

# A Primal-Dual Parallel Method with $O(1/t)$ Convergence for Constrained Composite Convex Programs

Appeared in: **CDC 2016, ArXiv 2017**

Min:

$F(x)$

Subj to:

$G_k(x) \leq 0, \quad k \in \{1, \dots, m\}$

$x \in \mathcal{X}$

Hao Yu (Amazon group)

Michael J. Neely (USC)

# Convex program

Min:  $F(x)$

Subj to:  $G_k(x) \leq 0, k \text{ in } \{1, \dots, m\}$   
 $x \in X$

## “Composite” type

$$F(x) = f(x) + \tilde{f}(x)$$

$$G_k(x) = g_k(x) + \tilde{g}_k(x)$$

$f(x)$  = convex, smooth

$\tilde{f}(x)$  = convex, nonsmooth,  
**(hopefully)** separable

# Example LASSO Problem

$$\text{Min: } \|Ax - b\|^2 + \lambda \|x\|_1$$

$$\text{S.t.: } Cx \leq d$$

$$x \in \mathcal{X}$$

- Machine learning  
[James et al 2012]
- Financial portfolio optimization  
[Brodie et al]

# Algorithm Types

## **Interior point or Newton Type**

- Usually need smooth problems
- Small number of iterations.
- Each iteration is complex:  
[Hessians, matrix inversions]

## **Lagrangian Dual or Primal-Dual**

- Simpler iterations, but many of them

# Algorithm Types

## **Interior point or Newton Type**

- Usually need smooth problems
- Small number of iterations.
- Each iteration is complex:  
[Hessians, matrix inversions]

## **Lagrangian Dual or Primal-Dual**

- Simpler iterations, but many of them

# Prior Lagrangian Dual Algs

## Dual subgradient (smooth/non-smooth)

$$\begin{aligned} \text{Min : } & F(\mathbf{x}[t]) + \sum Q_k[t] G_k(\mathbf{x}[t]) \\ \text{S.t. : } & \mathbf{x}[t] \text{ in } \mathcal{X} \end{aligned}$$

- $1/\varepsilon^2$  iterations (slow)
- Each iteration is simple when  $F, G$  *separable*.

# Prior Lagrangian Dual Algs

**Yu-Neely 2017 (smooth/non-smooth)**

- $1/\epsilon$  iterations (faster)
- Each iteration is simple when  $F, G$  **separable**.

# Prior Primal-Dual Algs

## Arrow-Hurwicz-Uzawa (smooth/non-smooth)

- Use gradient or subgradient
- Each iteration is simple:  
*Using gradient turns nonseparable problem into a separable problem.*
- $1/\varepsilon^2$  iterations (slow)
- Needs to know bounds on Lagrange multipliers

# Prior (non-smooth) Composite

**Mirror descent on dual  $\rightarrow 1/\varepsilon^2$**

[Beck et al 2010]

**Single smooth constraint  $\rightarrow 1/\varepsilon$**

[Shefi-Teboulle 2016]

**Linear constraints  $\rightarrow 1/\varepsilon$  (random)**

[Gau-Xu-Zhang 2016]

# Proposed Algorithm

## **Accelerated Primal-Dual (Yu-Neely 20xx)**

- $1/\varepsilon$  convergence for general problems
- Simple iterations
- Can handle composite (non-smooth) structure
- Do not need Lagrange multiplier bounds
- Extends our prior accelerated dual alg.

# Recall

$$F(x) = f(x) + \tilde{f}(x)$$

$$G_k(x) = g_k(x) + \tilde{g}_k(x)$$

$f(x)$  = convex, smooth

$\tilde{f}(x)$  = convex, nonsmooth,  
**(hopefully)** separable

# Algorithm: Every slot $t$ in $\{0,1,2\dots\}$

**1. Choose  $x[t]$  in  $X$  to minimize:**

$$f'(x[t-1]) \bullet x[t] + \tilde{f}(x[t])$$

$$+ \sum W_k[t] [ g'_k(x[t-1]) \bullet x[t] + \tilde{g}_k(x[t]) ]$$

$$+ \alpha[t] \| x[t] - x[t-1] \|^2$$

$$W_k[t] = Q_k[t] + G_k(x(t-1))$$

$\alpha[t]$  = Dynamic stepsize parameter

# Algorithm: Every slot t in {0,1,2...}

## 2. Virtual Queue Update:

$$Q_k[t+1] = \max[ Q_k[t] + G_k(x[t]), -G_k(x[t-1]) ]$$

# Algorithm: Every slot t in {0,1,2...}

## 2. Virtual Queue Update:

$$Q_k[t+1] = \max[ Q_k[t] + G_k(x[t]), -G_k(x[t-1]) ]$$

Compare with standard LM Update:

$$Q_k[t+1] = \max[Q_k[t] + G_k(x[t]), 0]$$

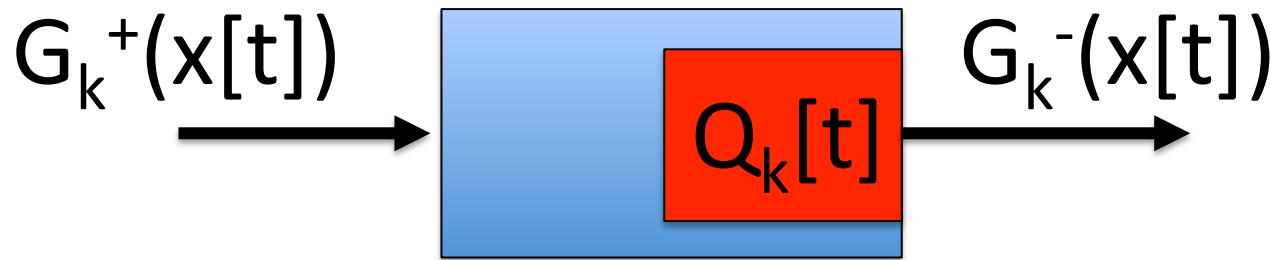
# Algorithm: Every slot t in {0,1,2...}

## 2. Virtual Queue Update:

$$Q_k[t+1] = \max[ Q_k[t] + G_k(x[t]), -G_k(x[t-1]) ]$$

Compare with standard LM Update:

$$Q_k[t+1] = \max[Q_k[t] + G_k(x[t]), 0]$$



**Algorithm: Every slot t in {0,1,2...}**

### **3. Time Average the Results:**

$$\bar{x}[T] = (1/T) \sum_{t=0}^{T-1} x[t]$$

# Algorithm: Every slot t in {0,1,2...}

## 3. Time Average the Results:

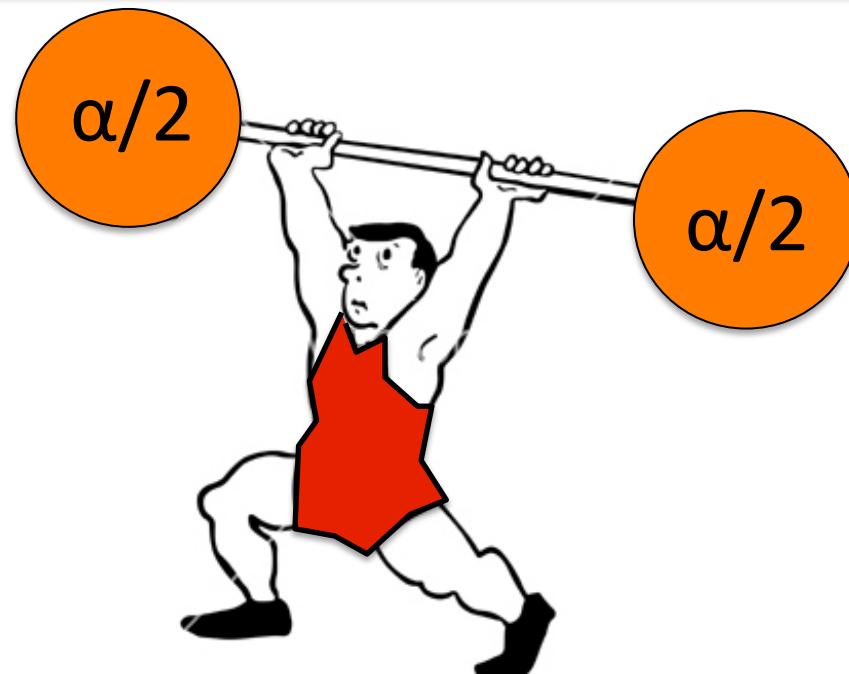
$$\bar{x}[T] = (1/T) \sum_{t=0}^{T-1} x[t]$$

Analysis: Create ***strong convexity*** where there is none to begin with...

# Strongly convex functions are mighty!

Def:  $f(x)$  is *strongly convex with modulus  $\alpha$*   
if the following is convex:

$$f(x) - (\alpha/2)\|x\|^2$$



# Construction Property

Suppose  $f(x)$  is convex.

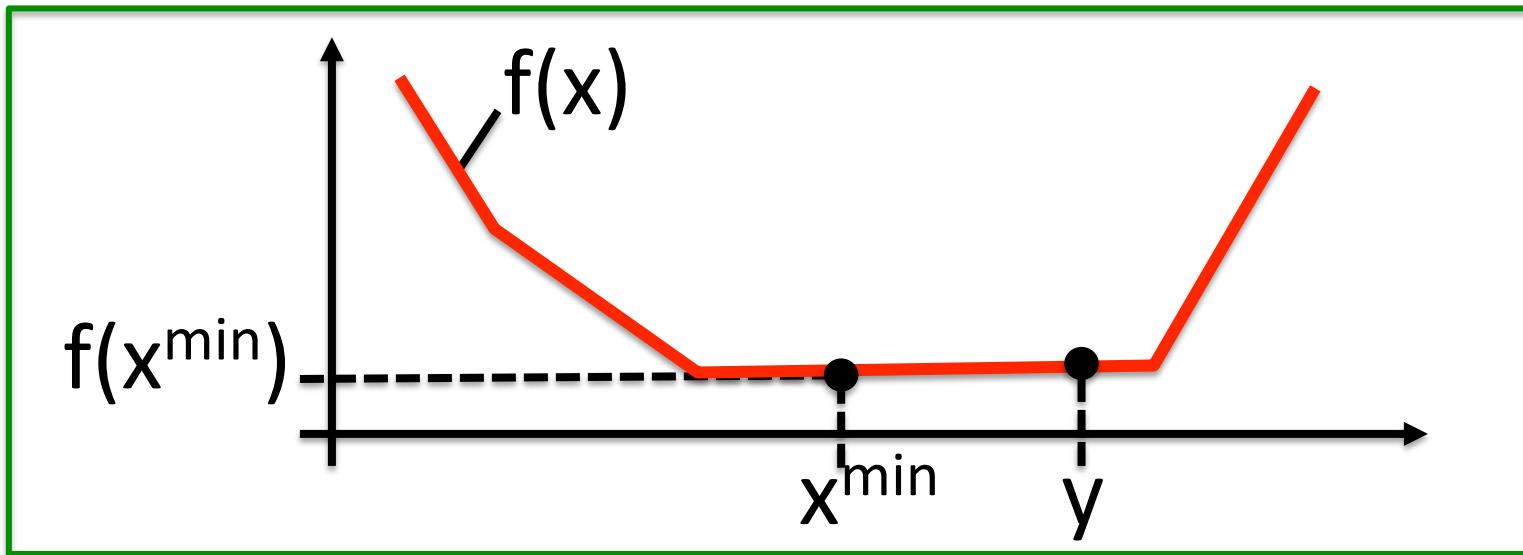
For any fixed vector  $v$ , build  $h(x)$ :

$$h(x) = f(x) + \underbrace{(\alpha/2)\|x-v\|^2}_{\text{new part}}$$



$h(x)$  is strongly convex with modulus  $\alpha$ .

## Minimization of *General* $f(x)$



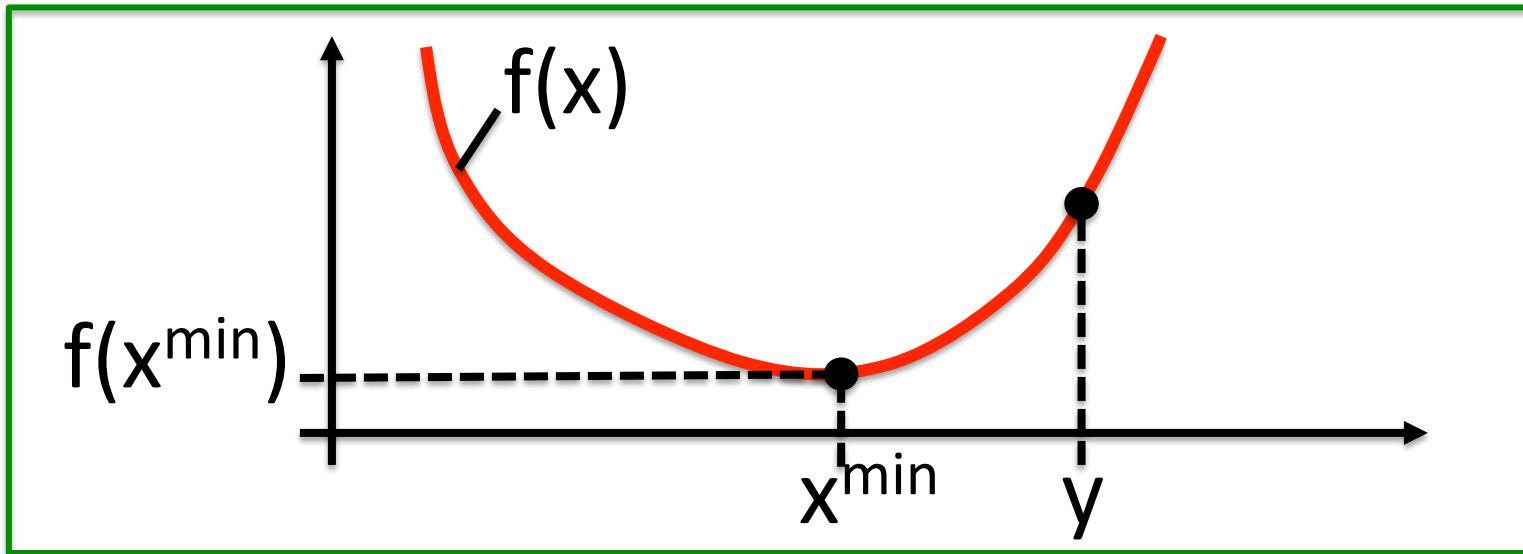
Suppose  $x^{\min}$  minimizes  $f(x)$ .

Then:

$$f(x^{\min}) \leq f(y)$$

for all vectors  $y$ .

## Minimization of *Strongly Convex* $f(x)$

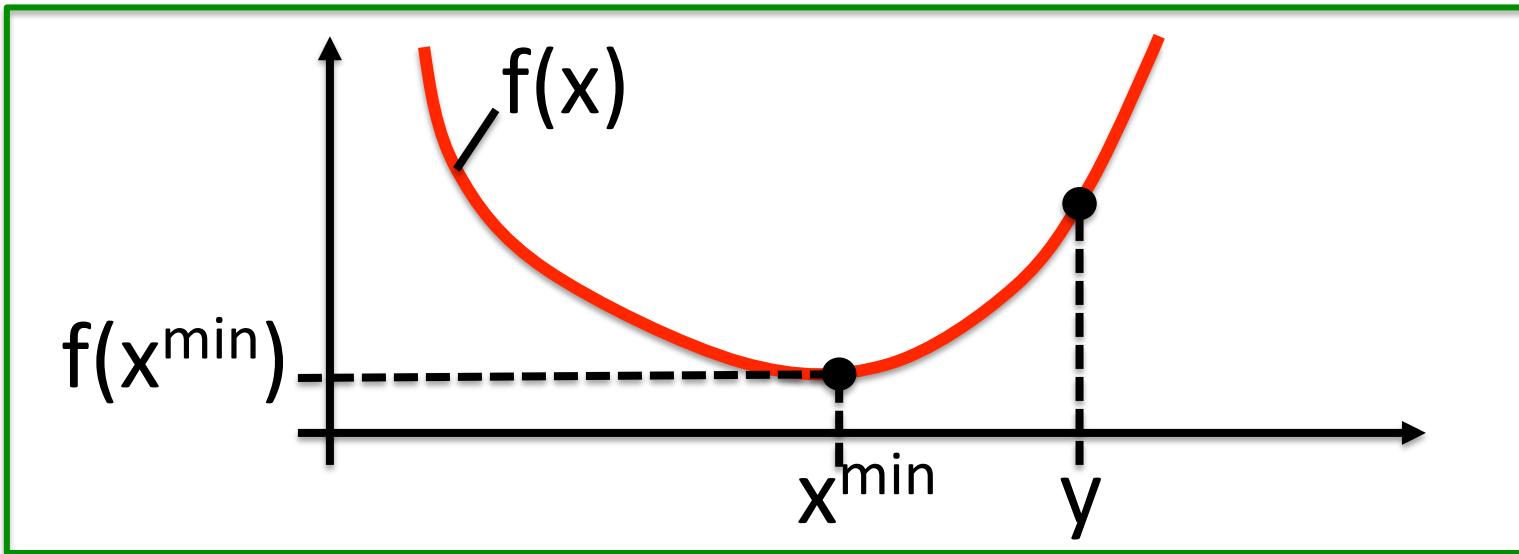


Suppose  $x^{\min}$  minimizes the *strongly convex*  $f(x)$ .  
Then:

$$f(x^{\min}) \leq f(y) - (\alpha/2) \|x^{\min} - y\|^2$$

for all vectors  $y$ .

## Minimization of *Strongly Convex* $f(x)$



Suppose  $x^{\min}$  minimizes the *strongly convex*  $f(x)$ .  
Then:

$$f(x^{\min}) \leq f(y) - (\alpha/2) \|x^{\min} - y\|^2$$

for all vectors  $y$ .

“Pushback” term

# Drift-Plus-Penalty (DPP) Analysis

$$L[t] = (1/2) \|Q[t]\|^2$$

$$\Delta[t] = L[t+1] - L[t]$$

$$\begin{aligned} DPP[t] \\ = \Delta[t] + F(x[t]) + \alpha \|x[t]-x[t-1]\|^2 \end{aligned}$$

# Drift-Plus-Penalty Analysis

$$\begin{aligned} DPP[t] = & B[t] + f'(x[t-1]) \color{red}{x[t]} + \tilde{f}(x[t]) \\ & + \sum W_k[t] [ g_k'(x[t-1]) \color{red}{x[t]} + \tilde{g}_k(x[t]) ] \\ & + \alpha \| \color{red}{x[t]} - x[t-1] \|^2 \end{aligned}$$

# Drift-Plus-Penalty Analysis

$$\begin{aligned} DPP[t] &= B[t] + f'(x[t-1]) \color{red}{x[t]} + \tilde{f}(x[t]) \\ &\quad + \sum W_k[t] [ g_k'(x[t-1]) \color{red}{x[t]} + \tilde{g}_k(x[t]) ] \\ &\quad + \alpha \| \color{red}{x[t]} - x[t-1] \|^2 \\ &\leq B[t] + f'(x[t-1]) \color{red}{x^*} + \tilde{f}(x^*) \\ &\quad + \sum W_k[t] [ g_k'(x[t-1]) \color{red}{x^*} + \tilde{g}_k(x^*) ] \\ &\quad + \alpha \| \color{red}{x^*} - x[t-1] \|^2 \end{aligned}$$

# Drift-Plus-Penalty Analysis

$$\begin{aligned} \text{DPP}[t] &= B[t] + f'(x[t-1])\mathbf{x}[t] + \tilde{f}(x[t]) \\ &\quad + \sum W_k[t] [ g_k'(x[t-1])\mathbf{x}[t] + \tilde{g}_k(x[t]) ] \\ &\quad + \alpha \| \mathbf{x}[t] - x[t-1] \|^2 \\ &\leq B[t] + f'(x[t-1])\mathbf{x}^* + \tilde{f}(x^*) \\ &\quad + \sum W_k[t] [ g_k'(x[t-1])\mathbf{x}^* + \tilde{g}_k(x^*) ] \\ &\quad + \alpha \| \mathbf{x}^* - x[t-1] \|^2 - \underbrace{\alpha \| \mathbf{x}^* - x[t-1] \|^2}_{\text{Pushback!}} \end{aligned}$$

# Drift-Plus-Penalty Analysis

$$\begin{aligned} \text{DPP}[t] &= B[t] + f'(x[t-1])\mathbf{x}[t] + \tilde{f}(x[t]) \\ &\quad + \sum W_k[t] [ g_k'(x[t-1])\mathbf{x}[t] + \tilde{g}_k(x[t]) ] \\ &\quad + \alpha \| \mathbf{x}[t] - x[t-1] \|^2 \\ &\leq B[t] + f'(x[t-1])\mathbf{x}^* + \tilde{f}(x^*) \\ &\quad + \sum W_k[t] [ g_k'(x[t-1])\mathbf{x}^* + \tilde{g}_k(x^*) ] \\ &\quad + \alpha \| \mathbf{x}^* - x[t-1] \|^2 - \alpha \| \mathbf{x}^* - x[t-1] \|^2 \end{aligned}$$

Telescopes when summed!

# Theorem

- IF:
1.  $F(x)$ ,  $G_1(x), \dots, G_m(x)$  Convex
  2.  $G_k(x)$  Lipschitz continuous for all  $k$
  3. Problem is feasible and has L.M.  $\mu$

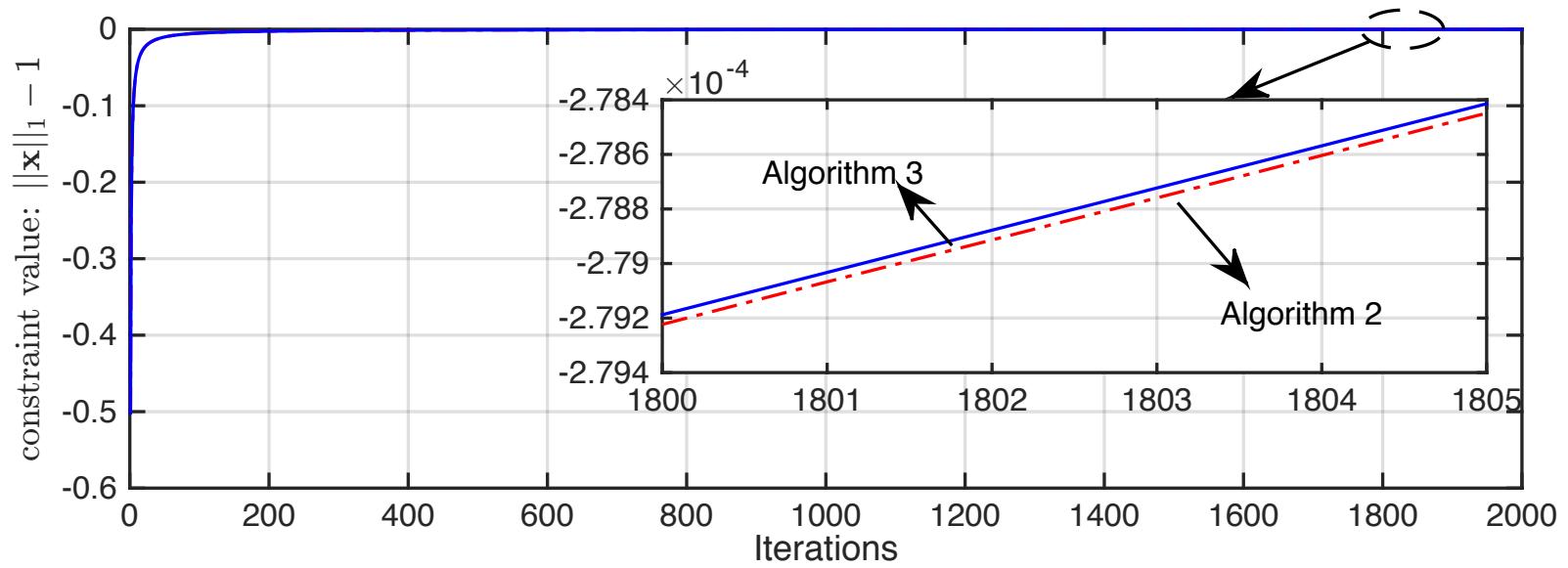
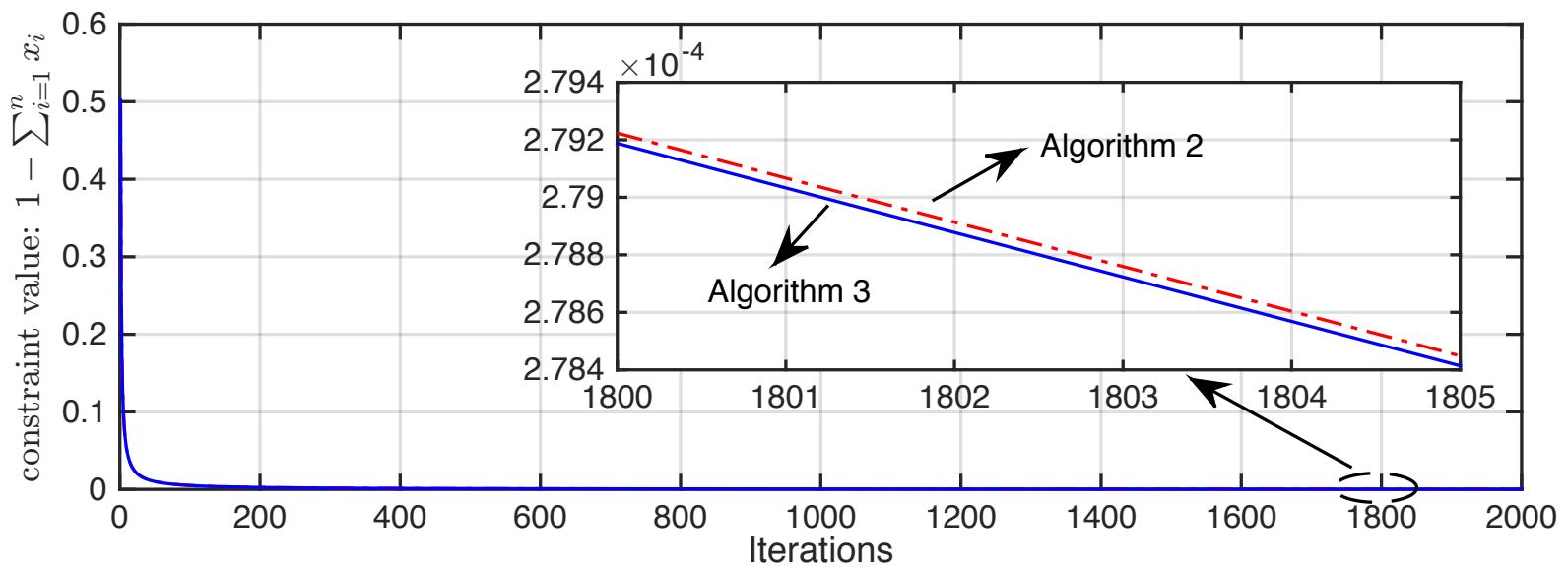
THEN FOR ALL  $T$  in  $\{1, 2, 3, \dots\}$ :

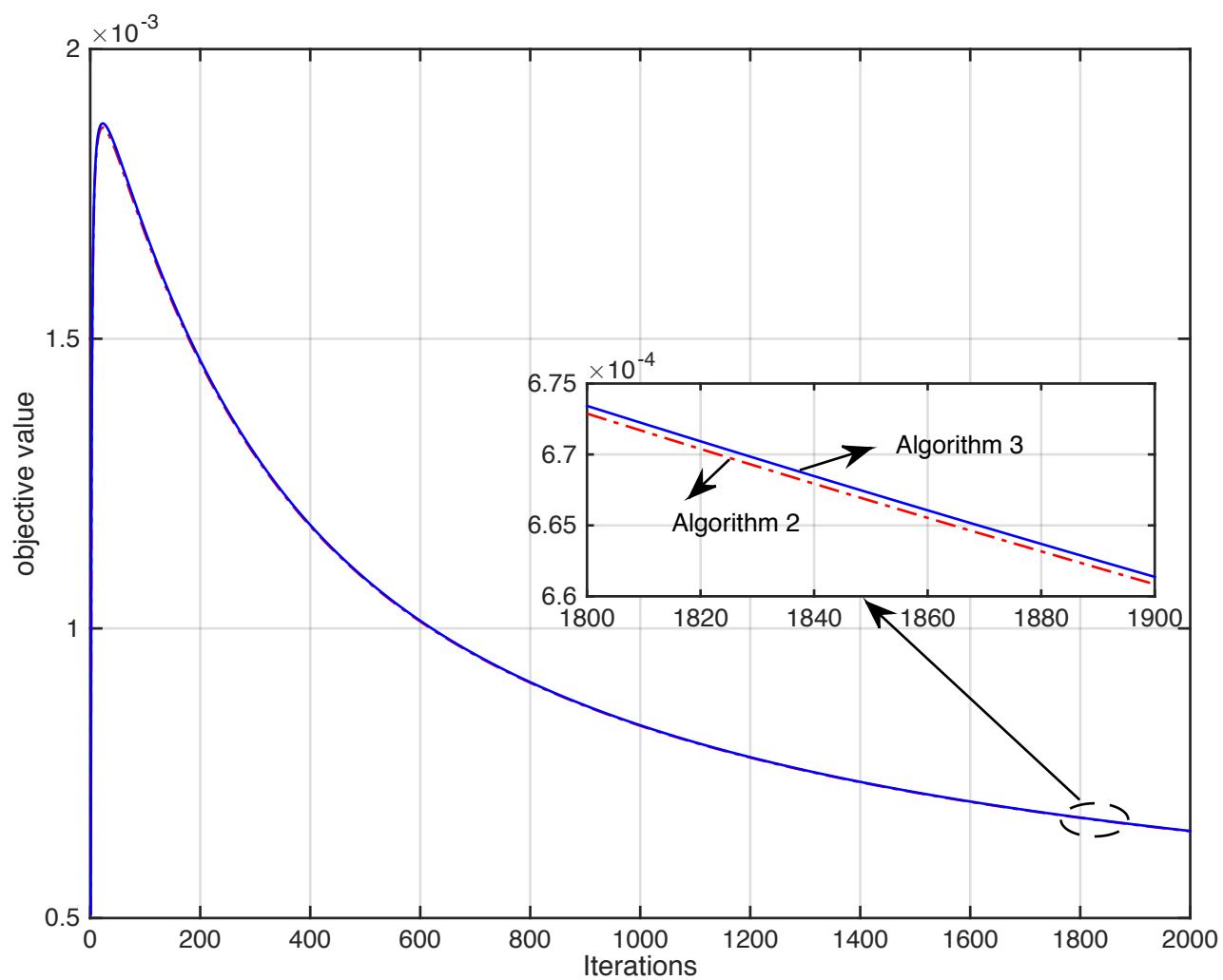
$$F(\bar{x}(T)) \leq F(x^*) + O(1/T)$$

$$G_k(\bar{x}(T)) \leq O(1/T) \quad \text{for all } k \in \{1, \dots, m\}$$

# Simulation [Compare Dual and Primal-Dual]

- Minimum Variance Portfolio optimization with (nonsmooth) L1 Norm constraint
- 500 variables
- **RESULTS:** Similar solution quality after each iteration, but each primal-dual iteration is ***1800 times faster***





# Conclusion

- Generalize accelerated **dual** Lagrangian alg to **primal-dual**
- Faster to implement (**1800x gain**)
- 3 crucial techniques:
  - ✓ New virtual queue update
  - ✓ New DPP weights
  - ✓ Use prox term effectively

# Paper available on

## 1. Dynamic $\alpha[t]$ (no knowledge of LM bound):

H. Yu and M. J. Neely, “A Primal-Dual Parallel Method with  $O(1/\varepsilon)$  Convergence for Constrained Composite Convex Programs,” Arxiv technical report arXiv:1708.00322 July 2017.

<https://arxiv.org/abs/1708.00322>

## 2. Fixed $\alpha[t]$ (need to know LM bound):

H. Yu and M. J. Neely, A primal-dual type algorithm with the  $O(1=t)$  convergence rate for large scale constrained convex programs, in Proceedings of IEEE Conference on Decision and Control (CDC), 2016.

## 3. Dual alg:

H. Yu and M. J. Neely, A simple parallel algorithm with an  $O(1=t)$  convergence rate for general convex programs, SIAM Journal on Optimization, 27 (2017), pp. 759-783.