Parallel Block Coordinate Descent Methods for Huge-Scale Partially Separable Optimization Problems

Peter Richtárik

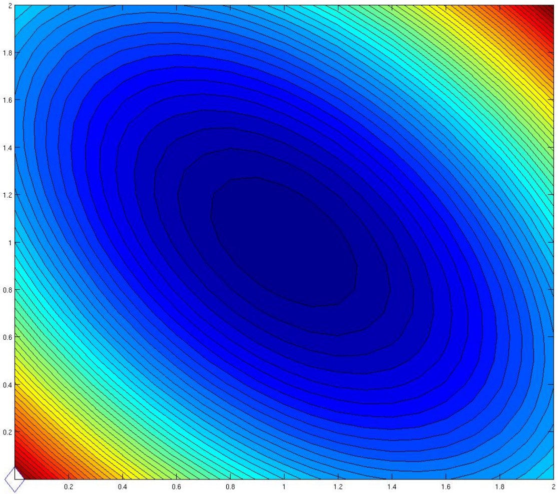
School of Mathematics
The University of Edinburgh

Joint work with Martin Takáč (Edinburgh)

Advances in Large-Scale Optimization ◇ May 24, 2012

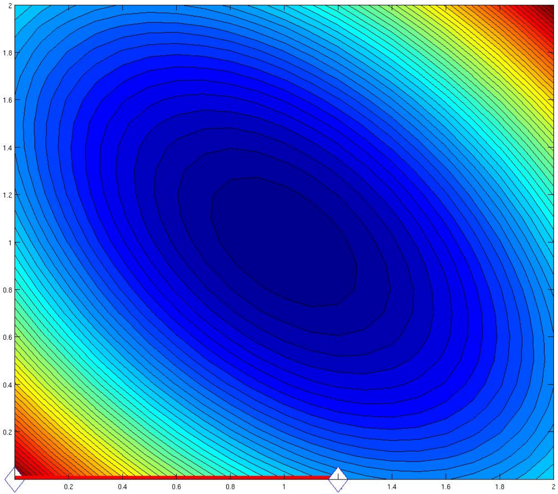
Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



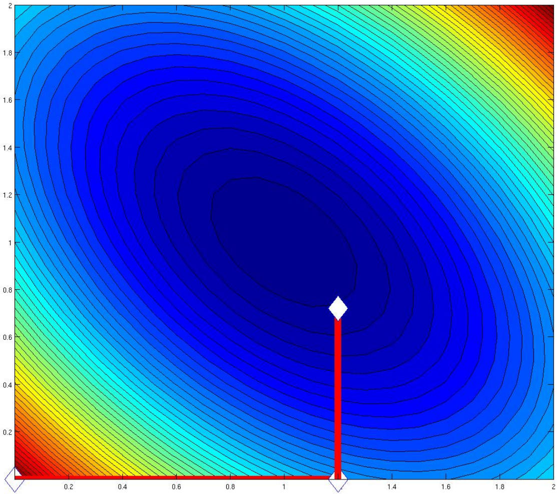
Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



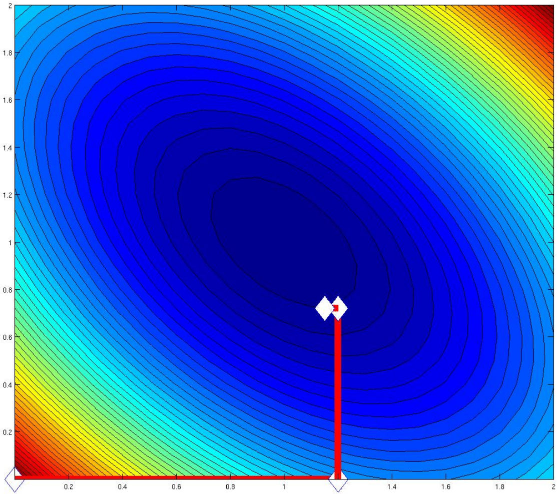
Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



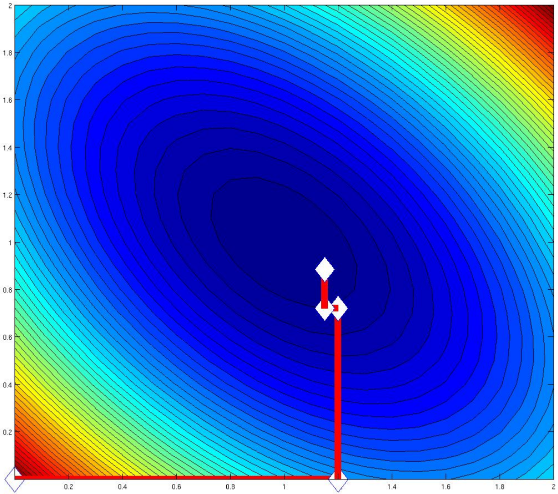
Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



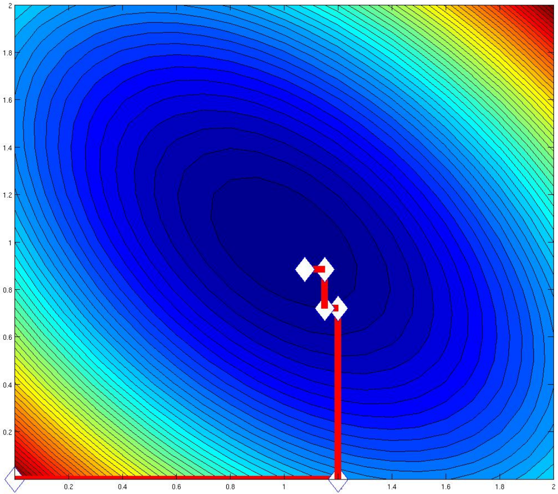
Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



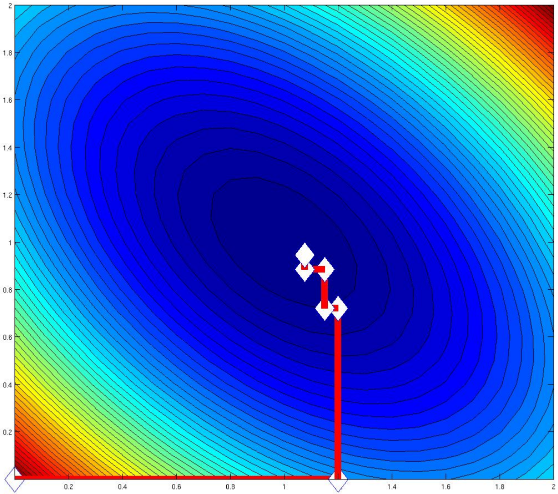
Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



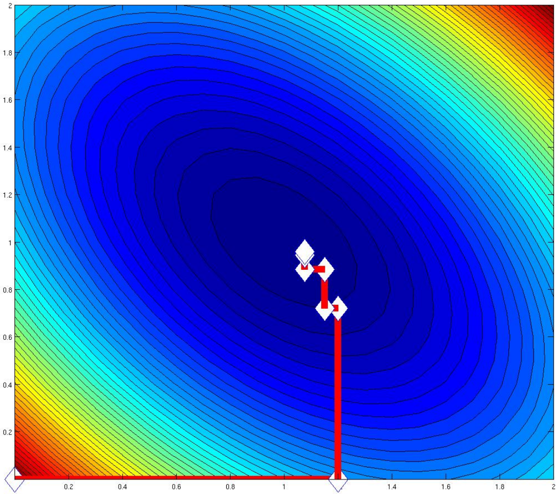
Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



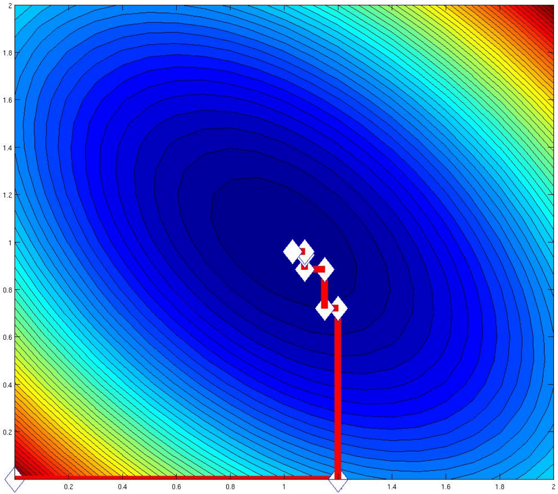
Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



Serial Randomized Coordinate Descent: a 2D example

$$f(x) = \frac{1}{2}x^T \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} x - \begin{pmatrix} 1.5 & 1.5 \end{pmatrix} x, \quad x_0 = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$$



Part I.

Introduction

Blocks

$x \in \mathbf{R}^N$ is partitioned into n blocks/groups of variables:

$$x^{(1)}, \ldots, x^{(n)}, \qquad x^{(i)} \in \mathbf{R}^{N_i}, \qquad \sum_{i=1}^{n} N_i = N$$

Example: $N = 5, n = 2$

$$x = (x_1, x_2, x_3, x_4, x_5)^T = (\underbrace{(x_1, x_3)}_{x^{(1)} \in \mathbf{R}^2}, \underbrace{(x_2, x_4, x_5)}_{x^{(2)} \in \mathbf{R}^3})$$

$$U_1 \underbrace{\begin{pmatrix} x_1 \\ x_3 \end{pmatrix}}_{x^{(1)} \in \mathbf{R}^2} = \begin{pmatrix} x_1 \\ 0 \\ x_3 \\ 0 \\ 0 \end{pmatrix}, \qquad U_2 \underbrace{\begin{pmatrix} x_2 \\ x_4 \\ x_5 \end{pmatrix}}_{x^{(2)} \in \mathbf{R}^3} = \begin{pmatrix} 0 \\ x_2 \\ 0 \\ x_4 \\ x_5 \end{pmatrix}$$

$$x = U_1 x^{(1)} + U_2 x^{(2)}, \quad x^{(i)} = U_i^T x$$

The Problem

$$\min_{x\in\mathbf{R}^N}\{F(x)\overset{\text{def}}{=}f(x)+\Psi(x)\}, \qquad \text{(P)}$$

Assumptions:

- f is convex and smooth
- f is (block/group) partially separable:

$$f=\sum_J f_J, \text{ each } f_J \text{ depends on a few blocks } x^{(i)} \text{ only}$$

- Ψ is convex, nonsmooth and simple
- Ψ is (block/group) separable: $\Psi(x)=\sum_{i=1}^n \Psi_i(x^{(i)})$

$\Psi(x)$	meaning
0	smooth minimization
$\lambda\|x\|_1$ $(N=n)$	term inducing sparsity
$\lambda\sum_{i=1}^n \|x^{(i)}\|_2$	group-sparsity term LASSO
indicator function of a set	(block) separable constraints

- n is huge $(n\approx 10^9)$

Assumption: Smoothness of f

Definition (Block norms and norm in $\mathbf{R}^N$)

We measure the size of block $h^{(i)} \in \mathbf{R}^{N_i}$ of $h \in \mathbf{R}^N$, using a dedicated norm: $\|h^{(i)}\|_{(i)}$, $i = 1, \dots, n$. and measure $x \in \mathbf{R}^N$ via

$$\|x\|_W^2 \stackrel{\text{def}}{=} \sum_{i=1}^{n} w_i \|x^{(i)}\|_{(i)}^2.$$

Assumption (Smoothness of f)

The gradient of f is block coordinate-wise Lipschitz with positive constants $L_1, L_2, \dots, L_n$:

$$\|\nabla_i f(x + U_i t) - \nabla_i f(x)\|_{(i)}^* \le L_i \|t\|_{(i)}, \; x \in \mathbf{R}^N, \; t \in \mathbf{R}^{N_i}, \; i = 1, \dots, n,$$

where

$$\nabla_i f(x) \stackrel{\text{def}}{=} (\nabla f(x))^{(i)} = U_i^T \nabla f(x) \in \mathbf{R}^{N_i}.$$

Applications

- **Machine Learning:** L_1 regularized linear support vector machines (SVM) with various (smooth) loss functions
- **Ranking:** Google problem
- **Engineering:** truss topology design
- **Statistics:** least squares, ridge regression, sparse least squares (LASSO), elastic net, group lasso, (sparse) group LASSO
- **Optimal Design:** c and A optimality

Part II.

PR-BCD: Parallel Randomized Block Coordinate Descent

Parallel Randomized Block Coordinate Descent

Algorithm PR-BCD

Input: Choose initial point $x_0 \in \mathbf{R}^N$

for $k = 0, 1, 2, \ldots$ **do**

1. Choose random subset of blocks (sampling) $\hat{S} \subset \{1, 2, \ldots, n\}$
2. In parallel for $i \in \hat{S}$ do:
   (a) Compute block update: $h^{(i)}(x_k) \in \mathbf{R}^{N_i}$
   (b) Update the block: $x_k^{(i)} = x_k^{(i)} + h^{(i)}(x_k)$
   (c) $x_{k+1} = x_k$

end for

Leads to different algorithms for different samplings $\hat{S}$:

- $\hat{S} = \{i\}, \; i = 1, 2, \ldots, n$, with probability $\frac{1}{n}$ (serial uniform)
- $\hat{S} = \{i\}, \; i = 1, 2, \ldots, n$, with probability p_i (serial non-uniform)
- $\hat{S} = \{1, 2, \ldots, n\}$ with probability 1 (fully parallel)
- $\hat{S} =$ **many interesting choices in between!**

ESO: Expected Separable Overapproximation

Definition (ESO)

f admits an (α, β)-ESO with respect to sampling $\hat{S}$ if

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

- The overapproximation is (block) separable in h:

$$\langle \nabla f(x), h \rangle + \frac{\beta}{2} \|h\|_L^2 = \sum_{i=1}^{n} \left(\langle \nabla_i f(x), h^{(i)} \rangle + \frac{\beta L_i}{2} \|h^{(i)}\|_{(i)}^2 \right) .$$

Step 2(a): Block Update

Update of block i at iteration k:

$$h^{(i)}(x_k) = \arg \min_{h^{(i)} \in \mathbf{R}^{N_i}} \{\langle \nabla_i f(x_k), h^{(i)} \rangle + \tfrac{\beta L_i}{2} \| h^{(i)} \|_{(i)}^2 + \Psi_i(x_k^{(i)} + h^{(i)})\}$$

Step 2(a): Block Update

Update of block i at iteration k:

$$h^{(i)}(x_k) = \arg \min_{h^{(i)} \in \mathbf{R}^{N_i}} \{\langle \nabla_i f(x_k), h^{(i)} \rangle + \tfrac{\beta L_i}{2} \|h^{(i)}\|_{(i)}^2 + \Psi_i(x_k^{(i)} + h^{(i)})\}$$

Special cases: Let $\Psi \equiv 0$ and $\|t\|_{(i)}^2 = \langle B_i t, t\rangle$ for $i = 1, 2, \ldots, n$.

- $n = 1 \implies h^{(1)}(x_k) = -\frac{1}{\beta L_1} B_1^{-1} \nabla f(x_k)$ (steepest descent)
- $n = N \implies h^{(i)}(x_k) = -\frac{1}{\beta L_i} B_i^{-1} \nabla_i f(x_k)$, $i = 1, 2, \ldots, n$ (coordinate descent)

Main Result: Iteration Complexity

Theorem

Assume f admits an (α, β)-ESO with respect to sampling $\hat{S}$. Choose $x_0 \in \mathbf{R}^N$, confidence level $0 < \rho < 1$, target accuracy

$$0 < \epsilon < \min\{\beta \mathcal{R}_L^2(x_0), F(x_0) - F^*\}$$

and iteration counter

$$k \geq \frac{\beta}{\alpha} \left(\frac{2\mathcal{R}_L^2(x_0)}{\epsilon} \log \frac{F(x_0) - F^*}{\epsilon \rho} \right).$$

If x_k is the random point generated by PR-BCD as applied to F, then

$$\mathbf{P}(F(x_k) - F^* \leq \epsilon) \geq 1 - \rho$$

Remarks:

- Specialized results for smooth functions, strongly convex functions, ...

ESO: Computing α, β

Constants α, β, and hence the complexity factor $\frac{\beta}{\alpha}$, depend on

- sampling $\hat{S}$
- function f

$$\frac{\beta}{\alpha} \quad \begin{cases} = n, & \text{serial uniform } \hat{S} \text{ [RT2011]} \\ \ll n, & \text{doubly uniform } \hat{S}, \text{ (block) partially separable } f \text{ [RT2012]} \end{cases}$$

[RT2011] P. R. and Martin Takáč
Iteration complexity of randomized block coordinate descent methods for minimizing a composite function

[RT2012] P. R. and Martin Takáč
Parallel block coordinate descent methods for huge-scale partially separable optimization problems

Doubly Uniform Samplings

Definition

Sampling $\hat{S}$ is doubly uniform if for all $S_1, S_2 \subset \{1, 2, \ldots, n\}$:

$$|S_1| = |S_2| \quad \Longrightarrow \quad \mathbf{P}(\hat{S} = S_1) = \mathbf{P}(\hat{S} = S_2).$$

Doubly Uniform Samplings

Definition

Sampling $\hat{S}$ is doubly uniform if for all $S_1, S_2 \subset \{1, 2, \ldots, n\}$:

$$|S_1| = |S_2| \quad \Longrightarrow \quad \mathbf{P}(\hat{S} = S_1) = \mathbf{P}(\hat{S} = S_2).$$

Remark: A doubly uniform sampling is uniquely characterized by

$$q_k \overset{\text{def}}{=} \mathbf{P}(|\hat{S}| = k), \qquad k = 0, 1, 2, \ldots, n.$$

Indeed, we have

$$\mathbf{P}(\hat{S} = S) = \frac{q_{|S|}}{\binom{n}{|S|}}$$

Doubly Uniform Samplings

Definition

Sampling $\hat{S}$ is doubly uniform if for all $S_1, S_2 \subset \{1, 2, \ldots, n\}$:

$$|S_1| = |S_2| \quad \Longrightarrow \quad \mathbf{P}(\hat{S} = S_1) = \mathbf{P}(\hat{S} = S_2).$$

Remark: A doubly uniform sampling is uniquely characterized by

$$q_k \stackrel{\text{def}}{=} \mathbf{P}(|\hat{S}| = k), \qquad k = 0, 1, 2, \ldots, n.$$

Indeed, we have

$$\mathbf{P}(\hat{S} = S) = \frac{q_{|S|}}{\binom{n}{|S|}}$$

Example: $n = 3$

- $\mathbf{P}(\emptyset) = q_0$
- $\mathbf{P}(\{1\}) = \mathbf{P}(\{2\}) = \mathbf{P}(\{3\}) = \mathbf{P}(\{4\}) = \frac{q_1}{4}$
- $\mathbf{P}(\{1,2\}) = \mathbf{P}(\{1,3\}) = \mathbf{P}(\{1,4\}) = \mathbf{P}(\{2,3\}) = \mathbf{P}(\{2,4\}) = \mathbf{P}(\{3,4\}) = \frac{q_2}{6}$
- $\mathbf{P}(\{1,2,3\}) = \mathbf{P}(\{1,2,4\}) = \mathbf{P}(\{1,3,4\}) = \mathbf{P}(\{2,3,4\}) = \frac{q_3}{4}$
- $\mathbf{P}(\{1,2,3,4\}) = q_4$

What Can We Model with Doubly Uniform Samplings?

Recall that

$$q_k = \mathbf{P}(|\hat{S}| = k)$$

Two Examples of Computing Environments:

- τ reliable processors: $q_\tau = 1$
- τ unreliable/busy processors with each giving an answer, independently, with probability p:

$$q_k = \binom{\tau}{k} p^k (1-p)^k, \qquad k = 0, 1, \ldots, \tau$$

Partial Separability of f

Definition

Function f is (block) partially separable of degree ω if

$$f(x) = \sum_{J \in \mathcal{J}} f_J(x), \qquad \mathcal{J} \text{ consists of (some) subsets of } \{1, 2, \dots, n\}$$

- f_J depends on blocks $x^{(i)}$ for $i \in J$ only
- $\max_{J \in \mathcal{J}} |J| \leq \omega$

Partial Separability of f

Definition

Function f is (block) partially separable of degree ω if

$$f(x) = \sum_{J \in \mathcal{J}} f_J(x), \qquad \mathcal{J} \text{ consists of (some) subsets of } \{1, 2, \ldots, n\}$$

- f_J depends on blocks $x^{(i)}$ for $i \in J$ only
- $\max_{J \in \mathcal{J}} |J| \leq \omega$

Observe that:

- $1 \leq \omega \leq n$

Partial Separability of f

Definition

Function f is (block) partially separable of degree ω if

$$f(x) = \sum_{J \in \mathcal{J}} f_J(x), \qquad \mathcal{J} \text{ consists of (some) subsets of } \{1, 2, \ldots, n\}$$

- f_J depends on blocks $x^{(i)}$ for $i \in J$ only
- $\max_{J \in \mathcal{J}} |J| \leq \omega$

Observe that:

- $1 \leq \omega \leq n$
- If $\omega = 1$, f is (block) separable (as Ψ) and hence the main optimization problem can be decomposed into n independent problems

Partial Separability of f

Definition

Function f is (block) partially separable of degree ω if

$$f(x) = \sum_{J \in \mathcal{J}} f_J(x), \qquad \mathcal{J} \text{ consists of (some) subsets of } \{1, 2, \ldots, n\}$$

- f_J depends on blocks $x^{(i)}$ for $i \in J$ only
- $\max_{J \in \mathcal{J}} |J| \leq \omega$

Observe that:

- $1 \leq \omega \leq n$
- If $\omega = 1$, f is (block) separable (as Ψ) and hence the main optimization problem can be decomposed into n independent problems

Example: $f(x) = \frac{1}{2}\|Ax - b\|^2 = \sum_j \underbrace{\frac{1}{2}(A_j^T x - b_j)^2}_{f_j(x)}$,

- $\omega = \max \#$ of nonzeros in A_j (a row of A)

ESO for Partially Separable f and Doubly Uniform $\hat{S}$

Theorem

Assume that

- f is partially separable of degree ω
- $\hat{S}$ is a doubly uniform sampling

Then f admits an (α, β)-ESO with respect to $\hat{S}$ with

$$\alpha = \frac{\mathbf{E}[|\hat{S}|]}{n}, \qquad \beta = 1 + \frac{(\omega - 1)\left(\frac{\mathbf{E}[|\hat{S}|^2]}{\mathbf{E}[|\hat{S}|]} - 1\right)}{\max(1, n-1)}.$$

Remarks:

- We have computed ESO for some other samplings as well, but will not talk about them here

ESO Coefficients

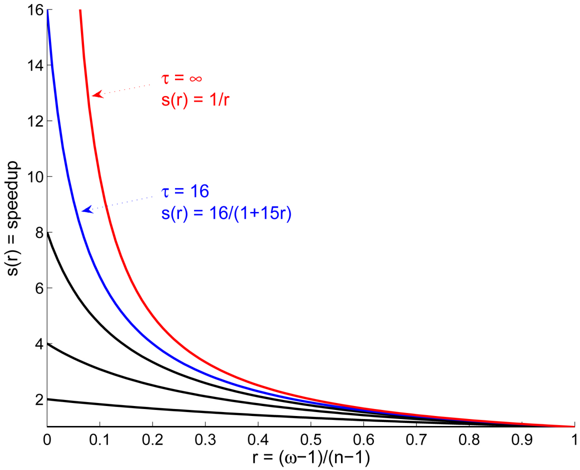
Sampling $\hat{S}$	α	β
Doubly Uniform (DU)	$\frac{\mathbf{E}[\lvert\hat{S}\rvert]}{n}$	$1 + \frac{(\omega-1)\left(\frac{\mathbf{E}[\lvert\hat{S}\rvert^2]}{\mathbf{E}[\lvert\hat{S}\rvert]}-1\right)}{\max(1,n-1)}$
DU unreliable	$\frac{\tau p}{n}$	$1 + \frac{(\omega-1)(\tau-1)p}{\max(1,n-1)}$
DU reliable	$\frac{\tau}{n}$	$1 + \frac{(\omega-1)(\tau-1)}{\max(1,n-1)}$
DU fully parallel	1	ω
DU serial	$\frac{1}{n}$	1

Parallelization Speedup for DU Samplings

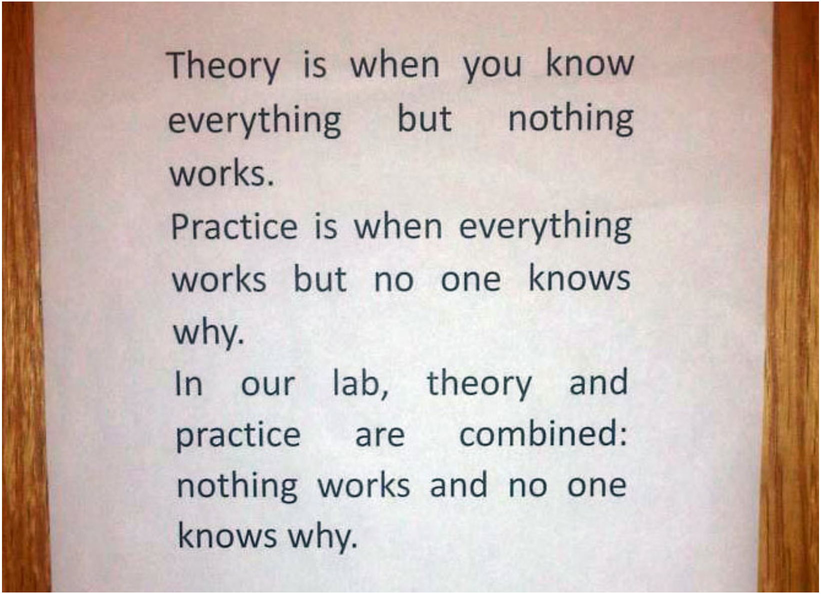
Sampling $\hat{S}$	$\frac{\beta}{\alpha}$ of sampling / $\frac{\beta}{\alpha}$ of DU serial
Doubly Uniform (DU)	$\dfrac{\max(1,n-1)}{\left(\frac{\mathbf{E}[\lvert\hat{S}\rvert^2]}{(\mathbf{E}\lvert\hat{S}\rvert)^2}-\frac{1}{\mathbf{E}[\lvert\hat{S}\rvert]}\right)(\omega-1)+\frac{1}{\mathbf{E}[\lvert\hat{S}\rvert]}\max(1,n-1)}$
DU unreliable	$\dfrac{\max(1,n-1)}{\left(1-\frac{1}{\tau}\right)(\omega-1)+\frac{1}{\tau}\frac{\max(1,n-1)}{p}}$
DU reliable	$\dfrac{\max(1,n-1)}{\left(1-\frac{1}{\tau}\right)(\omega-1)+\frac{1}{\tau}\max(1,n-1)}$
DU fully parallel	$\frac{n}{\omega}$
DU serial	1

Parallelization Speedup for DU Reliable Sampling

$$s(r) = \frac{\tau}{1 + (\tau - 1)r}, \qquad r = \frac{\omega - 1}{\max(n - 1, 1)} \in [0, 1]$$

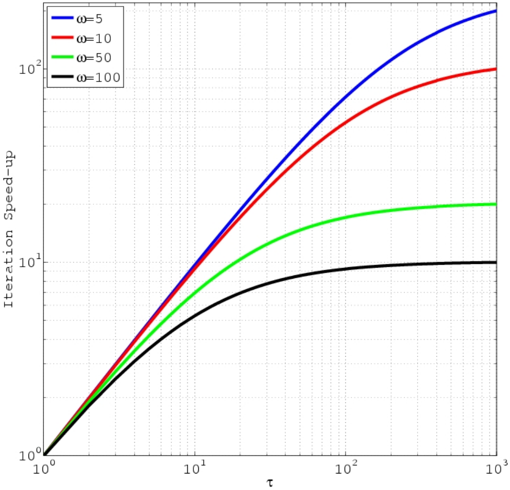


Parallelization Speedup: Theory vs Practice



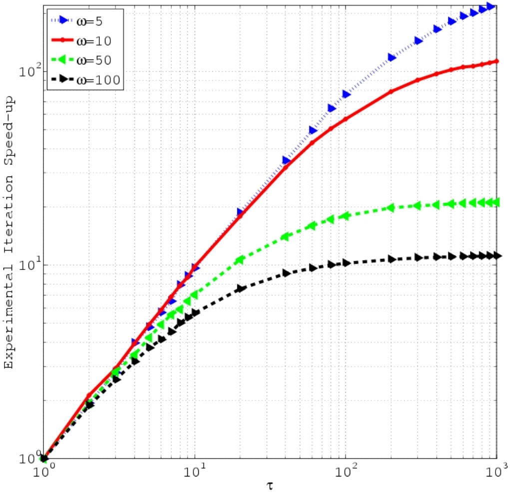
Parallelization Speedup: Theory

$$F(x) = f(x) = \frac{1}{2}\|Ax - b\|_2^2, \qquad A \in \mathbf{R}^{3000\times 1000}$$



Parallelization Speedup: Experiment

$$F(x) = f(x) = \frac{1}{2}\|Ax - b\|_2^2, \qquad A \in \mathbf{R}^{3000\times 1000}$$



Part III.

Serial Methods

Serial Methods: Comparison with Nesterov's Approach

...for achieving $\mathbf{P}(F(x_k) - F^* \leq \epsilon) \geq 1 - \rho$ in the general case (F not strongly convex):

Alg	Ψ	p_i	norms	complexity	objective
[N10]	0	$\frac{L_i}{S}$	Euclid.	$\frac{8n\mathcal{R}_L^2(x_0)}{\epsilon} \log \frac{4(f(x_0)-f^*)}{\epsilon\rho}$	$f(x) + \frac{\epsilon\\|x-x_0\\|_L^2}{8\mathcal{R}_L^2(x_0)}$
[N10]	0	$\frac{1}{n}$	Euclid.	$(2n + \frac{8S\mathcal{R}_I^2(x_0)}{\epsilon}) \log \frac{4(f(x_0)-f^*)}{\epsilon\rho}$	$f(x) + \frac{\epsilon\\|x-x_0\\|_I^2}{8\mathcal{R}_I^2(x_0)}$
[RT11]	0	> 0	general	$\frac{2\mathcal{R}_{LP^{-1}}^2(x_0)}{\epsilon}(1 + \log \frac{1}{\rho}) - 2$	f
[RT11]	$\neq 0$	$\frac{1}{n}$	Euclid.	$\frac{2nM}{\epsilon}(1 + \log \frac{1}{\rho})$ $\frac{2n\mathcal{R}_L^2(x_0)}{\epsilon} \log \frac{F(x_0)-F^*}{\epsilon\rho}$	F

Symbols used in the table: $S = \sum L_i$, $M = \max\{\mathcal{R}_L^2(x_0), F(x_0) - F^*\}$.

[N10] Yu. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems. *CORE Discussion Paper #2010/2*, 2010

[RT11] P. R. and Martin Takáč, Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function, 2011

Sparse LS Instance 1: $A \in \mathbf{R}^{10^7 \times 10^6}$, $\|A\|_0 = 10^8$

k/n	$F(x_k) - F^*$	$\|x_k\|_0$	time [s]
0.0010	$< 10^{16}$	857	0.01
12.72	$< 10^{11}$	999,246	54.48
15.23	$< 10^{10}$	997,944	65.19
18.61	$< 10^{9}$	990,923	79.69
20.61	$< 10^{8}$	978,761	88.25
23.38	$< 10^{7}$	926,167	100.08
25.91	$< 10^{6}$	763,314	110.94
28.22	$< 10^{5}$	366,835	120.84
30.66	$< 10^{4}$	57,991	131.25
32.83	$< 10^{3}$	9,701	140.55
35.05	$< 10^{2}$	2,538	150.02
37.01	$< 10^{1}$	1,722	158.39
38.26	$< 10^{0}$	1,633	163.75
40.98	$< 10^{-1}$	1,604	175.38
42.71	$< 10^{-3}$	1,600	182.75
42.71	$< 10^{-4}$	1,600	182.77
43.28	$< 10^{-5}$	1,600	185.21
44.86	$< 10^{-6}$	1,600	191.94

Sparse LS Instance 2: $A \in \mathbf{R}^{10^9 \times 10^8}$, $\|A\|_0 = 2 \times 10^9$

k/n	$F(x_k) - F^*$	$\|x_k\|_0$	time [s]
0.01	$< 10^{18}$	18,486	1.32
9.35	$< 10^{14}$	99,837,255	1294.72
11.97	$< 10^{13}$	99,567,891	1657.32
14.78	$< 10^{12}$	98,630,735	2045.53
17.12	$< 10^{11}$	96,305,090	2370.07
20.09	$< 10^{10}$	86,242,708	2781.11
22.60	$< 10^{9}$	58,157,883	3128.49
24.97	$< 10^{8}$	19,926,459	3455.80
28.62	$< 10^{7}$	747,104	3960.96
31.47	$< 10^{6}$	266,180	4325.60
34.47	$< 10^{5}$	175,981	4693.44
36.84	$< 10^{4}$	163,297	5004.24
39.39	$< 10^{3}$	160,516	5347.71
41.08	$< 10^{2}$	160,138	5577.22
43.88	$< 10^{1}$	160,011	5941.72
45.94	$< 10^{0}$	160,002	6218.82
46.19	$< 10^{-1}$	160,001	6252.20
46.25	$< 10^{-2}$	160,000	6260.20
46.89	$< 10^{-3}$	160,000	6344.31
46.91	$< 10^{-4}$	160,000	6346.99
46.93	$< 10^{-5}$	160,000	6349.69

Sparse LS Instance 3: $A \in \mathbf{R}^{10^{10} \times 10^9}$, $\|A\|_0 = 2 \times 10^{10}$

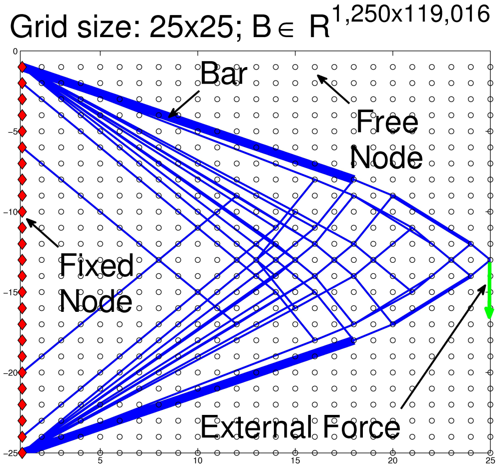
k/n	$\frac{F(x_k)-F^*}{F(x_0)-F^*}$	$\|x_k\|_0$	time [hours]
0	10^0	0	0.00
1	10^{-1}	14,923,993	1.43
3	10^{-2}	22,688,665	4.25
16	10^{-3}	24,090,068	22.65

Part IV.

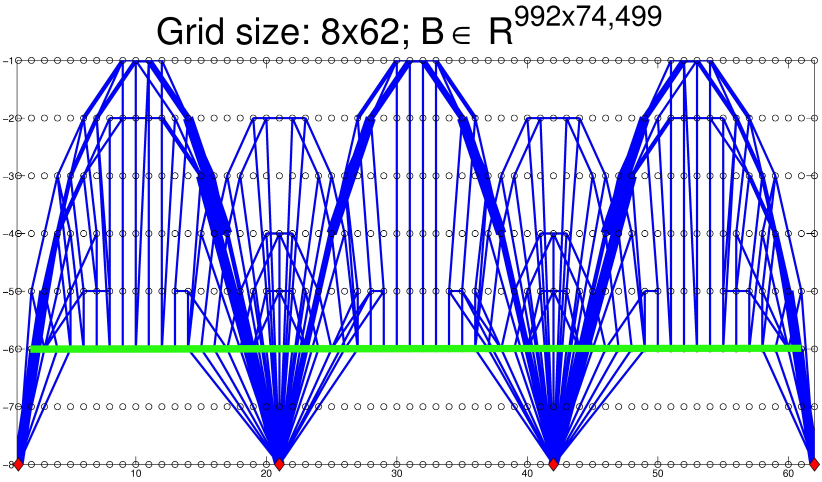
Computational Results with Parallel Code

Truss Topology Design: The Problem

$$f(x) = \frac{\gamma}{2}\|Bx - d\|_2^2, \quad \Psi(x) = \|x\|_1$$



Truss Topology Design: A Bridge



Truss Topology Design: Performance Test

$r \times c$	$2m$	N	$\|B\|_0 = 4n$
20x20	800	48,934	195,736
30x30	1800	246,690	986,760
40x40	3200	779,074	3,116,296
50x50	5000	1,901,930	7,607,720
80x80	12800	12,454,678	49,818,712
100x100	20000	30,398,894	121,595,576

$r \times c$	time (iter.) SeDuMi	it/sec SG	accel PG	it/sec SR	accel PR	it/sec NEST
20x20	61 (40)	5,101.13	0.62	2.4M	28.29	53.05
30x30	658 (10) NP	1,194.61	2.65	2.2M	26.13	9.01
40x40	8.6k (32) NP	380.36	7.95	1.9M	24.83	3.43
50x50	48.4k (4) NP	159.13	15.98	1.5M	27.15	1.41
80x80	X	25.63	42.34	1.5M	34.01	0.15
100x100	X	10.66	52.18	1.2M	30.86	0.05

Algorithms: **S**erial **G**reedy, **P**arallel **G**reedy, **S**erial **R**andom, **P**arallel **R**andom. For SeDuMi: NP = Numerical Problems, X = no iteration after 10 hours. NEST = Nesterov's accelerated method.