Directions*
   1. Either a new point $y \in P$ that improves the current objective by at least Φ/K (positive call)
   2. Or it asserts that all other points $z \in P$ improve no more than Φ (negative call)
2. Lazification approach [Braun, P., Zink '17]
   1. Use weaker oracle that allows for caching and early termination (no more expensive than LP)
   2. Advantage: huge speedups in wall-clock time when LP is hard to solve
      1. For hard LPs speedups can be as large as $10\uparrow 7$
   3. Disadvantage: weak separation oracle produces even weaker approx. than LP oracle
      1. Actual progress in iterations can be worse than with LP oracle
      2. Advantage vanishes if LP is very cheap and can be worse than original algorithm
      3. Caching is not "smart": it simple iterates over the already seen vertices
3. Optimal complexity for Weak Separation Oracle [Braun, Lan, P., Zhou '17]

Revisiting Conditional Gradients Blending

Bottleneck 2: Quality of gradient approximation
Frank-Wolfe vs. Projected Gradient Descent



Gradient Descent-s

Problem: ho
can we go into this di
leaving the feasi

Solution: do a gradient step and project back into feasible region. However can be *very* expensive

Frank-Wolfe approach

Use directions formed via vertices as approximations of the gradient and form convex combinations only.

Problem: Approximations can be bad, i.e., $\langle \nabla f(x_t), v - x_t \rangle$ small

⇒ Tradeoff between ensured feasibility and quality of gradient approximations!

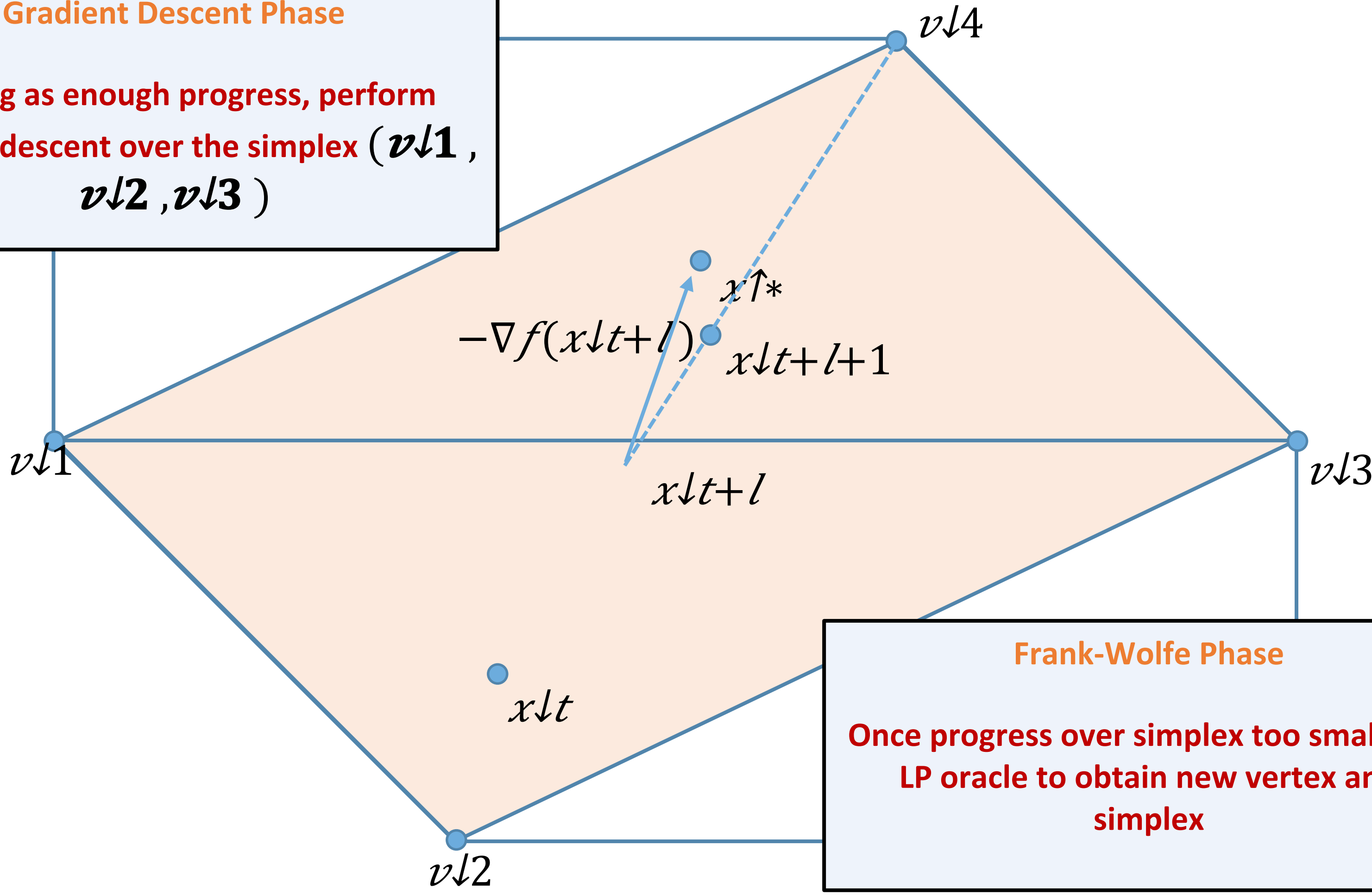
ZUSE INSTITUTE BERLIN

TU berlin

Blending of gradient steps and Frank-Wolfe steps

Gradient Descent Phase

As long as enough progress, perform gradient descent over the simplex (v_1, v_2, v_3)



Frank-Wolfe Phase

Once progress over simplex too small, call LP oracle to obtain new vertex and simplex

Main Theorem

You basically get what you expect.

Theorem. [Braun, P., Tu, Wright '18] Assume f is convex and smooth over the polytope P with curvature C and geometric strong convexity μ. Then Algorithm 1 ensures:

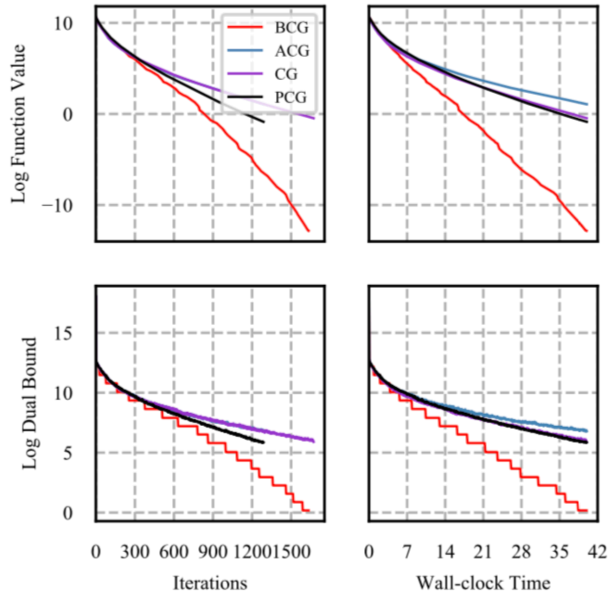
$$f(x_t) - f(x^*) \leq \varepsilon \qquad\qquad for\ t \geq \Omega\left(C/\mu \log \Phi_0/\varepsilon\right),$$

where x^* is an optimal solution to f over P and $\Phi_0 \geq f(x_0) - f(x^*)$.

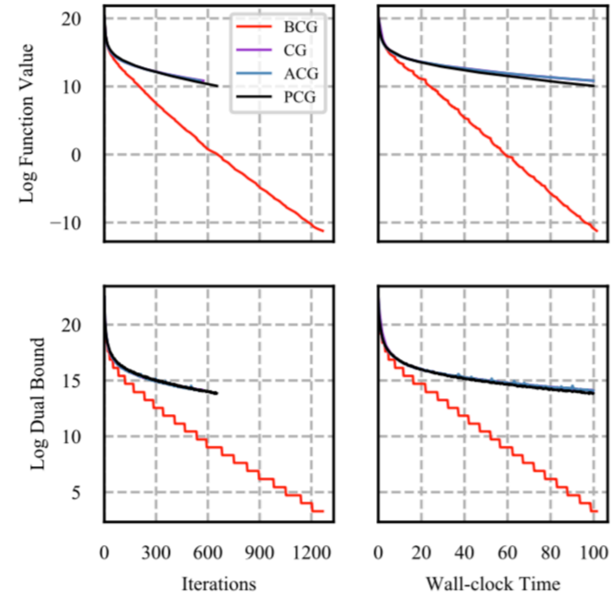
(For previous empirical work with similar idea see also [Rao, Shah, Wright '15])

Computational Results

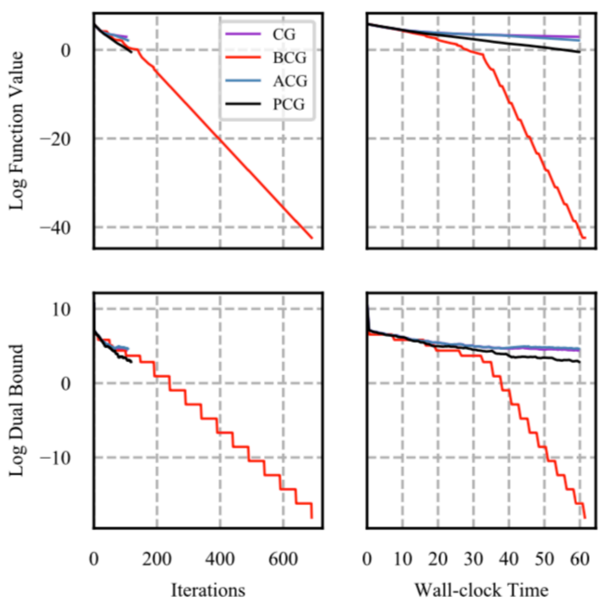
Lasso



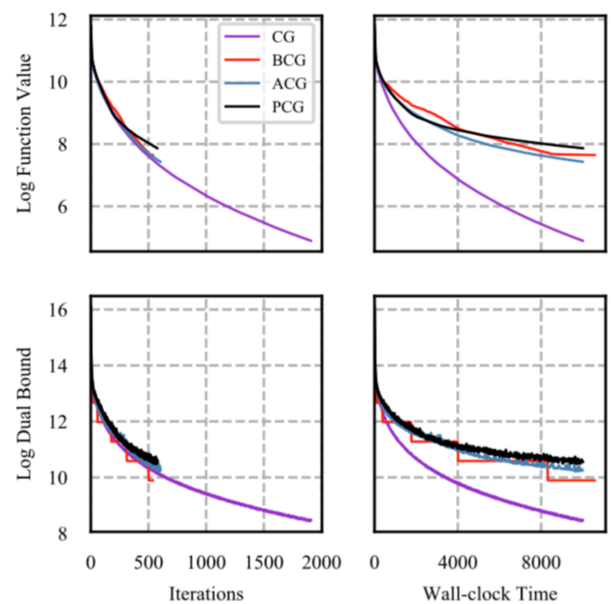
Structured Regression



Sparse Signal Recovery



Matrix Completion



ZUSE INSTITUTE BERLIN

TU berlin

Revisiting Conditional Gradients
Acceleration

How about acceleration?

The problem. Rates from standard proofs do not match known lower bounds:

1. Smooth convex case: $\mathcal{O}(1/\varepsilon)$ vs. $\Omega(1/\sqrt{\varepsilon})$
2. Smooth strongly convex case: $\mathcal{O}(\mu/L \log 1/\varepsilon)$ vs. $\Omega(\sqrt{\mu/L} \log 1/\varepsilon)$

Acceleration closes this gap. Various approaches:

1. Polyak's Heavy Ball method
2. Nemirovski Acceleration with Line Search
3. Nesterov Acceleration
4. ....

Limits to Acceleration for LP-based Methods

Lower bound. Consider the optimization problem:

[Jaeggi 2013, Lan 2013]

$$\min_{x \in \Delta(n)} \|x\|_2^2$$

where $\Delta(n) = \{x \in \mathbb{R}_+^n \mid \sum x_i = 1\}$ probability simplex.

Now, after k iterations the primal gap h_k is lower bounded as follows:

$$\boldsymbol{h_k \geq 1/k - 1/n}$$

1. Smooth convex: After $n/2$ iterations $\boldsymbol{h_{n/2} \geq 1/n}$
   => Vanilla FW rate is optimal (up to constant factors)

2. Smooth strongly convex: If $h_t \leq h_0 (1-r)^t$, then $\boldsymbol{r \leq 2 \log n / n}$
   => Away-Step FW rate of $(1 - 1/8n)$ is optimal (up to log factors)

Acceleration Beyond the Dimension Threshold?

Basic idea: The lower bound limits acceleration only up to the dimension. However, if we seek an *accelerated global rate* of the form:

$$h_t \leq h_0 (1-r)^t \text{, then } r \leq 2 \log n / n \text{ ,}$$

i.e., the lower bound also limits rates *beyond* the dimension threshold.

In a nutshell: We can design an algorithm that runs a **constant number** $\boldsymbol{T_0}$ of unaccelerated steps and then has "true" acceleration kick in.

=> Asymptotically optimal rate. Roughly: [Carderera, Diakonikolas, P. '19]

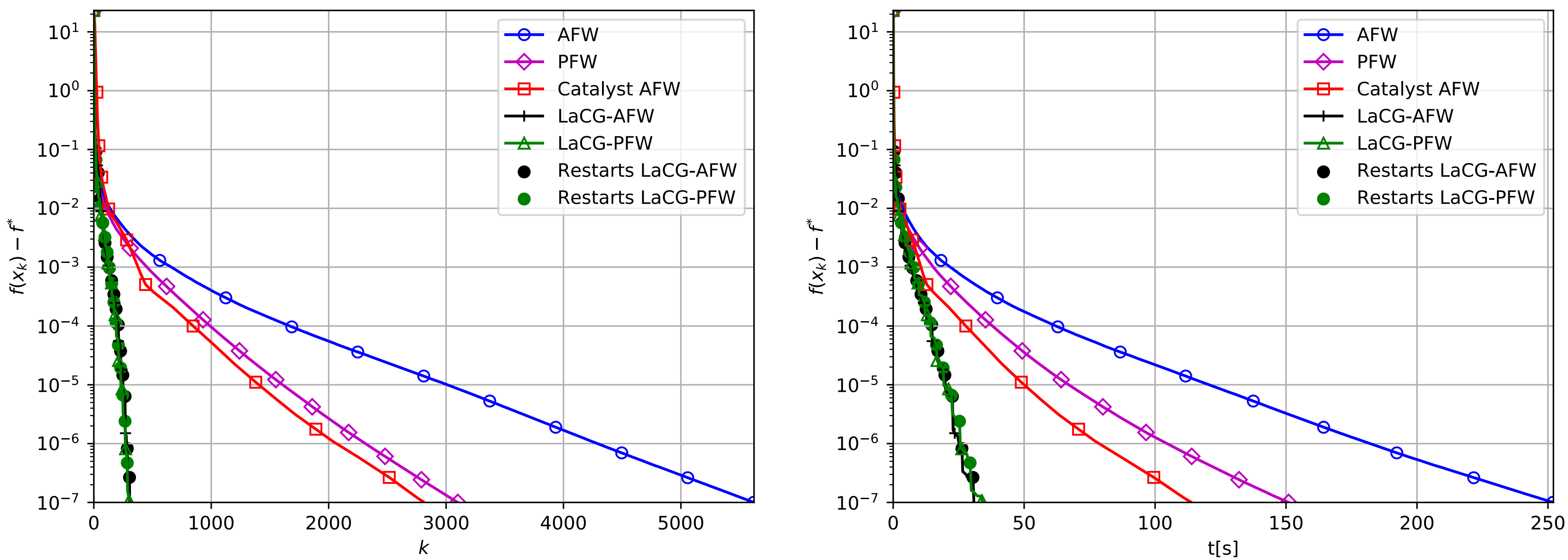
$$\boldsymbol{h_t \leq h_0 \left(1-\sqrt{\mu/L}\right)^{t-T_0}}$$

ZUSE INSTITUTE BERLIN

TU berlin

Preliminary Computational Results

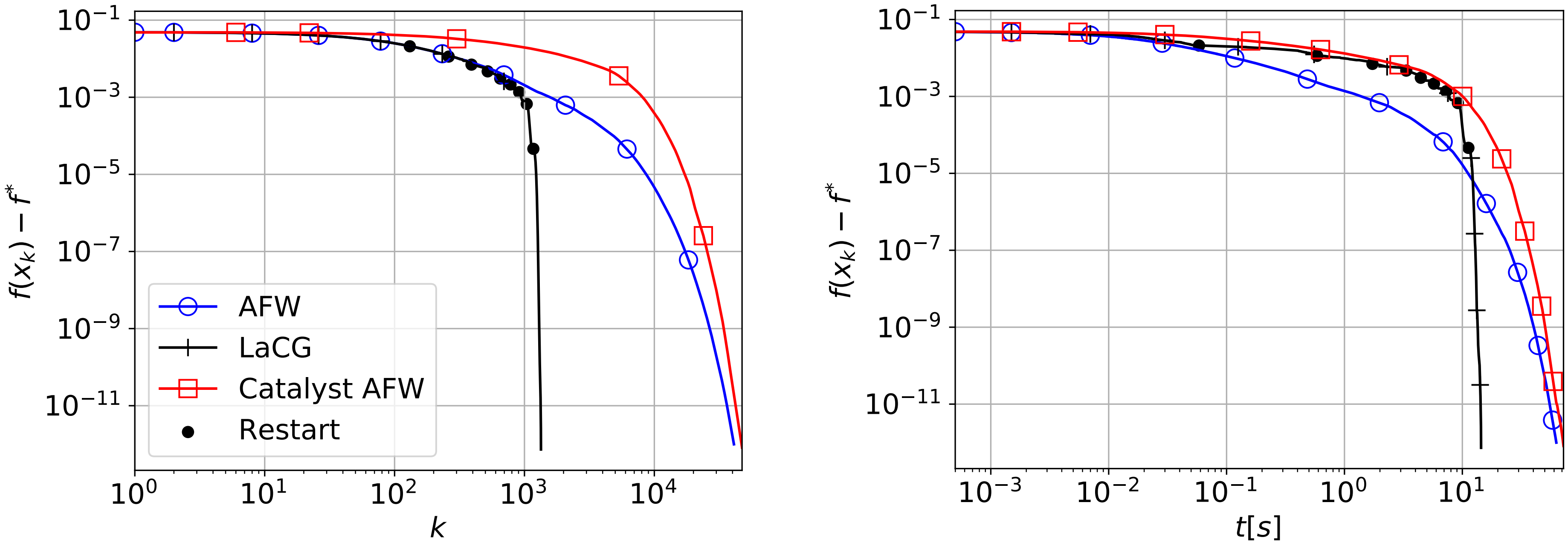
Setup: Quadratic over Birkhoff Polytope
=> small dim-dependent term



ZUSE INSTITUTE BERLIN

Preliminary Computational Results

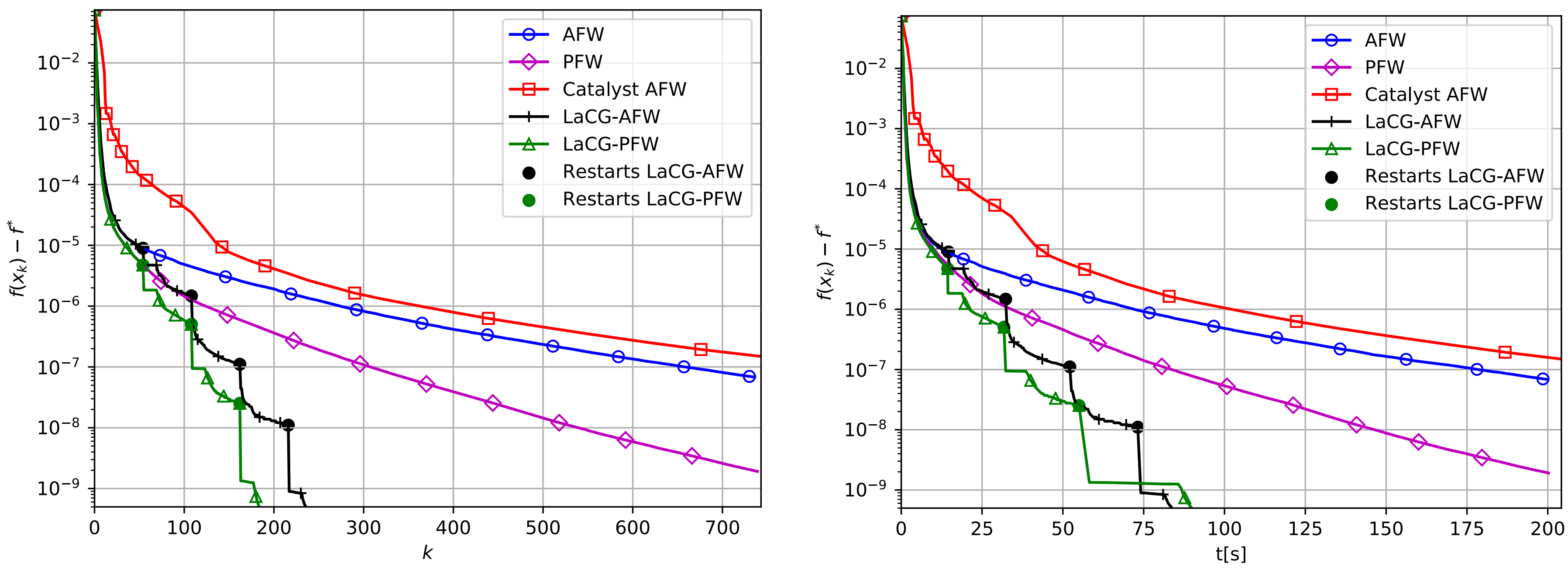
Setup: Quadratic over Probability Simplex (dim = 1000)
=> large dim-dependent term / lower bound instance



log-log scale

Preliminary Computational Results

Setup: Video Co-Localization



Want to know more?

Upcoming survey online in the next few weeks:

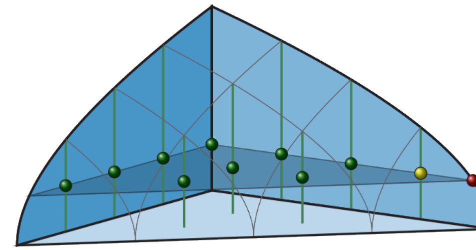
[Carderera, Combettes, P. “Conditional Gradients” ’19+]

Announcement: Combinatorial Optimization at Work

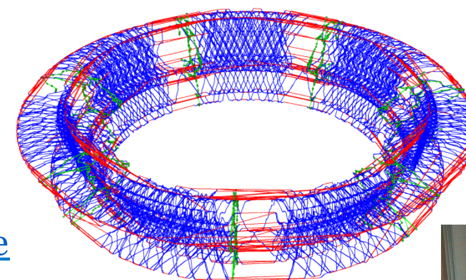
A summer school of TU Berlin in cooperation with MATH+ and the Berlin Mathematical School

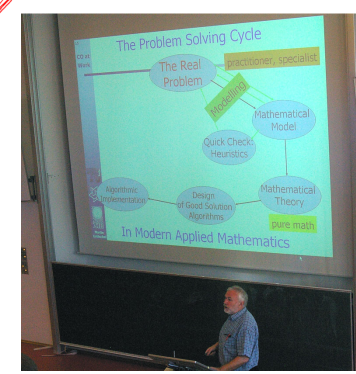
Everything you always wanted to know about LP/MIP and real-world industrial applications

(lectures and exercises)



- Dates of the course: September 14 – 26, 2020
- Language: English
- Location: Zuse Institute Berlin
- Application deadline: June 14, 2020
- Participation fee: none
- URL (info/application): http://co-at-work.zib.de
- Intended audience: master/PhD students, Post-docs
- Contact: coaw@zib.de
- Lectures by: the SCIP team, developers of Xpress, Gurobi, Gams, and many more





ZUSE INSTITUTE BERLIN

Smooth Constraint Convex Minimization via Conditional Gradients

Sebastian Pokutta

Zuse Institute Berlin
Technische Universität Berlin

Nice, 09/2019

twitter: @spokutta