

**SESOP: Sequential Subspace
Optimization Method
for Very Large Scale Optimization
Problems**

Michael Zibulevsky

Department of Computer Science
Technion – Israel Institute of Technology

Collaborators

Michael Elad

Boaz Matalon

Guy Narkiss

Arkadi Nemirovski

Joseph Stock ...

Problem Formulation

Find a local min of a smooth function of very large number of variables ($10^4 - 10^7$ and more):

$$\min f(\mathbf{x}), \mathbf{x} \in \mathcal{R}^n$$

Applications

Signal and image processing;

Inverse problems (e.g. Tomography)

Pattern recognition;

And many many others.

SESOP Properties

- Cost per iteration grows linearly in problem size n .
- Optimal worst case complexity for convex problems:

Let L - Lipschitz constant of ∇f . After iteration k

$$f(x^k) - f_{\text{opt}} < L \frac{\|x^0 - x^{\text{opt}}\|^2}{k^2}$$

,

- Very good behavior in many practical problems:
 - Sparse signal/image decomposition using frames;
 - Image denoising and deblurring based on sparsity;
 - Computerized Tomography;
 - Support Vector Machines (SVM)

Basic SESOP Iteration

\mathcal{M}_k – affine subspace including current iterate x^k and directions of grad. g^k and few previous steps

$$\mathcal{M}_k = \{x^k + P_k \alpha, \quad \alpha \in \mathcal{R}^{M_k}\}$$

$$P_k = [g^k, x^k - x^{k-1}, x^{k-1} - x^{k-2}, \dots, x^{k-M-1} - x^{k-M-2}]$$

SESOP Iteration k :

minimize f over subspace \mathcal{M}_k

$$\alpha^k = \arg \min_{\alpha} f(x^k + P_k \alpha)$$

$$x^{k+1} = x^k + P_k \alpha^k$$

Reduced computation in subspace

Let $f(\mathbf{x}) = \varphi(\mathbf{Ax})$.

- Computing \mathbf{Ax} – expensive; $\varphi(\cdot)$ – cheap.
- For $\mathbf{x} = \mathbf{x}^k + \mathbf{P}\alpha$, term \mathbf{Ax} is broken into

$$\mathbf{Ax} = \mathbf{Ax}^k + \sum_{i=1}^M \alpha_i \mathbf{A}\mathbf{p}_i = \mathbf{v}_0 + \sum_{i=1}^M \alpha_i \mathbf{v}_i,$$

- One matrix-vector multiplication per subspace optimization: Compute and save

$$\mathbf{v}_{new} = \mathbf{A}\mathbf{p}_{new}$$

- Re-computations inside the subspace – cheap.

Optimal worst-case complexity for convex problems

If two additional *Nemirovski directions* are included into search subspace:

$$\mathbf{x}^k - \mathbf{x}^0 \quad \text{and} \quad \sum_{i=0}^k w_i \mathbf{g}^i,$$

where $w_0 = 1$, and $w_k = \frac{1}{2} + \sqrt{\frac{1}{4} + w_{k-1}^2}$ for $k > 0$,

Then SESOP obtains the optimal worst-case complexity

$$f(x^k) - f_{\text{opt}} < L \frac{\|x^0 - x^{\text{opt}}\|^2}{k^2}$$

Usually, in practice, these two directions are not necessary.

Link to Conjugate Gradients (CG)

- **CG iteration:** exact line search

$$x^{k+1} = x^k + \gamma d^k$$

in direction $d^k = -g^k + \beta d^{k-1}$

where $\beta = \|g^k\|^2 / \|g^{k-1}\|^2$

- When f is *quadratic*, CG step minimizes it over affine subspace $\mathcal{M}_k = x^k + \text{span}(g^k, d^{k-1})$, i.e. is equivalent to SESOP-1

- By *Expanding Manifold property*, k-th CG step minimizes quadratic f also over subspace

$$\{x^k + \text{span}(g^k, d^{k-1}, d^{k-2}, \dots, d^0)\}$$

i.e. CG is equivalent to SESOP-(1...n)

Non-linear CG versus SESOP

- When f is not quadratic, non-linear CG produce accurate line search in direction

$$d^k = -g^k + \beta d^{k-1},$$

Most popular and robust Polak-Ribiere formula

$$\beta = \frac{(g^k)^T (g^k - g^{k-1})}{\|g^{k-1}\|^2}$$

leads to standard CG when f is quadratic

- SESOP is an alternative generalization of quadratic CG: it preserves explicitly the principle of subspace optimization.
- In contrast to Polak-Ribiere CG, SESOP has proven optimal worst-case complexity.

Acceleration by Preconditioning

- W – efficient approximation of inverse Hessian (e.g. its inverse diagonal).
- Wg^k substitutes g^k in SESOP subspace
- Imitates Newton direction.
- In quadratic case the method is equivalent to Preconditioned CG

Adding Elad's direction of PCD

(Parallel Coordinate Descent)

- Perform set of coordinate-wise optimizations without moving away

$$s_i^k = \arg \min_{\mu} f(x^k + \mu e_i), \quad i = 1, \dots, n$$

- Vector s^k substitutes g^k in SESOP subspace.
- Extremely efficient in diagonally penalized linear least-square problems (e.g. Basis Pursuit)

Computational Examples

- Tomography
- Basis Pursuit Denoising and Deblurring
- Support Vector Machines

Example 1: Sparse Tomography

- Tomography problem for sparse images:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{R}\mathbf{x} - \mathbf{y}\|_2^2 + \mu \|\mathbf{x}\|_1,$$

\mathbf{R} – Radon transform

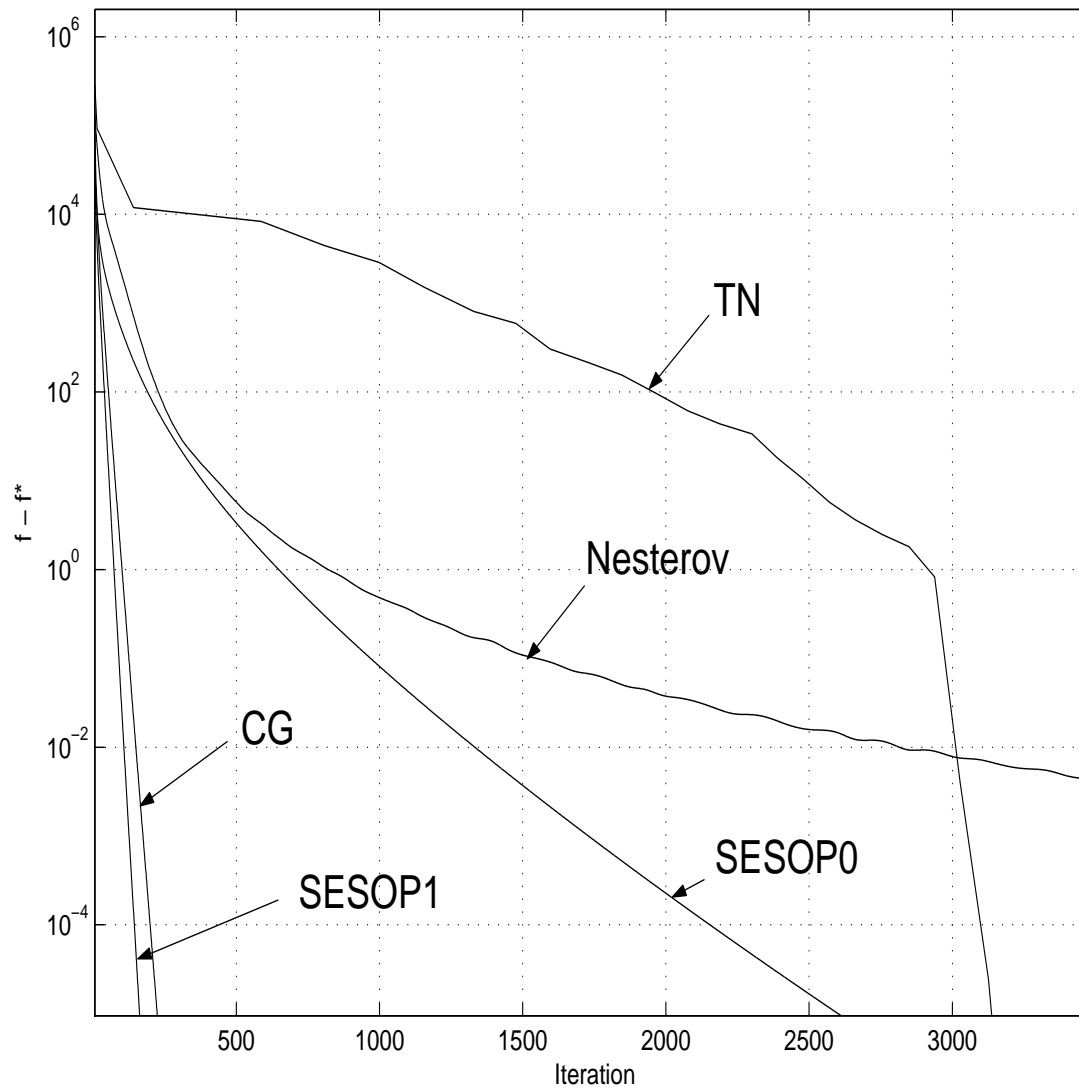
\mathbf{x} – unknown original image

\mathbf{y} – observed noisy projection data

μ – regularization parameter.

- We use smooth approximation of the l_1 -norm.
- SESOP outperforms CG by 25% – 50%

Sparse Tomography: $f - f_{opt}$ with iterations



Example 2: Basis Pursuit Denoising

$$\min_{\mathbf{c}} \frac{1}{2} \|\Phi \mathbf{c} - \mathbf{y}\|_2^2 + \mu \|\mathbf{c}\|_1,$$

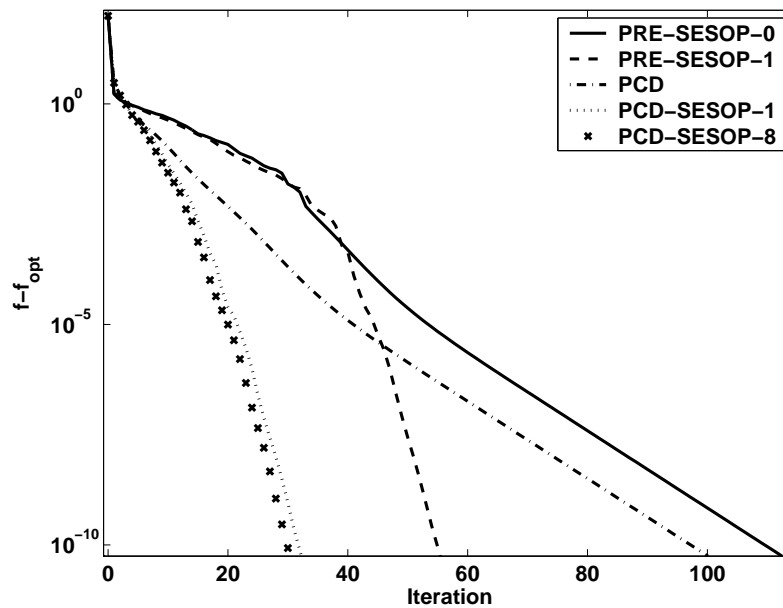
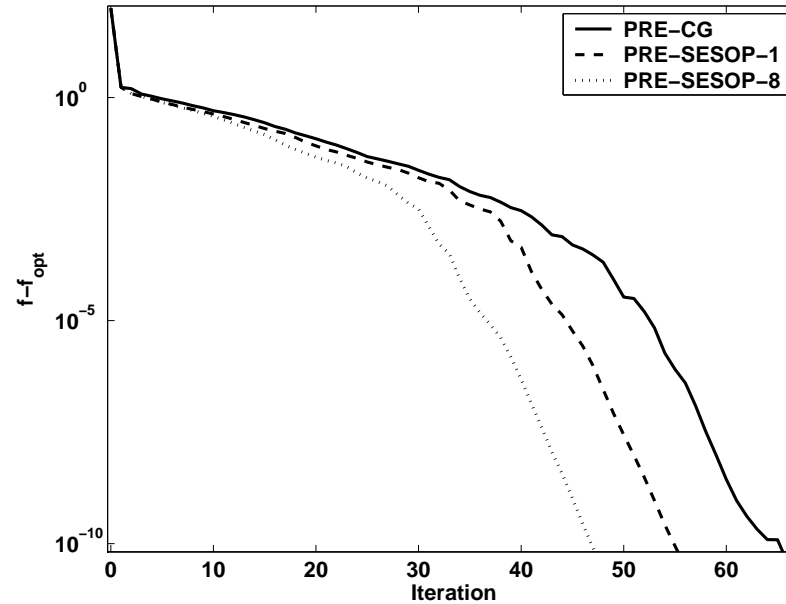
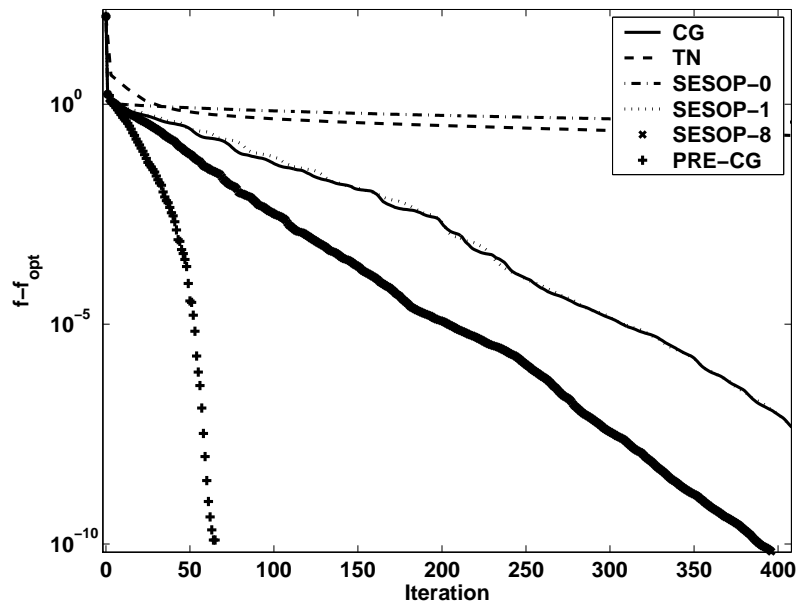
Φ – Overcomplete signal dictionary

\mathbf{c} – unknown sparse decomposition coefficients

\mathbf{y} – observed noisy signal

μ – regularization parameter.

We use smooth approximation of the l_1 -norm.



Basis Pursuit: Evolution of objective func. error

Original

Noisy Blurred

Restored



Deblurring results with the PCD-SESOP

. Noise variance: Top $\sigma^2 = 2$; bottom $\sigma^2 = 8$.

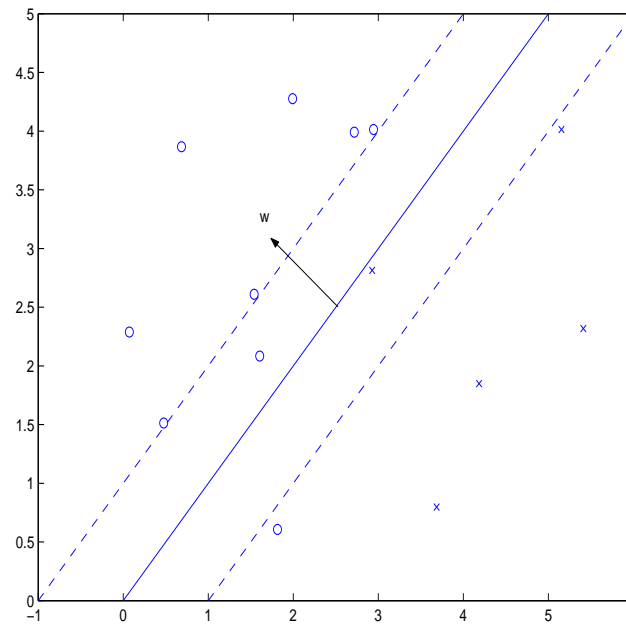
Example 3: Support Vector Machines

Soft-margin L_1 and L_2 -SVM:

$$\min_{\mathbf{w}, b, \xi} \|\mathbf{w}\|_p^p + c \sum_i \xi_i^q$$

$$s.t. \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i,$$

$$p, q \in \{0, 1\}$$



Support Vector Machines (cont)

Equivalent unconstrained problem:

$$\min_{\mathbf{w}, b} \|\mathbf{w}\|_p^p + C \sum_{i=1}^m \tilde{\varphi}_q(-y_i(\mathbf{w}^T \mathbf{x}_i + b)),$$

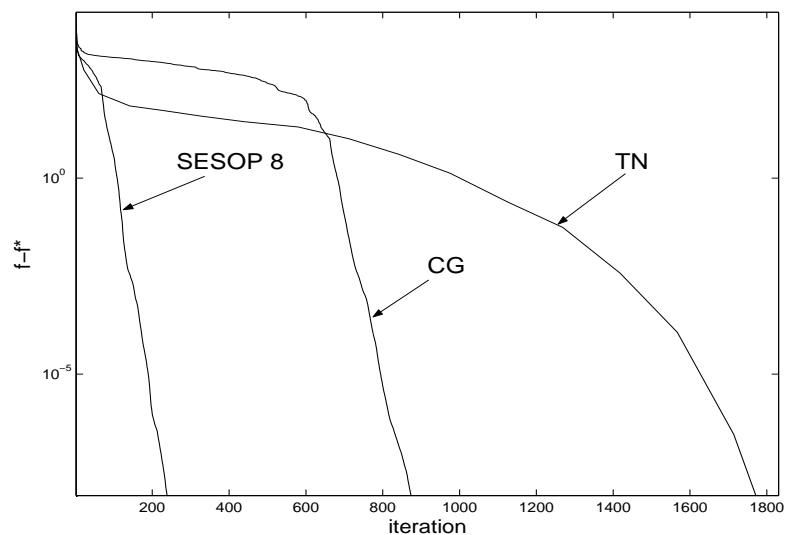
Penalty functions:

$$\tilde{\varphi}_1(t) = \frac{1}{2}(|t + 1| + t + 1)$$
$$\tilde{\varphi}_2(t) = \begin{cases} 0 & \text{for } t \leq -1, \\ (t + 1)^2 & \text{otherwise,} \end{cases}$$

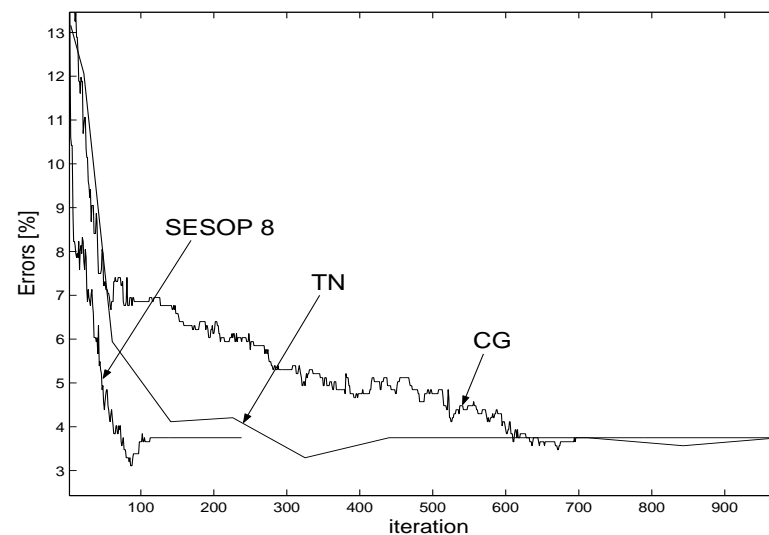
Support Vector Machines (cont)

- Six problems with $10^3 - 10^6$ variables were solved.
- SESOP was consistently the fastest method
- SESOP outperformed (on average):
 - TN – ten times;
 - CG – two times.

Accuracy of the obj. func.



Validation error



Data set 'Internet_ads'

SESOP-TN: Combining SESOP with Truncated Newton

At every outer iteration **Truncated Newton (TN)** [Dembo-1983, Nash-2000] approximately minimizes quadratic Taylor expansion

$$q_k(x) = f(x^k) + g^{kT} (x - x^k) + \frac{1}{2}(x - x^k)^T H_k (x - x^k), \quad (1)$$

around the current iterate x^k , using limited number of CG steps, where

$g^k = \nabla f(x^k)$ – gradient of f at x^k

$H_k = \nabla^2 f(x^k)$ – Hessian.

The outer iteration of TN is accomplished with a line search, in order to guarantee function decrease.

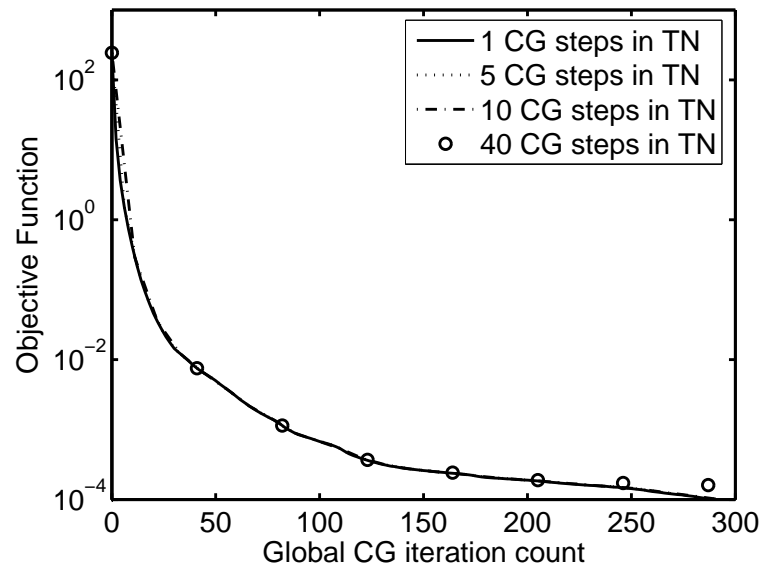
The overall effectiveness of the TN method is rather sensitive to the choice of stopping rule for the internal CG optimization. We overcome this difficulty, replacing line search with subspace optimization. In this way we allow the CG iterations to stay matched through consequent TN steps.

Summary of SESOP-TN algorithm

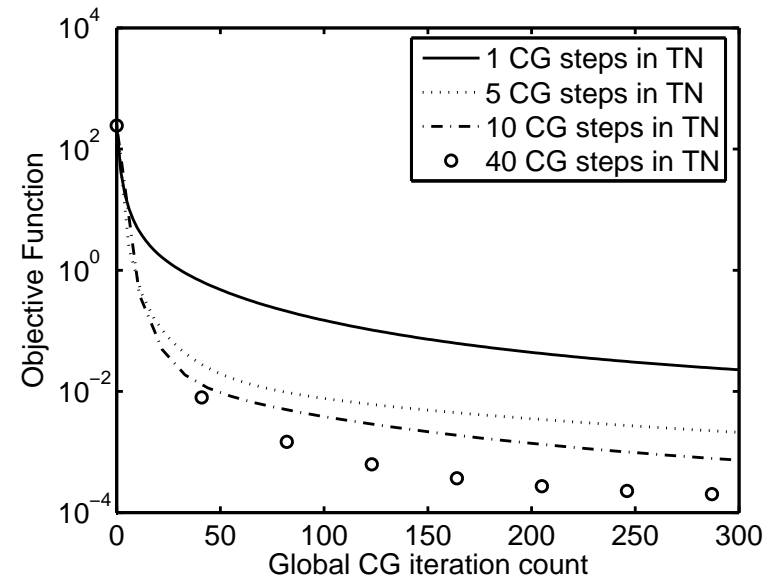
1. **TN step** Solve approximately Newton system $\nabla^2 f(x^k)d_{TN}^k = -\nabla f(x^k)$, i.e. minimize quadratic model $q_k(x)$ in (1), using l steps of CG.
2. **Subspace optimization step** $x^{k+1} \approx \arg \min_{x \in S_k} f(x)$:
affine subspace S_k passes through x^k and is spanned by:
 - * TN Direction $d_{TN}^k = x^{kl} - x^k$ (x^{kl} – last CG iterate)
 - * Last gradient of quadratic model $\nabla q_k(x^{kl})$ used in TN
 - * Last used CG direction in TN: $(x^{kl} - x^{k,l-1})$
 - * [Optionally] directions of several previous outer steps and gradients of f .
3. **Goto TN step**, while performing the first new CG step as an optimization of quadratic model $q_{k+1}(x)$ over 2D subspace spanned by $(x^{k+1} - x^{kl})$ and $\nabla f(x^{k+1})$.

Example: linear least squares, 400 variables

SESOP-TN

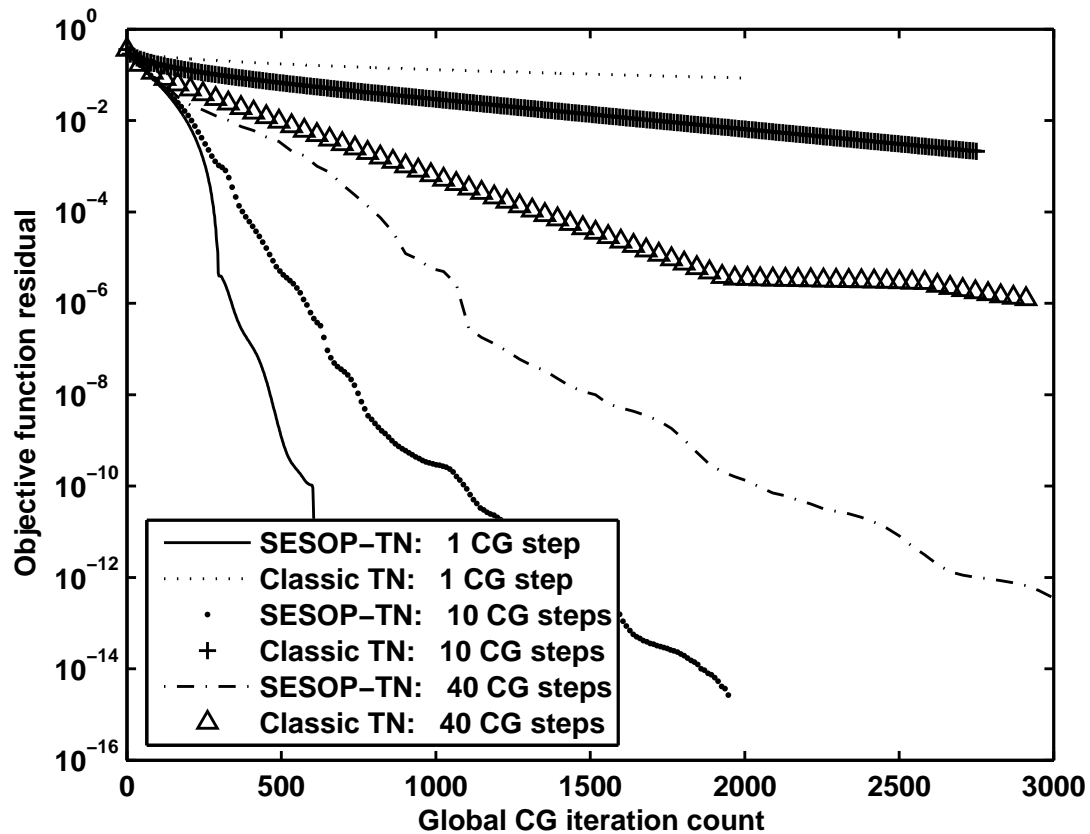


Standard TN



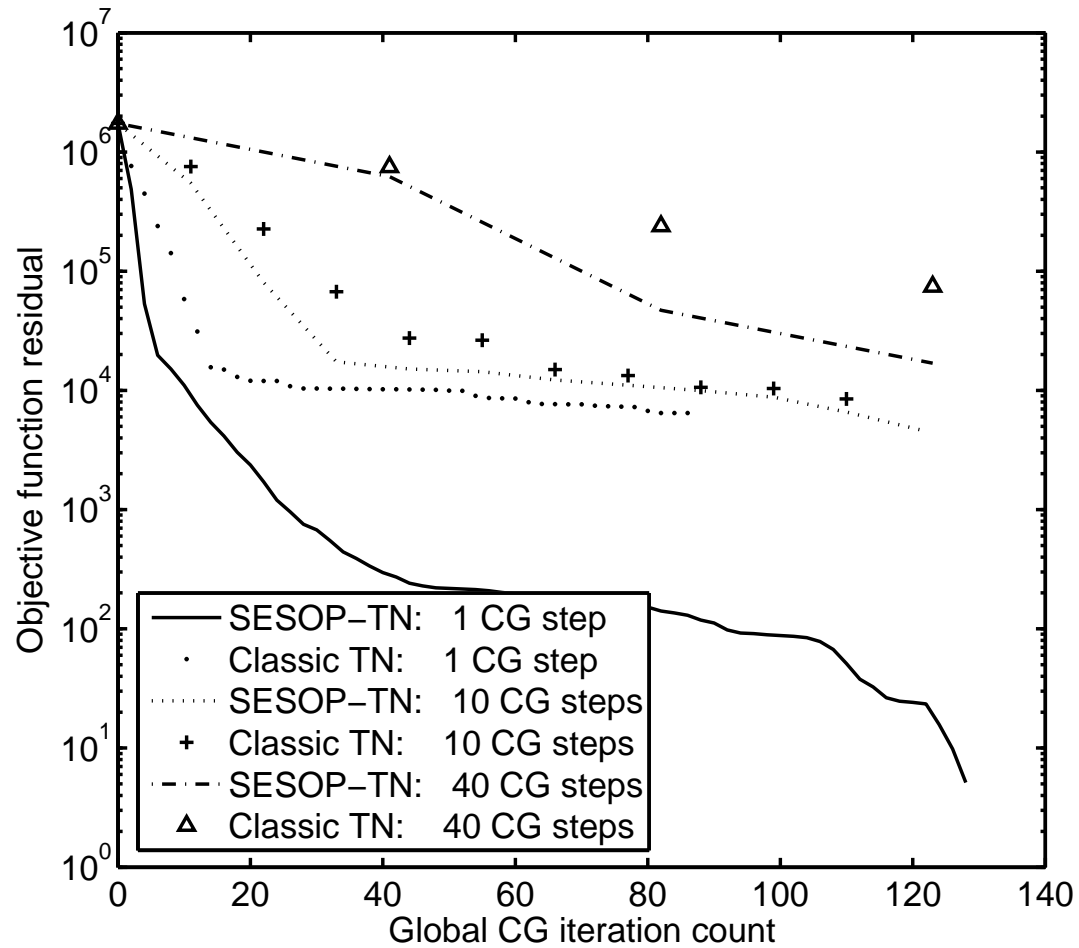
SESOP-TN trajectory does not depend on the number of CG iterations in TN step. Standard TN converges more slowly, when CG is truncated too early.

Exponents-and-Squares, 200 variables



Residual between objective and the optimal value versus CG iteration count.

Linear SVM, 99758 variables



The plots show the residual between the current objective and the optimal value versus CG iteration count.

Conclusions

- SESOP provides efficient framework for large-scale unconstrained optimization.
- Optimal worst-case complexity;
- Low memory and computational load per iteration;
- Can incorporate descent directions of other methods;
- Attractive as a building block for constrained optimization (Example: barrier LP or CP)