Accelerated, Parallel and Proximal Coordinate Descent

Olivier Fercoq

Joint work with P. Richtárik

22 June 2015

17th Leslie Fox Prize competition

Minimisation of composite functions

Minimise the composite function F for $x \in \mathbb{R}^N$

$$\min_{x \in \mathbb{R}^N} \{F(x) = f(x) + \psi(x)\}$$

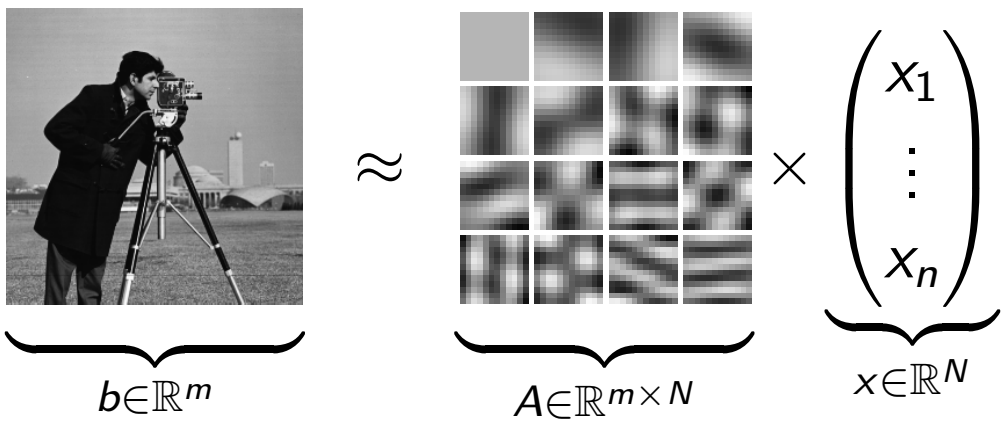
- $f : \mathbb{R}^N \to \mathbb{R}$, convex, differentiable, not strongly convex
- $\psi : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$, convex, separable

$$\psi(x) = \sum_{i=1}^{n} \psi_i(x^{(i)})$$

Example: L_1-regularised least squares (Lasso)

$$\min_{x \in \mathbb{R}^N} \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1$$

Determines the parameters x of the model

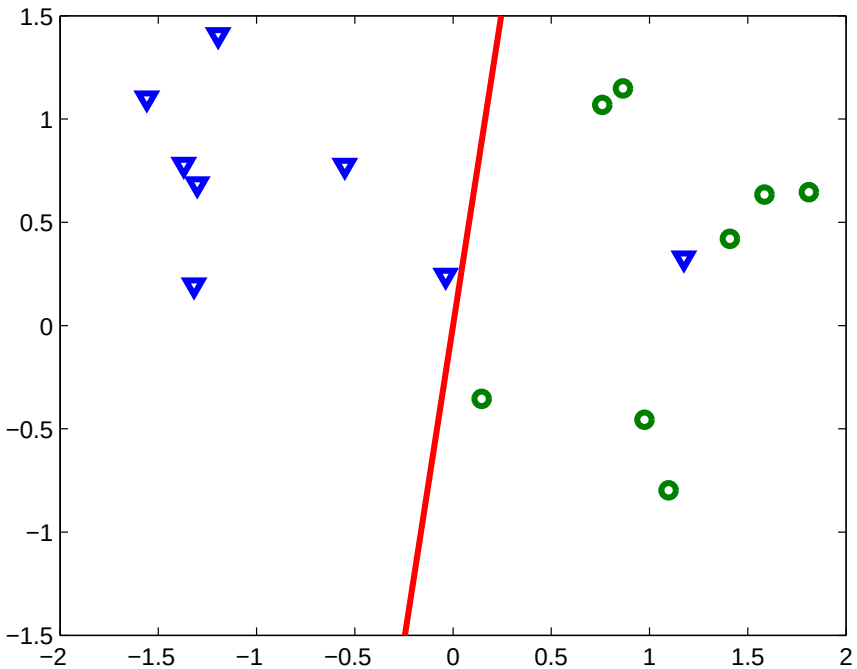


where we would like x sparse.

$\psi = \lambda\|\cdot\|_1$, f quadratic.

Example: Dual of Support Vector Machines

$$\min_{x\in[0,1]^N} \frac{1}{2\lambda N^2}\sum_{j=1}^{m}\left(\sum_{i=1}^{N} b_i A_{ji} x^{(i)}\right)^2 - \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$$



f quadratic

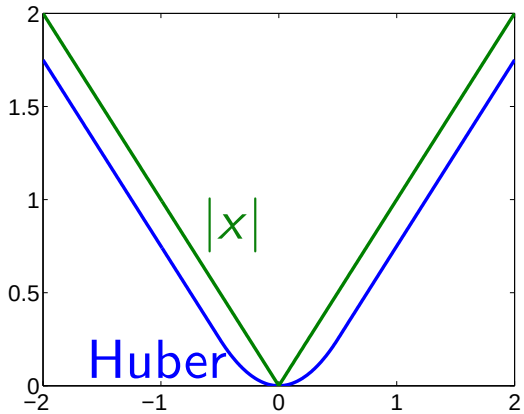
$\psi = I_{[0,1]^N}$

Example: L_1-regularised L_1 regression

$$\min_{x\in\mathbb{R}^N} \sum_{j=1}^{m} |e_j^T Ax - b_j| + \lambda\|x\|_1$$

$$\xrightarrow{\text{smoothing}} \sum_{j=1}^{m} f_\mu^j(e_j^T Ax - b_j) + \lambda\|x\|_1$$

- $\psi = \lambda\|\cdot\|_1$
- f_μ^j is a Huber function
- Differentiable, convex but not strongly convex



Coordinate descent

f has Lipschitz continuous directional derivatives:

$$f(x + te_i) \leq f(x) + \langle \nabla f(x), te_i \rangle + \frac{L_i}{2} \|te_i\|^2$$

At each iteration:

1. Choose randomly a coordinate i
2. Compute the update $t \in \mathbb{R}$ that minimises the overapproximation of $f(x + te_i)$
3. Update the variable $x \leftarrow x + te_i$

Remarks:

- Many iterations are needed
- Each iteration is cheap
- Well fitted to sparse matrices in column format

Proximal coordinate descent

$F = f + \psi$:

$$F(x + te_i) \leq \psi(x + te_i) + f(x) + \langle \nabla f(x), te_i \rangle + \frac{L_i}{2} \|te_i\|^2$$

At each iteration:

1. Choose randomly a coordinate i
2. Compute the update $t \in \mathbb{R}$ that minimises the overapproximation of $F(x + te_i)$
3. Update the variable $x \leftarrow x + te_i$

Remarks:

- Essential to deal with non-smooth regularisers
- Involves the proximity operator of ψ_i: prox_{ψ_i}
  Ex: projection onto a box, soft thresholding
- Adding separable ψ : same speed of convergence

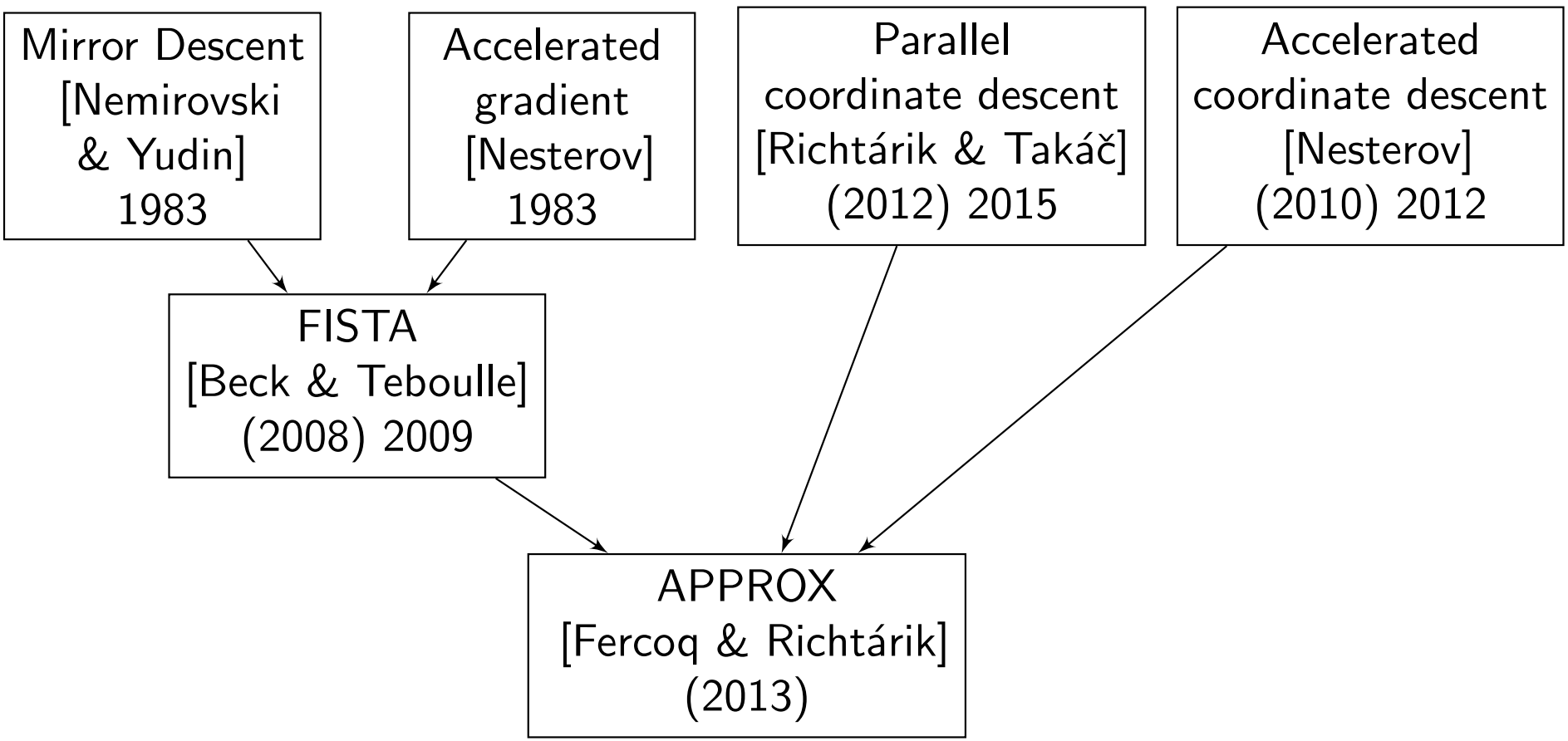
Accelerated gradient

k = iteration counter

Algorithm	**Convergence**	**Assumptions**
Sub-gradient	$\mathcal{O}(1/\sqrt{k})$	convex
Gradient	$\mathcal{O}(1/k)$	smooth, convex
Accelerated gradient	$\mathcal{O}(1/k^2)$	smooth, convex
Gradient	$\mathcal{O}((1-\kappa)^k)$	smooth, strongly convex
Accelerated gradient	$\mathcal{O}((1-\sqrt{\kappa})^k)$	smooth, strongly convex

Similar convergence rates for coordinate descent

Context


Mirror Descent
[Nemirovski
& Yudin]
1983
Accelerated
gradient
[Nesterov]
1983
Parallel
coordinate descent
[Richtárik & Takáč]
(2012) 2015
Accelerated
coordinate descent
[Nesterov]
(2010) 2012
FISTA
[Beck & Teboulle]
(2008) 2009
APPROX
[Fercoq & Richtárik]
(2013)

APPROX: **A**ccelerated, **P**arallel and **PROX**imal coordinate descent

$x_0 \in \operatorname{dom} \psi$, $z_0 = x_0$, $\theta_0 = \frac{\tau}{n}$ and $\tau = \mathbf{E}[|\hat{S}|]$

for $k \geq 0$ **do**

$y_k = (1 - \theta_k)x_k + \theta_k z_k$

Generate a random set of coordinates $S_{k+1} \sim \hat{S}$

$z_{k+1} \leftarrow z_k$

for $i \in S_{k+1}$ **do**

$$z_{k+1}^{(i)} = \operatorname*{argmin}_{z \in \mathbb{R}^{N_i}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

end for

$x_{k+1} = y_k + \frac{n}{\tau}\theta_k(z_{k+1} - z_k)$

$\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$

end for

Accelerated, Parallel and Proximal coordinate descent

$x_0 \in \operatorname{dom} \psi$, $z_0 = x_0$, $\theta_0 = \frac{\tau}{n}$ and $\tau = \mathbf{E}[|\hat{S}|]$

for $k \geq 0$ **do**

$y_k = (1 - \theta_k)x_k + \theta_k z_k$

Generate a random set of coordinates $S_{k+1} \sim \hat{S}$

$z_{k+1} \leftarrow z_k$

for $i \in S_{k+1}$ **do**

$$z_{k+1}^{(i)} = \operatorname*{argmin}_{z \in \mathbb{R}^{N_i}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

end for

$x_{k+1} = y_k + \frac{n}{\tau}\theta_k (z_{k+1} - z_k)$

$\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$

end for

Accelerated, Parallel and Proximal coordinate descent

$x_0 \in \operatorname{dom} \psi$, $z_0 = x_0$, $\theta_0 = \frac{\tau}{n}$ and $\tau = \mathbf{E}[|\hat{S}|]$

for $k \geq 0$ **do**

$y_k = (1 - \theta_k)x_k + \theta_k z_k$

Generate a random set of coordinates $S_{k+1} \sim \hat{S}$

$z_{k+1} \leftarrow z_k$

for $i \in S_{k+1}$ **do**

$$z_{k+1}^{(i)} = \operatorname*{argmin}_{z \in \mathbb{R}^{N_i}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

end for

$x_{k+1} = y_k + \frac{n}{\tau}\theta_k (z_{k+1} - z_k)$

$\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$

end for

Accelerated, Parallel and Proximal coordinate descent

$x_0 \in \operatorname{dom} \psi$, $z_0 = x_0$, $\theta_0 = \frac{\tau}{n}$ and $\tau = \mathbf{E}[|\hat{S}|]$

for $k \geq 0$ **do**

$y_k = (1 - \theta_k)x_k + \theta_k z_k$

Generate a random set of coordinates $S_{k+1} \sim \hat{S}$

$z_{k+1} \leftarrow z_k$

for $i \in S_{k+1}$ **do**

$$z_{k+1}^{(i)} = \operatorname*{argmin}_{z \in \mathbb{R}^{N_i}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

end for

$x_{k+1} = y_k + \frac{n}{\tau}\theta_k (z_{k+1} - z_k)$

$\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$

end for

Efficient implementation

Idea of [Lee & Sidford, 2013]

for $k \geq 0$ **do**

Generate a random set of coordinates $S_{k+1} \sim \hat{S}$

for $i \in S_{k+1}$ **do**

$$t_k^{(i)} = \arg\min_{t \in \mathbb{R}^{N_i}} \langle \nabla_i f(\theta_k^2 u_k + \tilde{z}_k), t \rangle + \frac{n\theta_k \beta L_i}{2\tau} \|t\|_{(i)}^2 + \psi_i(\tilde{z}_k^{(i)} + t)$$

$$\tilde{z}_{k+1}^{(i)} = \tilde{z}_k^{(i)} + t_k^{(i)}$$

$$u_{k+1}^{(i)} = u_k^{(i)} - \frac{1 - \frac{n}{\tau}\theta_k}{\theta_k^2} t_k^{(i)}$$

end for

end for

Can one compute $\nabla_i f\left(\theta_k^2 u_k + \tilde{z}_k\right)$ easily ?

- A special kind of partially separable functions:

$$f(x) = \sum_{j=1}^{m} \phi_j(e_j^T A x)$$

$$\omega_j = |C_j| = |\{i \ : \ A_{j,i} \neq 0\}| \ , \quad D_i = \{j \ : \ A_{j,i} \neq 0\}$$

- We stock and update $r_{u_k} = A u_k$ et $r_{\tilde{z}_k} = A\tilde{z}_k$:

$$\nabla_i f(\theta_k^2 u_k + \tilde{z}_k) = \sum_{j \in D_i} A_{ji}^T \phi_j'(\theta_k^2 r_{u_k}^j + r_{\tilde{z}_k}^j)$$

- Average cost to compute the τ partial derivatives:

$$\mathcal{C} = \mathbf{E}\Big[\sum_{i \in \hat{S}} \mathcal{O}(|D_i|)\Big] = \frac{\tau}{n} \sum_{i=1}^{n} \mathcal{O}(|D_i|)$$

Iteration complexity

Theorem

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2}\left(F(x_0) - F(x_*) + \frac{\beta}{2}\|x_0 - x_*\|_L^2\right)$$

Iteration complexity

Theorem

randomised algorithm

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2}\left(F(x_0) - F(x_*) + \frac{\beta}{2}\|x_0 - x_*\|_L^2\right)$$

Iteration complexity

Theorem

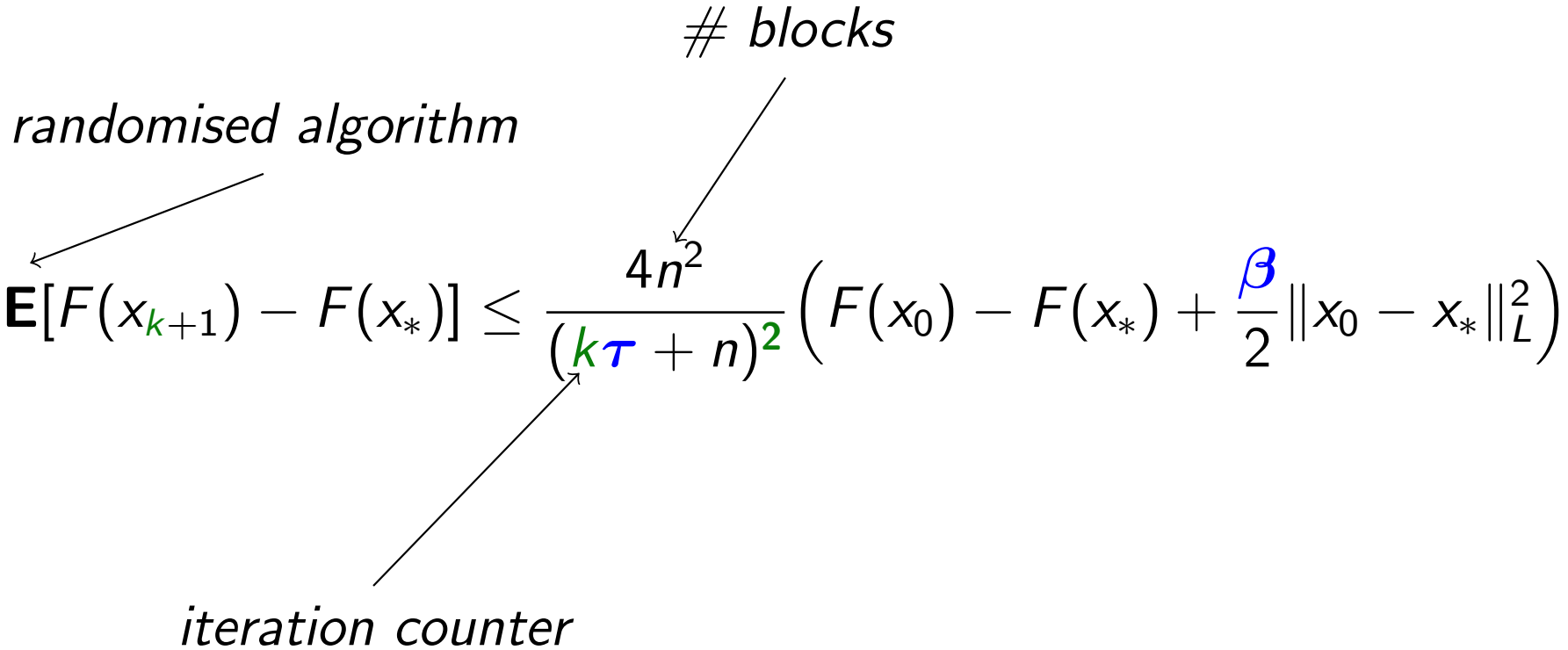
 # blocks

randomised algorithm

$$\mathbf{E}[F(x_{k+1}) - F(x_*)] \leq \frac{4n^2}{(k\tau + n)^2}\left(F(x_0) - F(x_*) + \frac{\beta}{2}\|x_0 - x_*\|_L^2\right)$$

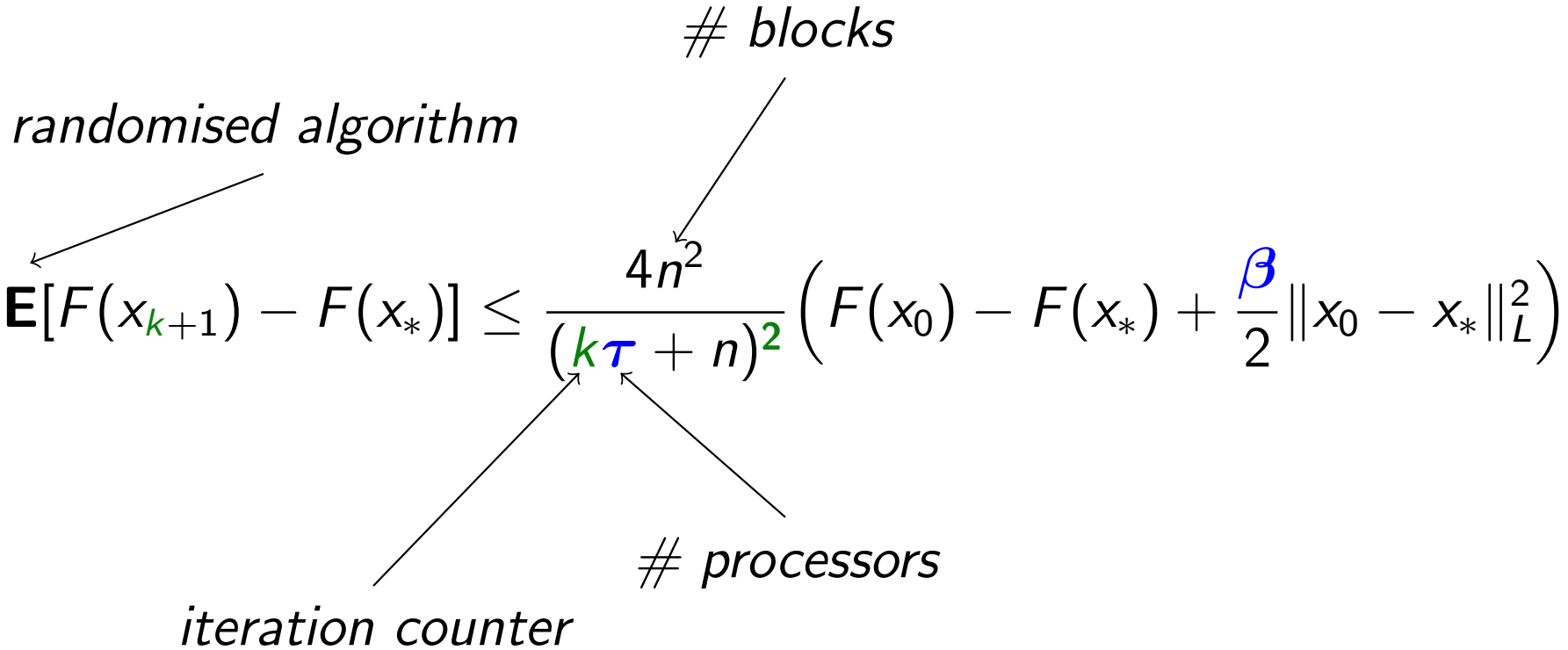
Iteration complexity

Theorem



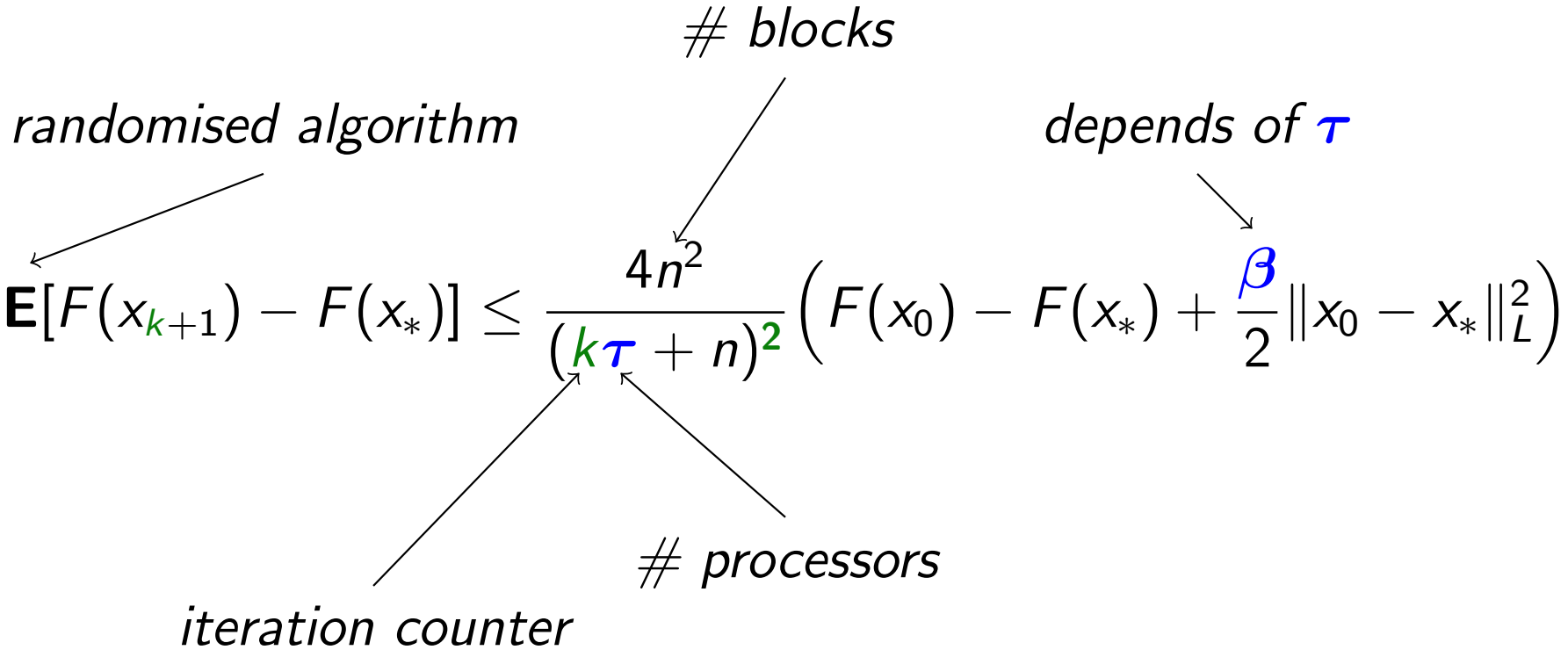
Iteration complexity

Theorem



Iteration complexity

Theorem



Parallel coordinate descent

- We need more general overapproximations to compute the updates:
  - $f(x + te_i) \leq f(x) + \langle \nabla f(x), te_i \rangle + \frac{L_i}{2} \|te_i\|^2$
    $\hookrightarrow$ not valid with several coordinates
  - $f(x + h) \leq f(x) + \langle \nabla f(x), h \rangle + \frac{L(\nabla f)}{2} \|h\|^2$
    $\hookrightarrow$ too weak
- [Richtárik, Takáč, 2012]
  Theory for partially separable functions

$$f(x) = \sum_{j=1}^{m} f_j(x)$$

f_j depends on $i \in C_j$ only
$\omega_j = |C_j| \leq \omega$ for all j

Separable overapproximation

Proposition [Richtárik, Takáč]

If f is partially separable, then for all $S \subseteq \{1, \ldots, n\}$ and $x, h \in \mathbb{R}^N$

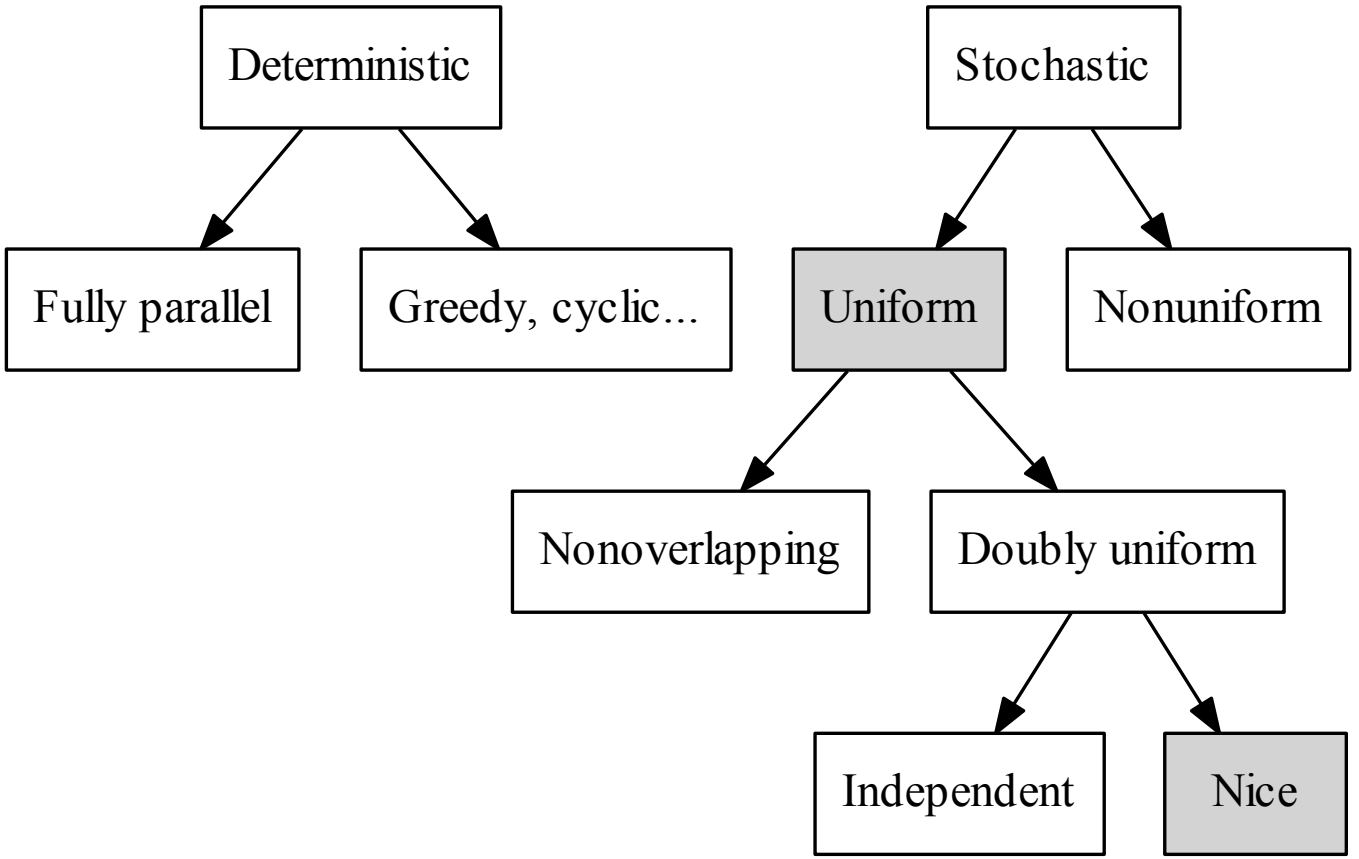
$$f(x + h_{[S]}) \leq f(x) + \langle \nabla f(x), h \rangle + \frac{\omega_S}{2} \|h\|_L^2$$

where

$$h_{[S]}^{(i)} = \begin{cases} h^{(i)} & i \in S \\ 0 & i \notin S \end{cases}$$

$$\omega_S = \min(|S|, \omega)$$

Sampling



Uniform sampling
$\forall i, \mathbf{P}(i \in \hat{S}) = \frac{\tau}{n}$

τ-nice sampling
if $|S| = \tau$,
$\mathbf{P}(\hat{S} = S) = \frac{1}{\binom{n}{\tau}}$

Expected Separable Overapproximation (ESO)

Proposition [Richtárik, Takáč]

If $\hat{S}$ is a τ-nice sampling, then for all $x, h \in \mathbb{R}^N$

$$\mathbf{E}\left[f(x + h_{[\hat{S}]})\right] \leq f(x) + \frac{\tau}{n}\left(\langle \nabla f(x), h\rangle + \frac{\beta}{2}\|h\|_L^2\right)$$

where

$$\beta = 1 + \frac{(\omega - 1)(\tau - 1)}{\max\{1, n-1\}}$$

Example: $n = 10^6, \omega = 10^4, \tau = 10^3$

ESO $\quad \beta \approx 11 \Rightarrow \frac{\tau}{\sqrt{\beta}} \approx 300$

whereas $\quad \omega_S = \min(\omega, \tau) = \tau \Rightarrow \frac{\tau}{\sqrt{\omega_S}} \approx 30$

Refined ESO

Proposition

Assume:

- $f(x) = \sum_{j=1}^{m} \phi_j(e_j^T Ax)$, *where* $\omega_j = |\{i \ : \ A_{j,i} \neq 0\}|$
- $\hat{S}$ *is a* τ*-nice sampling*

Then for all $x, h \in \mathbb{R}^N$

$$\mathbf{E}\left[f(x + h_{[\hat{S}]})\right] \leq f(x) + \frac{\tau}{n}\left(\langle \nabla f(x), h\rangle + \frac{1}{2}\|h\|_v^2\right)$$

where

$$v_i = \sum_{j=1}^{m} \left(1 + \frac{(\omega_j - 1)(\tau - 1)}{\max\{1, n-1\}}\right) L_{\phi_j} A_{j,i}^2$$

Weighted average of ω_j's instead of $\omega = \max_{1 \leq j \leq m} \omega_j$

Are the iterates feasible?

Assume $\psi = I_C$ where C is a convex set

$$z_{k+1}^{(i)} = \arg\min_{z \in \mathbb{R}^{N_i}} \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k v_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \psi_i(z)$$

so $z_{k+1} \in C$, for all k.

$$x_{k+1} = y_k + \frac{n}{\tau}\theta_k(z_{k+1} - z_k) = (1-\theta_k)x_k + \frac{n}{\tau}\theta_k z_{k+1} + \theta_k(1 - \frac{n}{\tau})z_k$$

$\theta_k(1 - \frac{n}{\tau}) \leq 0$: not a convex combination...

Feasibility

Lemma

Let $\{x_k, z_k\}_{k\geq 0}$ *be the iterates of APPROX*

Then for all $k \geq 0$,

$$x_k = \sum_{l=0}^{k} \gamma_k^l z_l$$

where $\gamma_k^0, \gamma_k^1, \ldots, \gamma_k^k \geq 0$ *and* $\sum_{l=0}^{k} \gamma_k^l = 1$

γ_k^l is defined recursively.

Supermartingale inequality

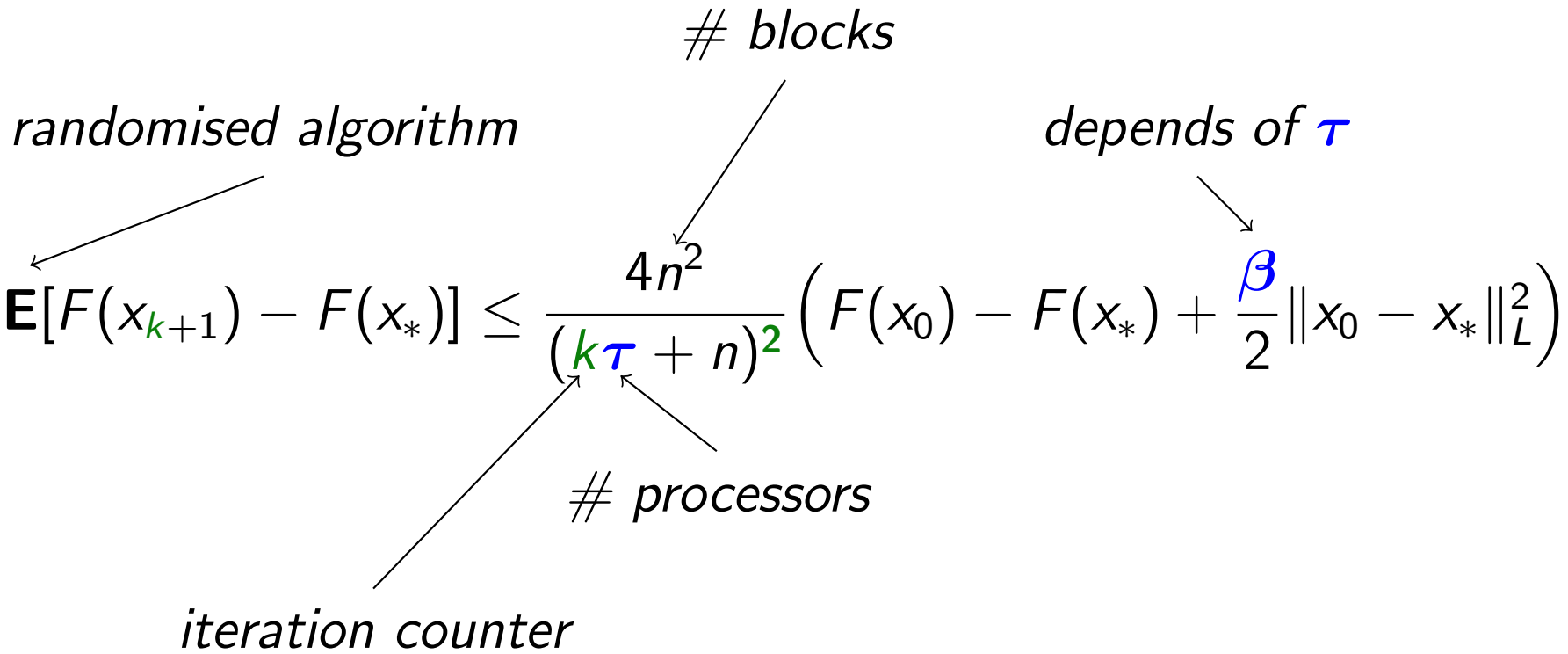
Define: $$\hat{F}_k = f(x_k) + \sum_{l=0}^{k} \gamma_k^l \boldsymbol{\psi}(z_l) \geq F(x_k)$$

Then

$$\mathbf{E}\left[\frac{1-\theta_{k+1}}{\theta_{k+1}^2}(\hat{F}_{k+1} - F(x_*)) + \frac{\boldsymbol{\beta} n^2}{2\boldsymbol{\tau}^2}\|x_* - z_{k+1}\|_L^2 \;\middle|\; S_{k+1}\right]$$
$$\leq \frac{1-\theta_k}{\theta_k^2}(\hat{F}_k - F(x_*)) + \frac{\boldsymbol{\beta} n^2}{2\boldsymbol{\tau}^2}\|x_* - z_k\|_L^2$$

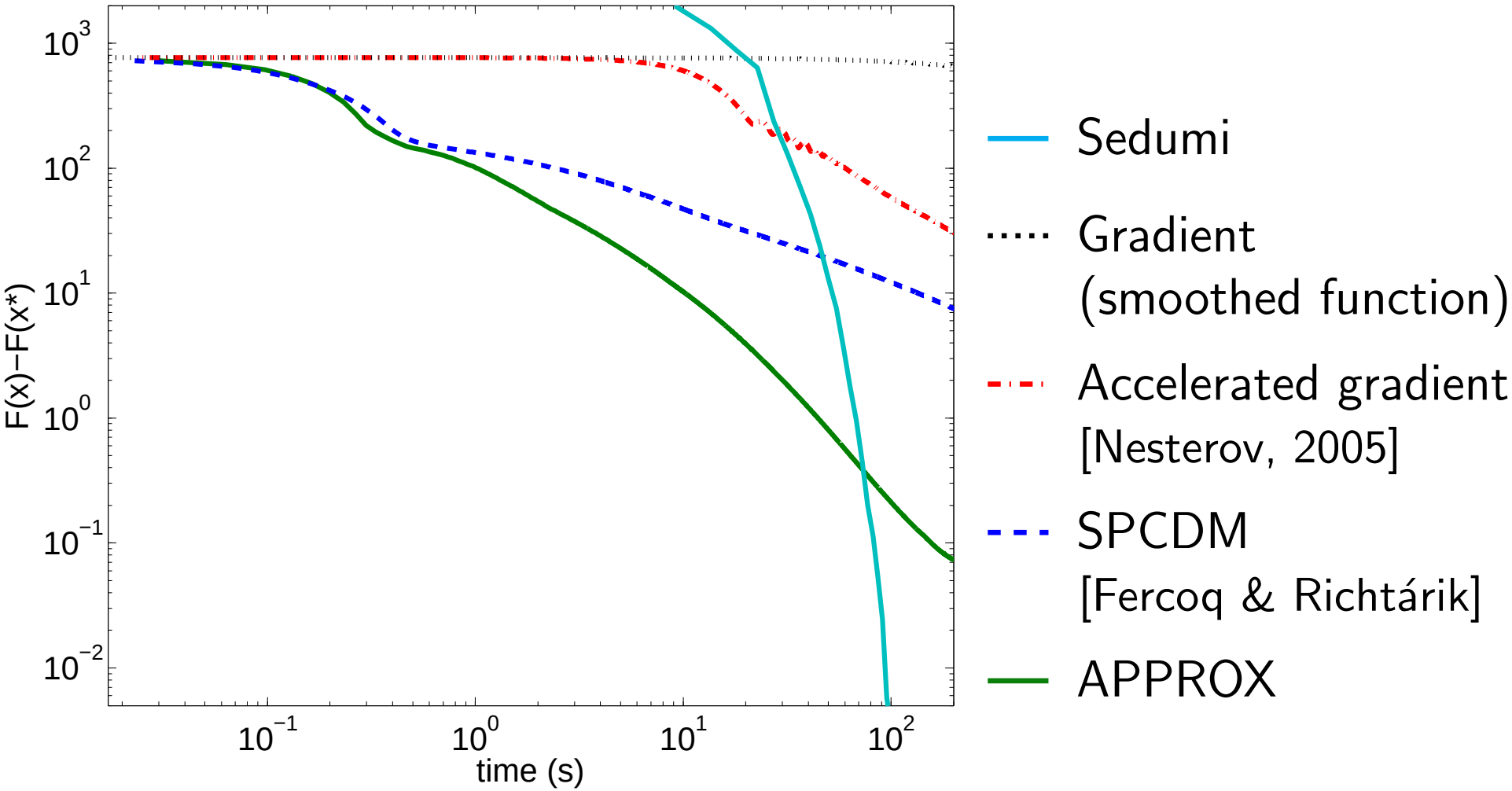
Iteration complexity

Theorem



L_1 Regression

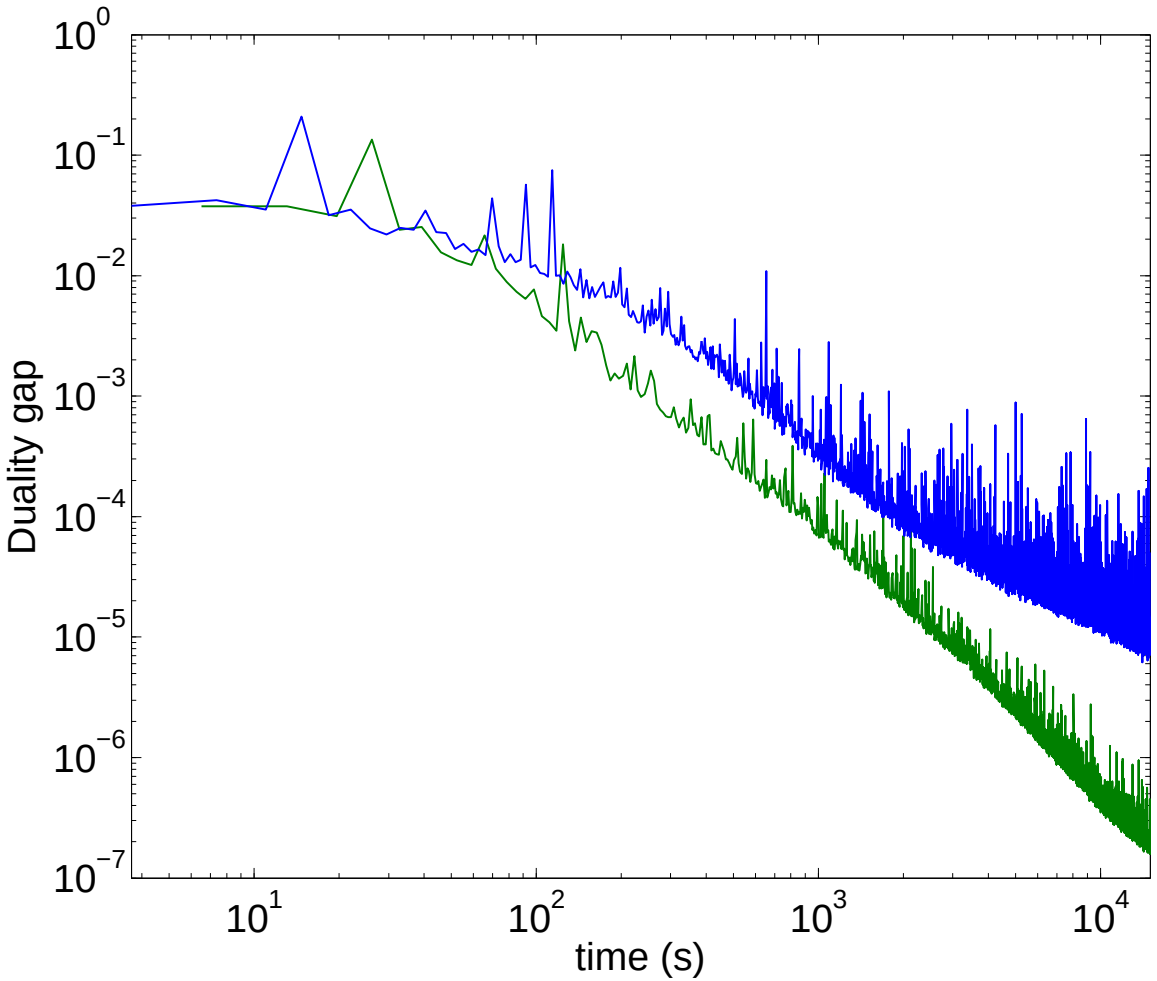
Dorothea : m=800, N=100 000, ω=6 061, $\epsilon = 0{,}1$, τ=4



Comparison of Algorithms for $F(x) = \|Ax - b\|_1 + \|x\|_1$

SVM Dual

Malicious URL: $m = 3{,}231{,}961$, $N = 2{,}396{,}130$, $\tau = 1$



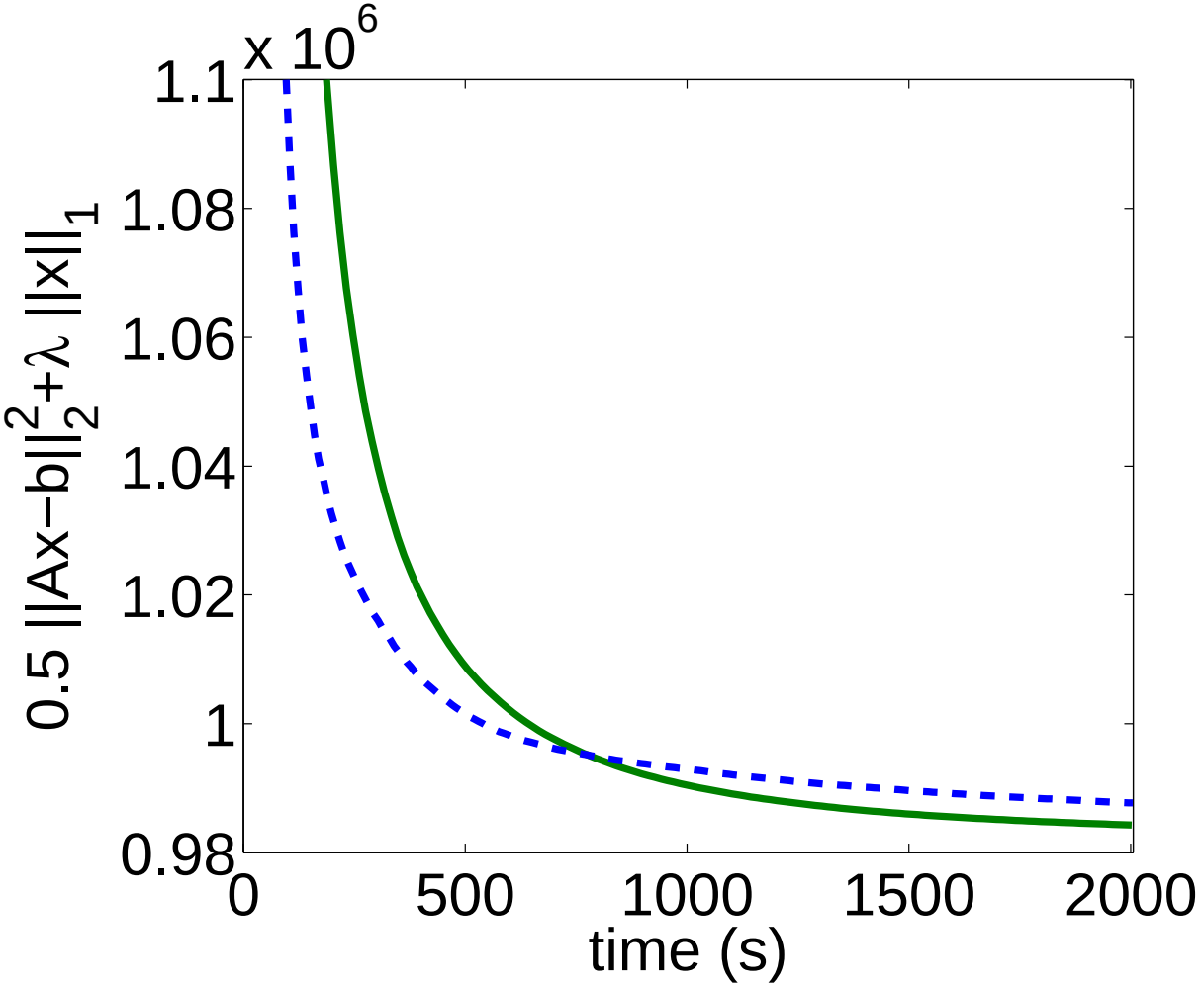
— SDCA [Shalev-Shwartz & Tewari]

— APPROX

Acceleration of Stochastic Dual Coordinate Ascent (SDCA)

Lasso

KDDB: $m = 29,890,095$, $N = 19,264,097$, $\omega = 75$, $\lambda = \frac{\lambda_{\max}}{10^3}$

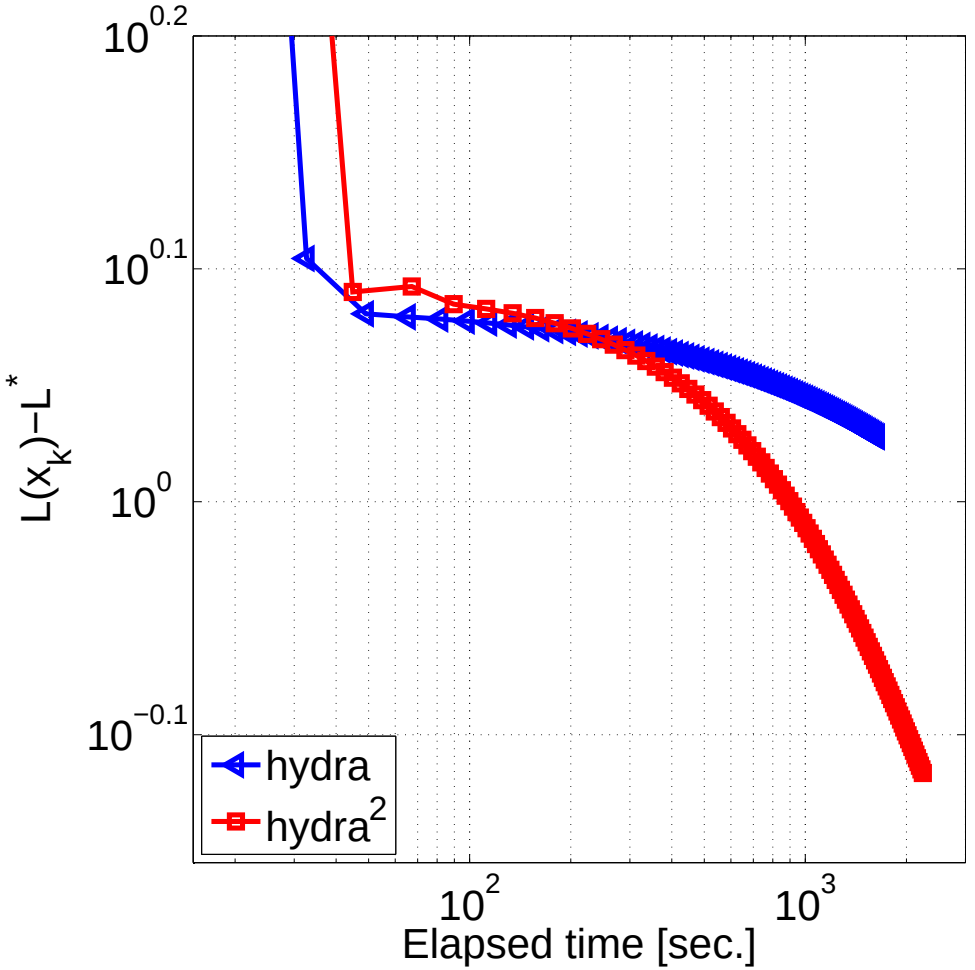


Coordinate descent, $\tau = 16$

APPROX, $\tau = 16$

Quick decrease in the beginning : $8.3\ 10^6 \rightarrow 1.1\ 10^6$

Large scale Lasso



Synthetic problem: 5TB

$m =$ 5,000,000

$N =$ 50,000,000,000

$\bar{\omega} =$ 60,000

Resolution:
256×24 processors of Archer supercomputer

Sampling designed for distributed computation

[Fercoq, Qu, Richtárik, Takáč, Fast Distributed Coordinate Descent for Non-Strongly Convex Losses, 2014]

Extensions

Nearly 50 citations:

[Qu, Richtárik, 2014]
Fixed non-uniform samplings

[Allen-Zhu, Orecchia, 2014]
Large scale continuous packing LP

[Lin, Lu, Xiao, 2014]
Strongly convex functions and primal-dual method

[Ene, Nguyen, 2015]
Restarting and minimisation of submodular functions

[Sun, Toh, Yang, 2015]
Least squares semidefinite programming

Conclusion

- Summary
  - 1^{st} accelerated, parallel and proximal coordinate descent method
  - Improved step sizes for parallel coordinate descent
  - Versatile algorithm, efficient for large scale problems
- Perspectives
  - Relax the i.i.d. sampling assumption
  - Primal-dual algorithm (non-separable & non-smooth)
  - Accelerating stochastic averaged gradient