Subgradient methods in nonsmooth nonconvex optimization

Adil Bagirov

SITE, University of Ballarat, Victoria, Australia

The 5th China-Australia Workshop on Optimization: Theory, Methods and Applications, December 15-18, Shanghai, China

Joint work with:
L. Jin, A. Al Nuaimat, N. Sultanova (Ballarat, Australia)
and N. Karmitsa (Turku, Finland)

Outline

- Introduction
- Subgradient methods
- Quasisecants
- Subgradient method for nonsmooth nonconvex problems
- Numerical results
- Conclusions

Introduction

Our aim is to design an algorithm for solving the following unconstrained minimization problem:

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \mathbb{R}^n \tag{1}$$

where the objective function f is assumed to be locally Lipschitz and this function is not necessarily differentiable nor convex.

Introduction

- Subgradient methods, including the space dilation methods (Shor, Polyak).

- Bundle methods for nonconvex nonsmooth problems (Kiwiel, Lemarechal, Mifflin, Hare, Sagastizabal)

- Bundle methods based on splitting of the set of generalized gradients (Gaudioso and his co-authors: SIOPT, 2004, OMS, 2005, OMS 2009)

- Gradients sampling method (Lewis, Overton, Burke, SIOPT, 2005)

- Discrete gradient method (Bagirov et al, JOTA, 2008)

Subgradient methods

The subgradient method is known to be the simplest method for solving Problem (1). The convergence of this method was proved only for convex objective functions. The subgradient method uses only one subgradient evaluation at each of its iteration. Unlike most of other algorithms of nonsmooth optimization the subgradient method does not contain any subproblem for finding either search directions or step lengths. Although this method is not very efficient it is also well known that it can be more successful than the bundle methods for solving large scale problems.

Subgradient methods

Let $x^0 \in \mathbb{R}^n$ be any starting point. Then the subgradient method:

$$x^{k+1} = x^k - \alpha_k v^k, \;\; k = 0, 1, \ldots$$

where $v^k \in \partial f(x^k)$ is any subgradient and $\alpha_k > 0$ is a step-length.

1.
$$\alpha_k > 0, \sum_{k=0}^{\infty} \alpha_k = \infty, \sum_{k=0}^{\infty} \alpha_k^2 < \infty.$$

2. The step size α_k is chosen as a sufficiently small constant.

Polyak method:

$$\alpha_k = \frac{f(x^k) - f_*}{\|v^k\|^2}$$

This version of the subgradient method was successfully applied for solving constrained optimization problems using augmented Lagrangians (R. Gasimov, JOGO, 2002)

Quasisecants

The subdifferential $\partial f(x)$ of a locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$ is given by

$$\partial f(x) = \text{co} \left\{ \lim_{i \to \infty} \nabla f(x_i) \mid x_i \to x \text{ and } \nabla f(x_i) \text{ exists} \right\}.$$

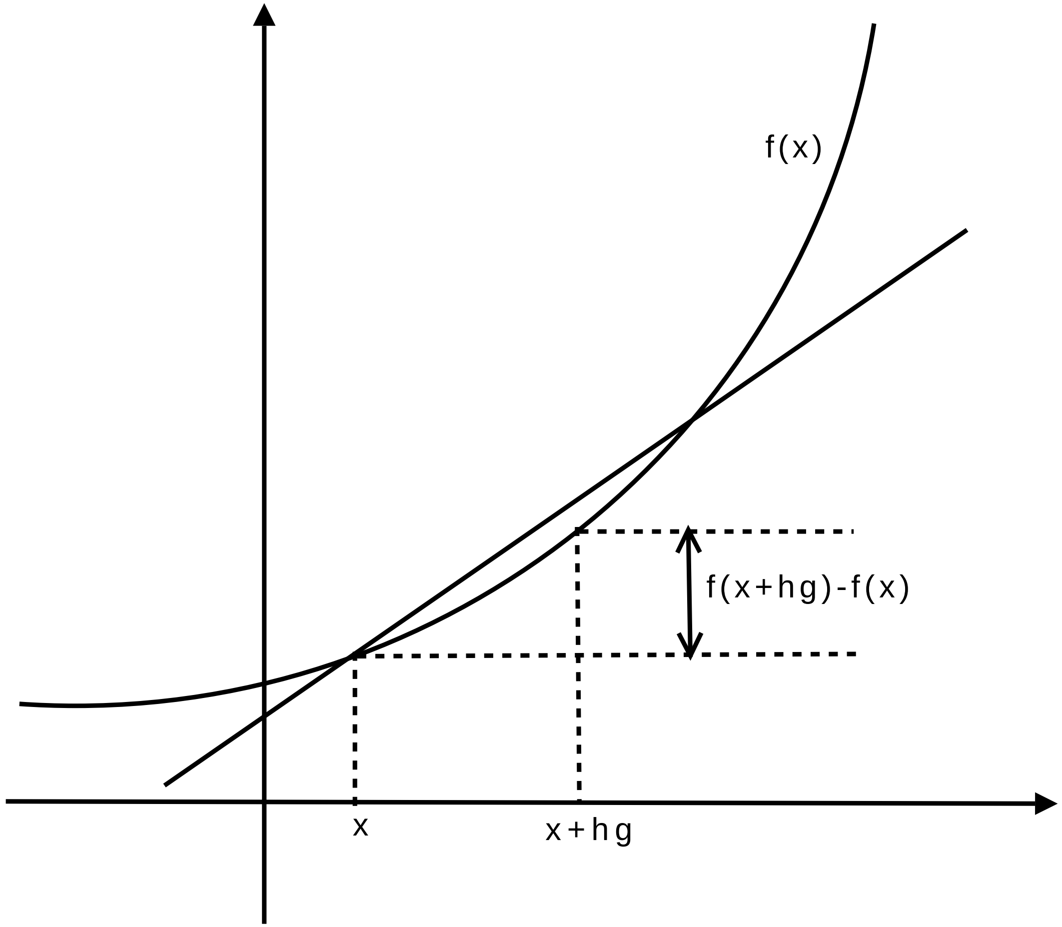
co denotes the convex hull of a set. $f'(x,g)$ is the directional and $f^0(x,g)$ is the generalized directional derivative of a function f at a point x in the direction $g \in \mathbb{R}^n$, respectively.

The notion of quasisecant was introduced in (A. Bagirov and A.N. Ganjehlou, OMS, 2010). It was demonstrated that quasisecants can be efficiently computed for many convex and nonconvex nonsmooth functions.

De nition 1 *A vector* $v \in \mathbb{R}^n$ *is called a quasisecant of the function* f *at the point* x *in the direction* $g \in S_1$ *with the length* $h > 0$ *iff*

$$f(x + hg) - f(x) \leq h\langle v, g\rangle. \tag{2}$$

Quasisecants



Quasisecants

We call the inequality (2) *a quasisecant inequality*. We will use the notation $v(x, g, h)$ for the quasisecant of the function f at the point x in the direction $g \in S_1$ with the length $h > 0$. It is clear that there are many vectors $v \in \mathbb{R}^n$ satisfying the quasisecant inequality for given $x \in \mathbb{R}^n$, $g \in S_1$ and $h > 0$. However, not all of them provide a local approximation for the function f. Such approximations can be provided by subgradients of the function f. Therefore, we require that each quasisecant not only satisfy the quasisecant inequality but also approximate subgradients of the function f. This leads us to the definition of the so-called subgradient-related (SR) quasisecants.

For given $x \in \mathbb{R}^n$ and $g \in S_1$ we consider the following set:

$$Q(x,g) = \{w \in \mathbb{R}^n : \quad \exists \{h_k > 0\} \text{ such that } \quad \lim_{k\to\infty} h_k = 0 \text{ and}$$

$$w = \lim_{k\to\infty} v(x,g,h_k)\}.$$

$Q(x,g)$ is a set of limit points of quasisecants $v(x,g,h)$ as $h \to +0$. These quasisecants are called SR if $Q(x,g) \subseteq \partial f(x)$. We will call such quasisecants SR-quasisecants.

Quasisecants

Now we construct the following set:

$$Q_0(x) = \text{cl} \bigcup_{g \in S_1} Q_S(x, g).$$

Here cl stands for a closure of a set. The set $Q_0(x)$ contains limit points of all SR-quasisecants at the point x.

Let $h > 0$ be a given number. At a point x consider the following set of all SR-quasisecants with the length no more than h:

$$Q_h(x) = \{u \in \mathbb{R}^n : \exists (g \in S_1,\ t \in (0, h]) : \quad u = v(x, g, t)\}.$$

Quasisecants

It follows from the definition of SR-quasisecants that for any $\varepsilon > 0$ there exists $\delta > 0$ such that

$$Q_h(y) \subset \partial f(x) + B_\varepsilon(0) \tag{3}$$

for all $h \in (0, \delta)$ and $y \in B_\delta(x)$.

Given $x \in \mathbb{R}^n$ and $h > 0$ we consider the following set:

$$W_h(x) = \bar{co} Q_h(x)$$

where $\bar{co}$ is a closed convex hull of a set. If $Q_h(x)$ is the SR-quasisecant mapping, then the set $W_h(x)$ is the approximation to the subdifferential $\partial f(x)$.

Quasisecants

Proposition 1 *The mapping* $x \mapsto Q_0(x)$ *has compact images for any* $x \in \mathbb{R}^n$.

Proposition 2 *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a locally Lipschitz function. Then*

$$\max_{w \in Q_0(x)} \langle w, g \rangle \leq f^0(x, g) \quad \forall g \in \mathbb{R}^n.$$

If the function f *is directionally differentiable then*

$$f'(x, g) \leq \max_{w \in Q_0(x)} \langle w, g \rangle \quad \forall g \in \mathbb{R}^n.$$

Corollary 1 *Suppose that the function f is locally Lipschitz and regular at x. Then*

$$\partial f(x) = \text{co}\, Q_0(x).$$

The quasisecant method was introduced in (A. Bagirov and A.N. Ganjehlou, OMS, 2010).

Subgradient method for nonsmooth nonconvex problems

Let $h > 0, c_1 > 0$ and $\delta > 0$ be given numbers.

Algorithm 1 Computation of the descent direction.

Step 1. Select any $d_1 \in S_1$ and compute a quasisecant $v_1 = v(x, d_1, h)$ with respect to this direction. Set $\tilde{v}_1 = v_1$ and $k = 1$.

Step 2. Solve the following problem:

$$\text{minimize} \quad \varphi_k(\lambda) \equiv \|\lambda v_k + (1-\lambda)\tilde{v}_k\|^2 \quad \text{subject to} \quad \lambda \in [0,1]. \qquad (4)$$

Let $\bar{\lambda}_k$ be a solution to this problem. Set

$$\bar{v}_k = \bar{\lambda}_k v_k + (1-\bar{\lambda}_k)\tilde{v}_k.$$

Step 3. (Stopping criterion) If

$$\|\bar{v}_k\| < \delta \qquad (5)$$

then stop. Otherwise go to Step 4.

Step 4. Compute the search direction by $\bar{d}_k = -\|\bar{v}_k\|^{-1}\bar{v}_k$.

Step 5. If

$$f(x + h\bar{d}_k) - f(x) \leq -c_1 h\|\bar{v}_k\|, \tag{6}$$

then stop. Otherwise go to Step 6.

Step 6. Compute a quasisecant $u = v(x, \bar{d}_k, h)$ in the direction $\bar{d}_k$. Set $v_{k+1} = u,\ \widetilde{v}_{k+1} = \bar{v}_k, k = k + 1$ and go to Step 2.

Proposition 3 *Suppose that f is a locally Lipschitz function, $h > 0$ and there exists $K, 0 < K < \infty$ such that*

$$\max\{\|v\| : v \in W_h(x)\} \le K. \tag{7}$$

If $c_1 \in (0,1)$ and $\delta \in (0,K)$, then Algorithm 1 stops after at most m steps, where

$$m \le 2\log_2(\delta/K)/\log_2 K_1 + 1, \quad K_1 = 1 - [(1-c_1)(2K)^{-1}\delta]^2.$$

Subgradient method for nonsmooth nonconvex problems

The use of subgradients: Let $c_1 > 0$ and $\delta > 0$ be given numbers.

Algorithm 2 Computation of the descent direction.

Step 1. Select any $d_1 \in S_1$ and compute a subgradient v_1 such that $f'(x, d_1) = \langle v_1, d_1 \rangle$. Set $\tilde{v}_1 = v_1$ and $k = 1$.

Step 2. Solve the following problem:

$$\text{minimize} \quad \varphi_k(\lambda) \equiv \|\lambda v_k + (1 - \lambda)\tilde{v}_k\|^2 \quad \text{subject to} \quad \lambda \in [0, 1]. \qquad (8)$$

Let $\bar{\lambda}_k$ be a solution to this problem. Set

$$\bar{v}_k = \bar{\lambda}_k v_k + (1 - \bar{\lambda}_k)\tilde{v}_k.$$

Step 3. (Stopping criterion) If

$$\|\bar{v}_k\| < \delta \tag{9}$$

then stop. Otherwise go to Step 4.

Step 4. Compute the search direction by $\bar{d}_k = -\|\bar{v}_k\|^{-1}\bar{v}_k$.

Step 5. If

$$f'(x, \bar{d}_k) \leq -c_1\|\bar{v}_k\|, \tag{10}$$

then stop. Otherwise go to Step 6.

Step 6. Compute a subgradient $u \in \partial f(x)$ such that $f'(x, \bar{d}_k) = \langle u, d_k \rangle$. Set $v_{k+1} = u,\ \widetilde{v}_{k+1} = \bar{v}_k, k = k + 1$ and go to Step 2.

Proposition 4 *Suppose that f is a locally Lipschitz function and there exists $K, 0 < K < \infty$ such that*

$$\max\{\|v\| : v \in \partial f(x)\} \leq K. \tag{11}$$

If $c_1 \in (0,1)$ and $\delta \in (0,K)$, then Algorithm 1 stops after at most m steps, where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 1, \quad K_1 = 1 - [(1-c_1)(2K)^{-1}\delta]^2.$$

A point x is called the (h, δ)-stationary point for Problem (1) iff:

$$\min\{\|v\| : \; v \in W_h(x)\} \leq \delta.$$

Algorithm 3 Computation of the (h, δ)-stationary points.

Step 0. **Input data:** starting point $x_0 \in \mathbf{R}^n$, numbers $h > 0, \; \delta > 0, \; c_1 \in (0, 1), \; c_2 \in (0, c_1]$.

Step 1. Set $k = 0$.

Step 2. Apply Algorithm 1 for the computation of the descent direction at $x = x_k$ for given $\delta > 0$ and $c_1 \in (0, 1)$. This algorithm terminates after finite number of iterations and as a result, we get the quasisecant v_k, the aggregated quasisecant $\widetilde{v}_k$ and an element $\bar{v}_k$ such that $\bar{v}_k = \bar{\lambda}_k v_k + (1 - \bar{\lambda}_k)\widetilde{v}_k$ where

$$\bar{\lambda}_k = \operatorname*{argmin}_{\lambda \in [0,1]} \|\lambda v_k + (1-\lambda)\widetilde{v}_k\|^2.$$

Further, either $\|\bar{v}_k\| < \delta$ or for the search direction $\bar{d}_k = -\|\bar{v}_k\|^{-1}\bar{v}_k$,

$$f(x_k + h\bar{d}_k) - f(x_k) \leq -c_1 h\|\bar{v}_k\| \tag{12}$$

Step 3. If $\|\bar{v}_k\| < \delta$ then stop. Otherwise go to Step 4.

Step 4. Compute $x_{k+1} = x_k + \sigma_k \bar{d}_k$, where σ_k is defined as follows

$$\sigma_k = \operatorname{argmax}\left\{\sigma \geq 0 : \ f(x_k + \sigma\bar{d}_k) - f(x_k) \leq -c_2\sigma\|\bar{v}_k\|\right\}.$$

Set $k = k + 1$ and go to Step 2.

Proposition 5 *Suppose that function* f *is bounded below*

$$f_* = \inf \ \{f(x) : x \in \mathbf{R}^n\} > -\infty. \tag{13}$$

Then Algorithm 3 terminates after finite many iterations $M > 0$ *and produces* (h, δ)*-stationary point* x_M *where*

$$M \leq M_0 \equiv \left\lfloor \frac{f(x_0) - f_*}{c_2 h \delta} \right\rfloor + 1. \tag{14}$$

Algorithm 4 Computation of stationary points.

Step 1. Select sequences $\{h_j\}, \{\delta_j\}$ such that $h_j > 0, \;\; \delta_j > 0$ and $h_j \to 0, \;\; \delta_j \to 0$ as $j \to \infty$. Choose any starting point $x_0 \in \mathbf{R}^n$, and set $k = 0$.

Step 2. If $h_k \leq \varepsilon$ and $\delta_k \leq \varepsilon$, then stop with x_k as the final solution.

Step 3. Apply Algorithm 3 starting from the point x_k with $h = h_k$ and $\delta = \delta_k$. This algorithm finds an (h_k, δ_k)-stationary point x_{k+1} after finite many iterations $M > 0$.

Step 4. Set $k = k + 1$ and go to Step 2.

For point $x_0 \in \mathbf{R}^n$, we consider the level set
$\mathcal{L}(x_0) = \{x \in \mathbf{R}^n : \ f(x) \leq f(x^0)\}$.

Proposition 6 *Suppose that the objective function f in Problem (1) is locally Liptchitz and the set $\mathcal{L}(x_0)$ is bounded for the starting point x_0. Then any accumulation point of the sequence $\{x_k\}$ generated by Algorithm 4 is stationary point of Problem (1).*

Numerical results

The proposed subgradient method (SUNNOPT - SUbgradient method for Nonsmooth Nonconvex OPTimization).

Test problems

- Unconstrained minimax problems (L. Lukśan and J. Vlćek, Test problems for nonsmooth unconstrained and linearly constrained optimization, Technical report No. 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, January 2000.);

- General nonsmooth unconstrained problems;

- Large scale problems (M. Haarala, K. Miettinen and M. Mäkelä, New limited memory bundle method for large-scale nonsmooth optimization. *Optimization Methods and Software* 19(6), 2004, 673–692.).

Numerical results

We use the following implementations of the subgradient method:

1. SUB1: in this version we take $\alpha_k = 1/k$, however after each 25000 iterations we update it to improve the convergence of the algorithm. Let p_k is the largest integer, smaller than or equal to $k/25000$. Then
$$\alpha_k = \frac{1}{k - 25000p_k}.$$
2. SUB2: in this version the step-length α_k is to some extent constant. We take $\alpha_k = 0.005$ for the first 10000 iterations, and $\alpha_k = 0.0001$ for all other iterations.

Numerical results

The subgradient method does not have stopping criterion based on necessary conditions. Therefore we stop the algorithm when the number of function evaluations reaches 10^6 and also the algorithm stops if it cannot decrease the value objective function with respect to some tolerance in 5000 successive iterations.

Parameters in the SUNNOPT were chosen as follows:

$c_1 = 0.2, \; c_2 = 0.05, \; h_{k+1} = 0.5h_k, \; k \geq 1, h_1 = 1, \delta_k \equiv 10^{-7}, \quad \forall k \geq 0.$

Numerical results

We used 20 random starting points for each problem and starting points are the same for all four algorithms. We say that an algorithm solves nonconvex nonsmooth optimization problem if it finds its local minimizer even if this local minimizer is different from the global one. An algorithm finds a solution to a problem with a tolerance $\varepsilon > 0$ if

$$|\bar{f} - f_{local}| \leq \varepsilon(1 + |f_{local}|).$$

Here f_{local} is the value of the objective at one of its known local minimizers and $\bar{f}$ is the value of an objective function at the solution found by an algorithm, respectively.

Numerical results

We implemented the SUNNOPT and subgradient methods SUB1 and SUB2 in Lahey Fortran 95 and used Fortran implementation of PBUN. Numerical experiments were carried out on PC Intel(R)Core(TM)2 with CPU 1.86 GHz and 1.97GB of RAM.

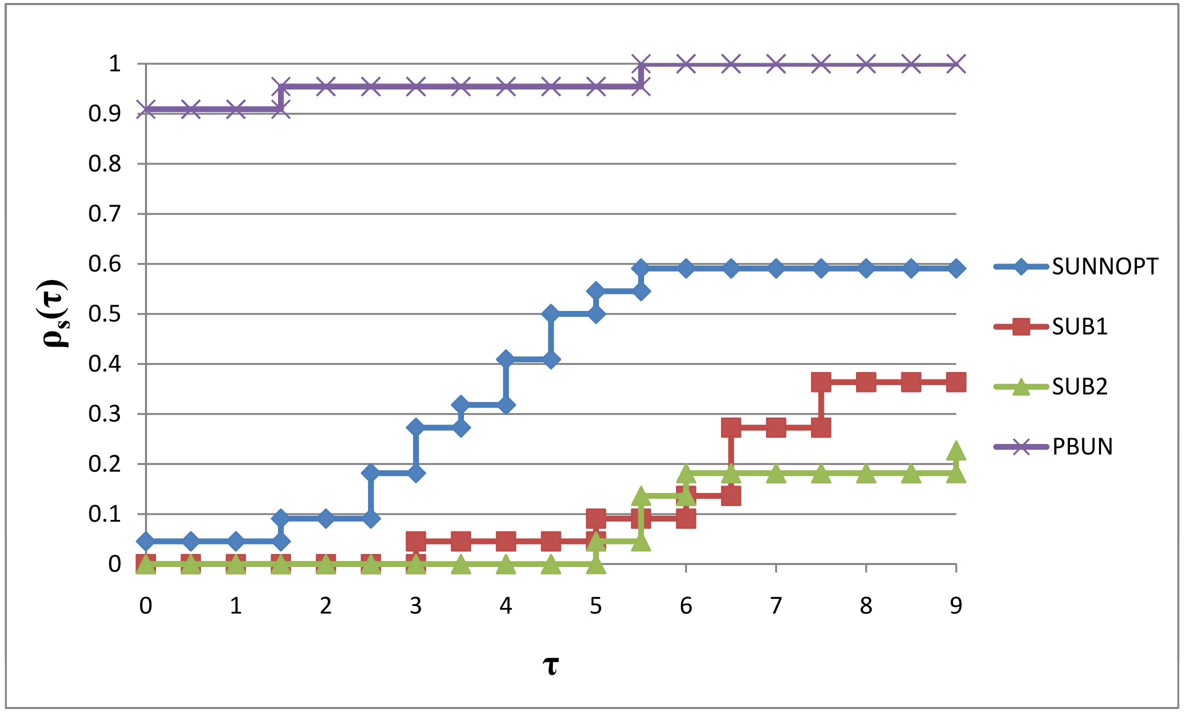
We analyze the results using the performance profiles introduced in (E.D. Dolan and J.J. More, Benchmarking optimization software with performance profiles. *Mathematical Programming* 91, 2002, 201–213.). Since the ratio of the number of function and subgradient evaluations by the subgradient methods and other two algorithms (the SUNNOPT and PBUN) is too large we scale this ratio using natural logarithm.

Numerical results

In the performance profiles, the value of $\rho_s(\tau)$ at $\tau = 0$ gives the percentage of test problems for which the corresponding algorithm is the best (it uses least computational time or function and subgradient evaluations) and the value of $\rho_s(\tau)$ at the rightmost abscissa gives the percentage of test problems that the corresponding algorithm can solve, that is, the robustness of the algorithm (this does not depend on the measured performance). Moreover, the relative efficiency of each algorithm can be directly seen from the performance profiles: the higher the particular curve, the better the corresponding algorithm.

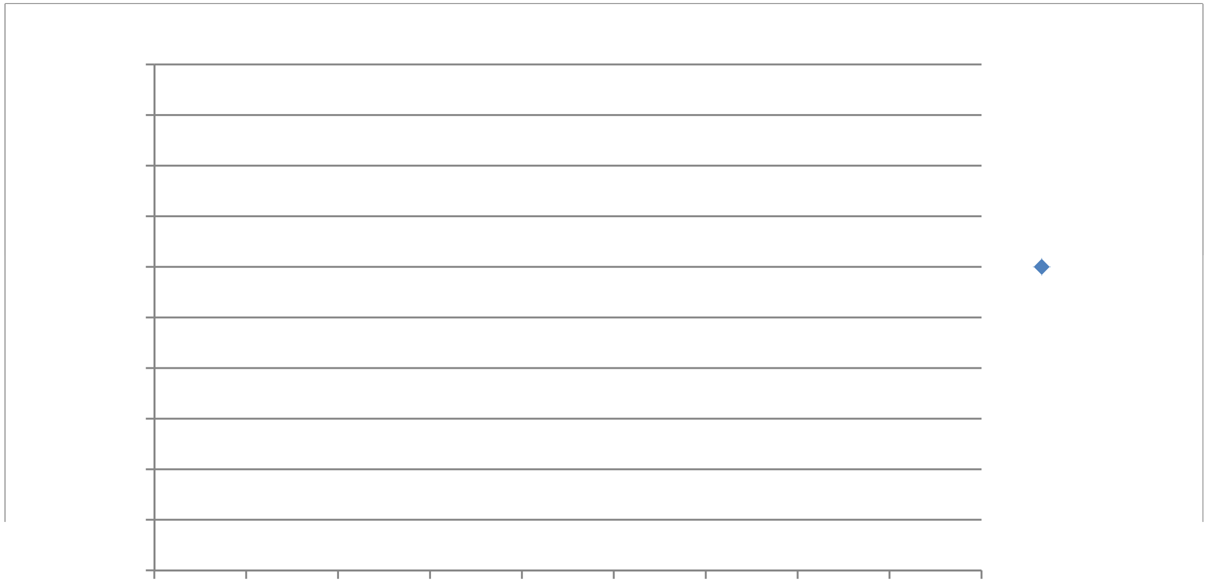
Numerical results

Unconstrained minimax problems (function evaluations, with single starting point):



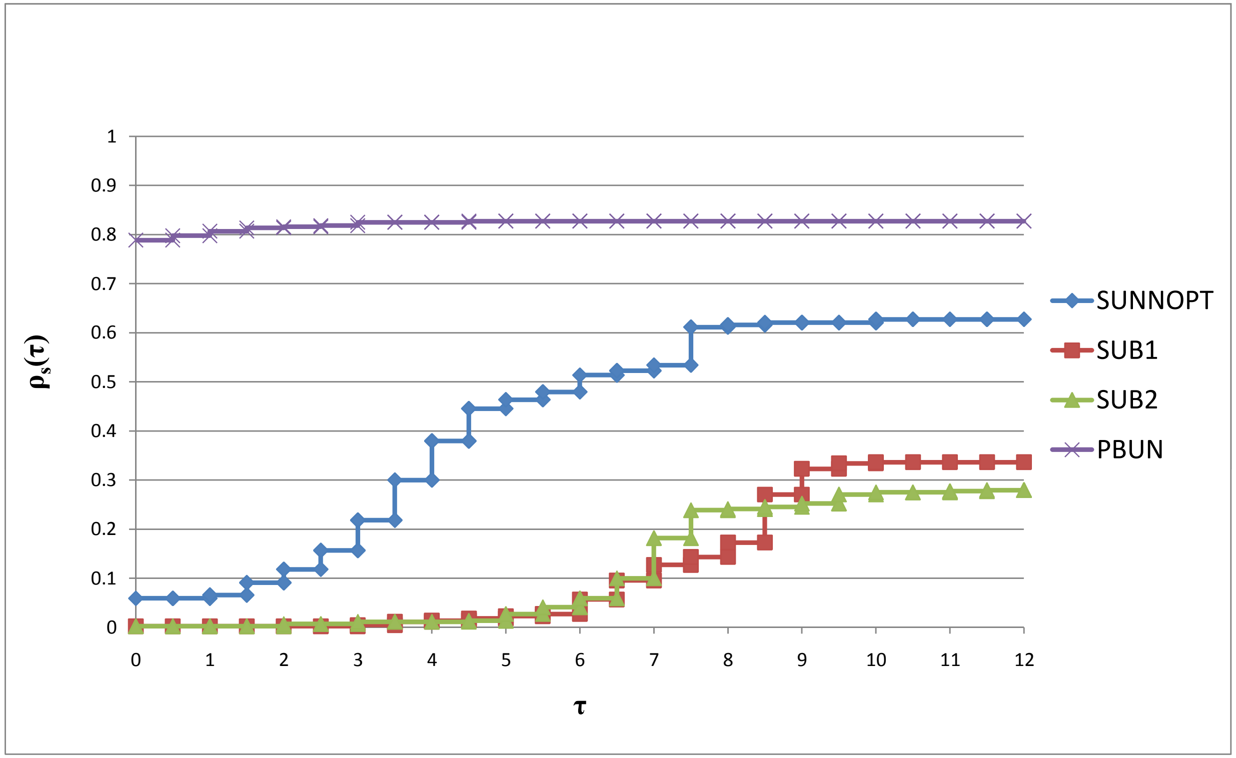
Numerical results

Unconstrained minimax problems (subgradient evaluations, with single starting point):



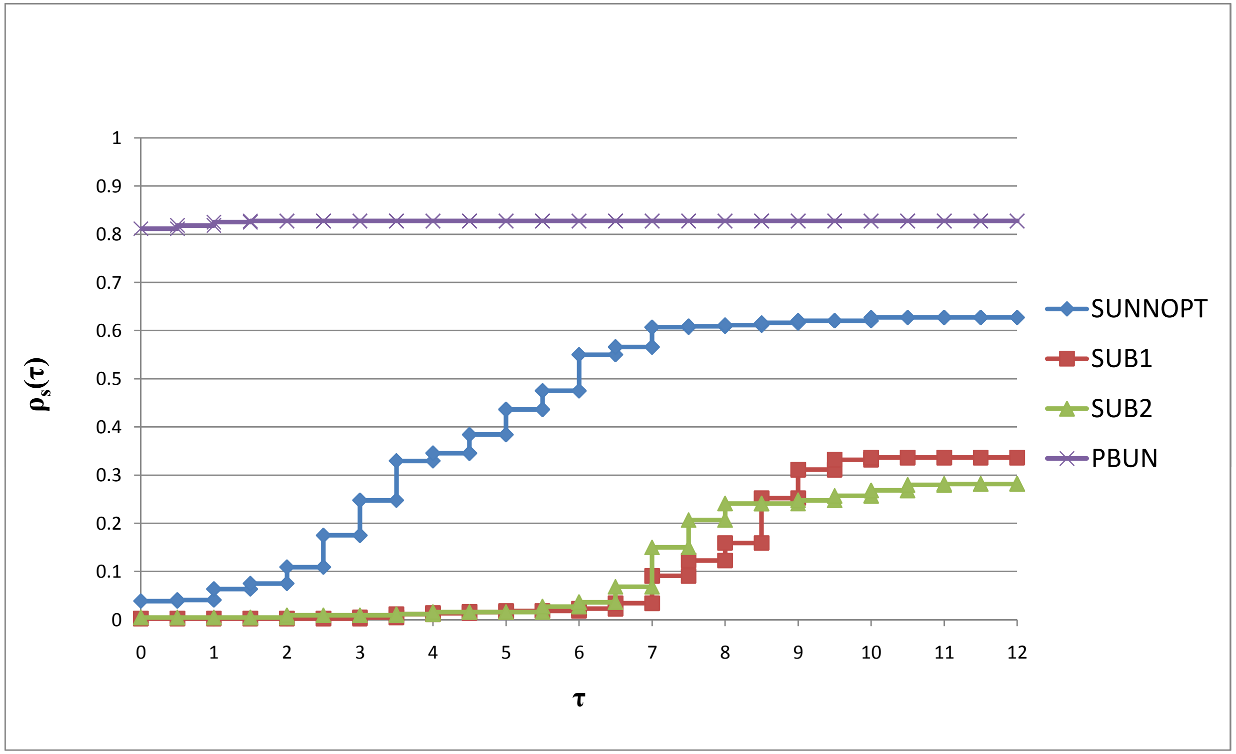
Numerical results

Unconstrained minimax problems (function evaluations, 20 random starting points):



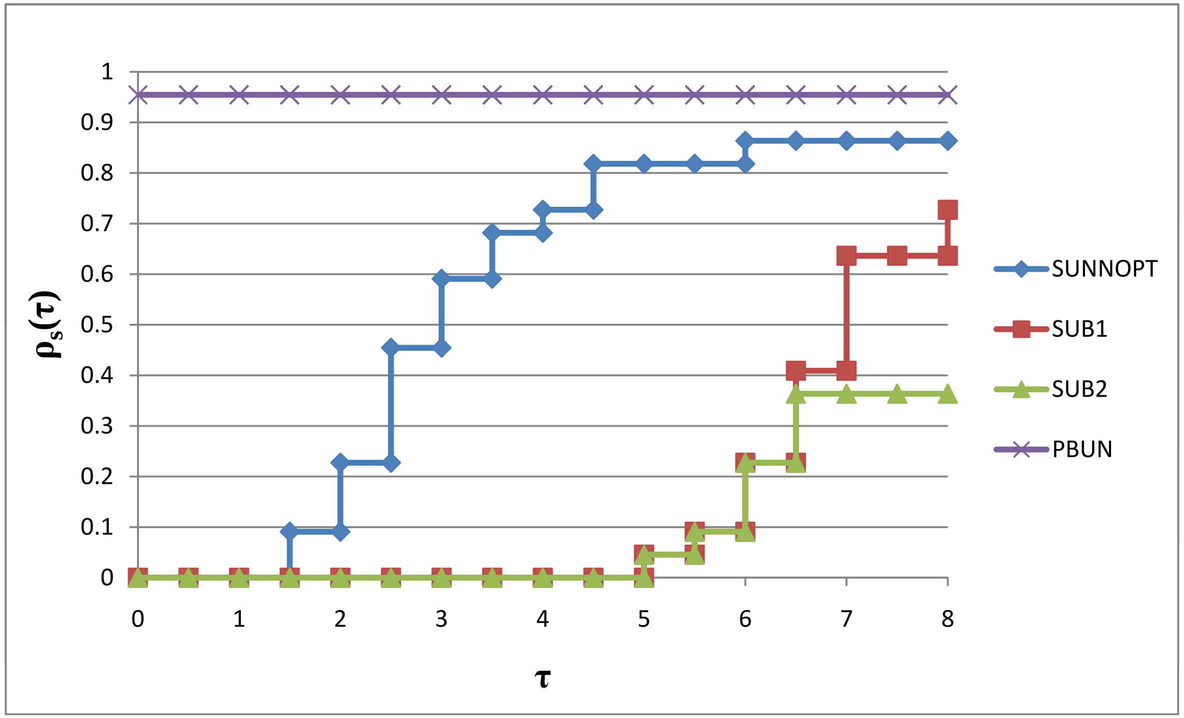
Numerical results

Unconstrained minimax problems (subgradient evaluations, 20 random starting points):



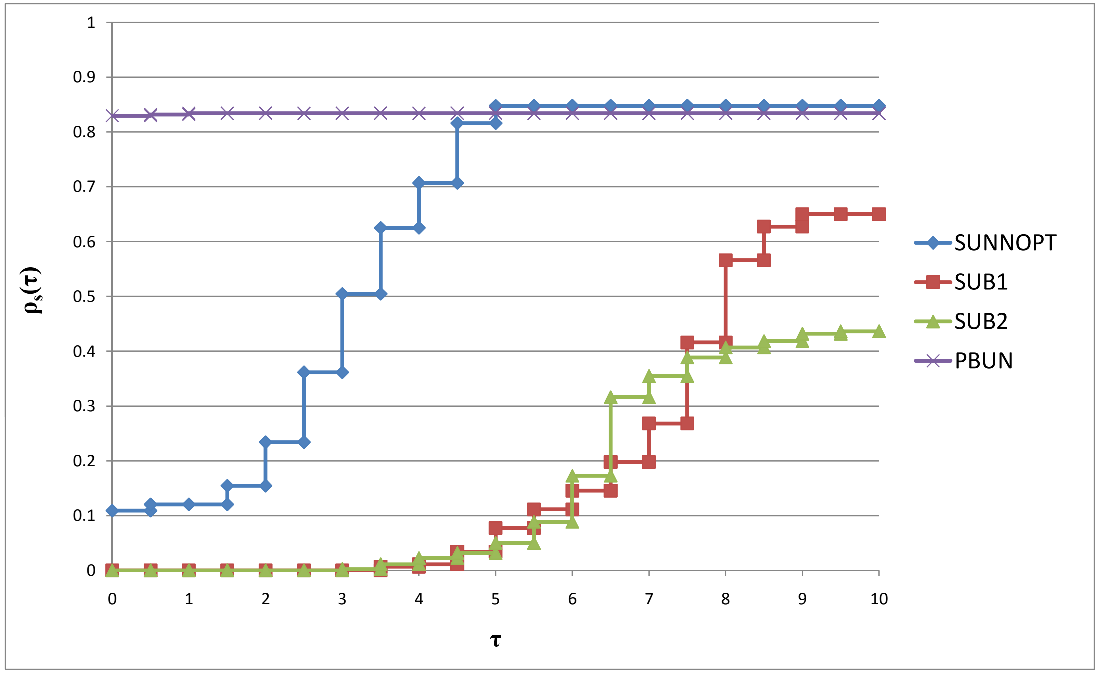
Numerical results

General nonsmooth unconstrained problems (function evaluations, single starting point):

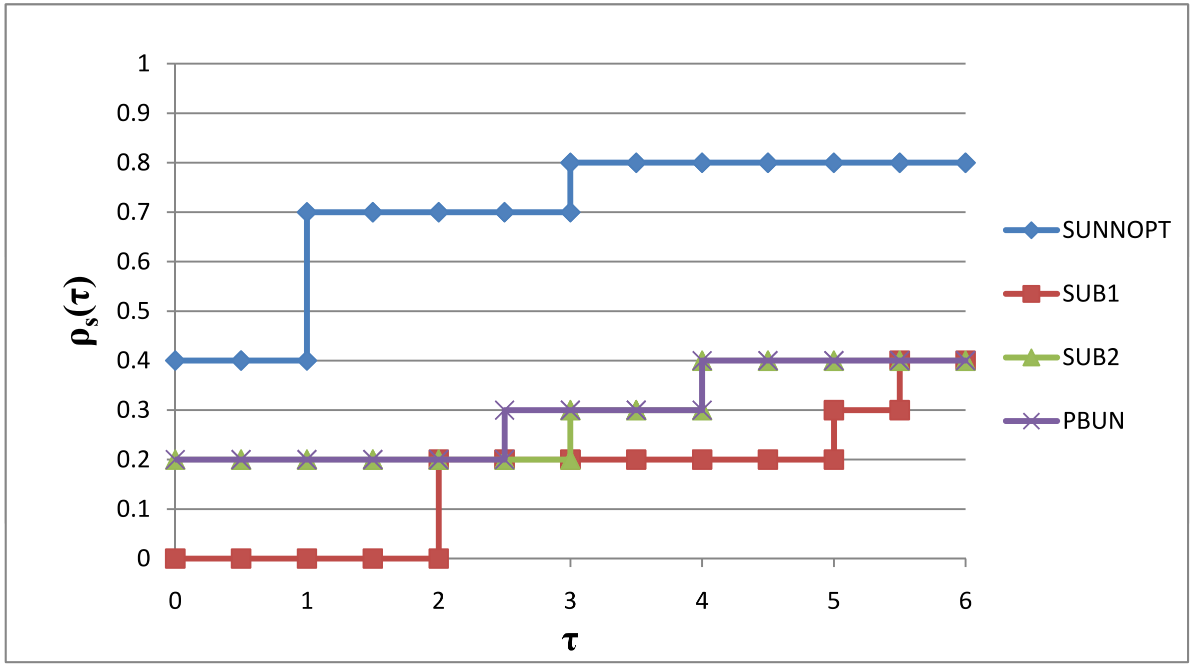


Numerical results

General nonsmooth unconstrained problems (function evaluations, 20 random starting points):

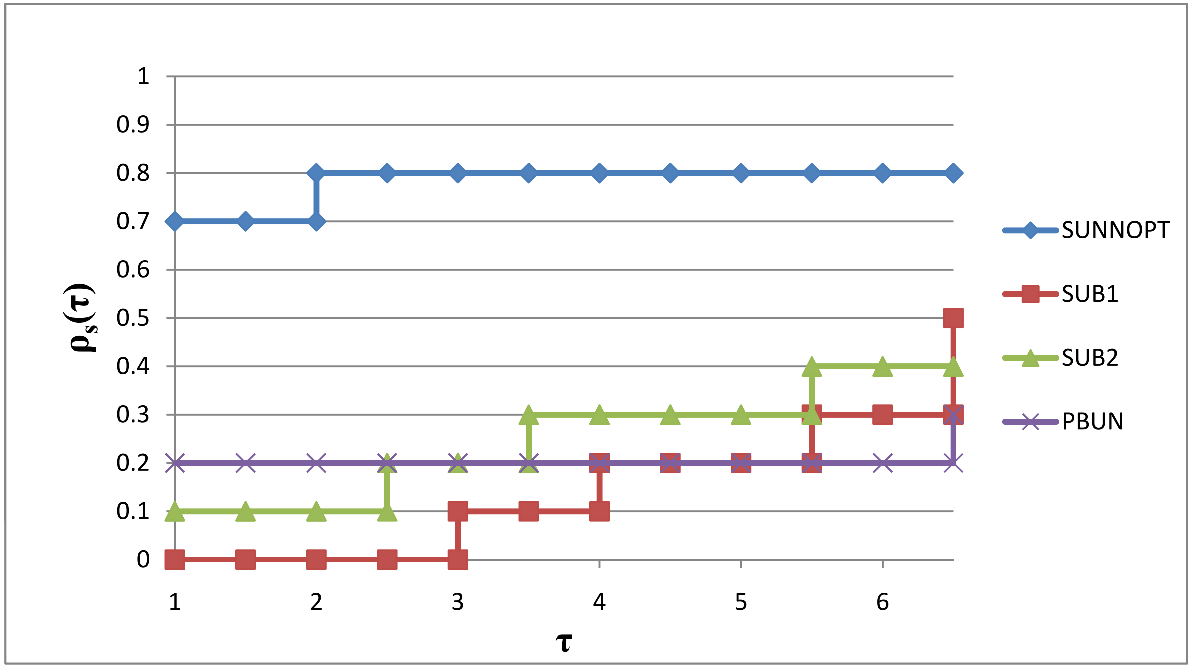


Large scale problems (CPU time, 500 variables, single starting point):



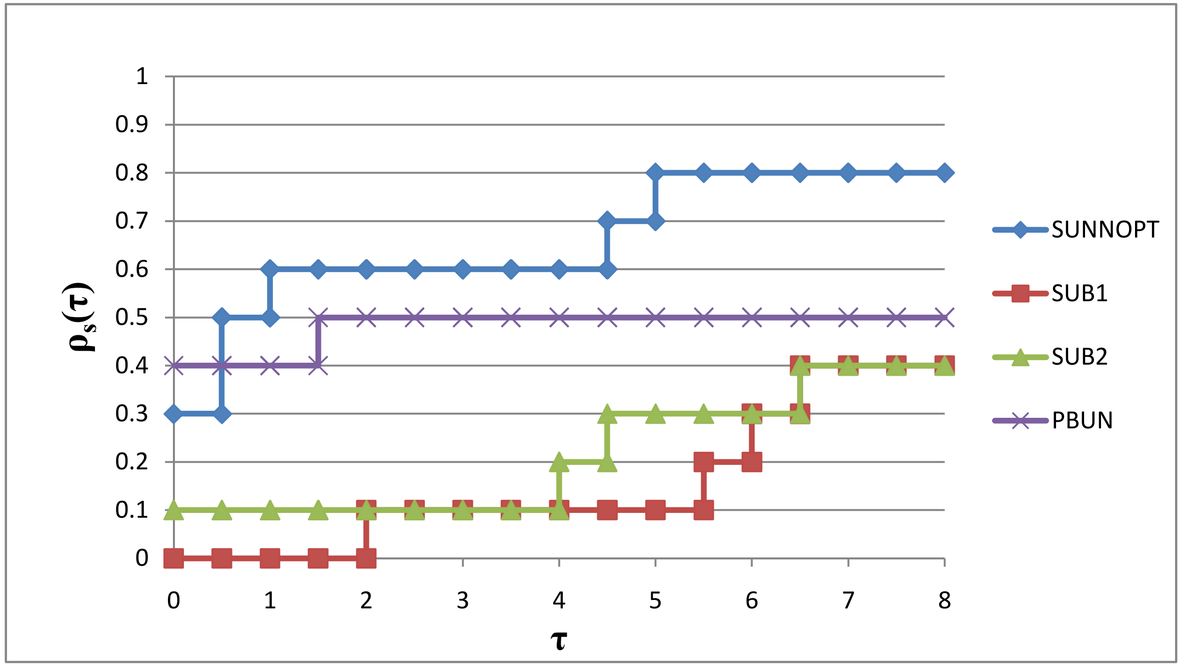
Numerical results

Large scale problems (CPU time, 1000 variables, single starting point):



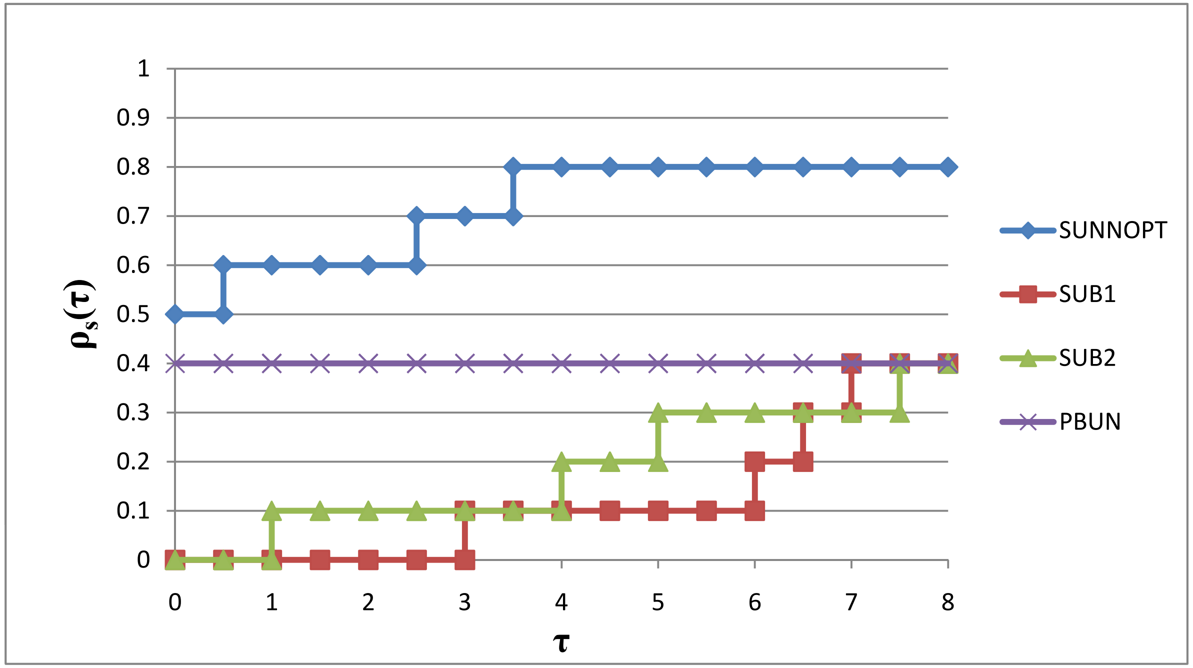
Numerical results

Large scale problems (function evaluations, 500 variables, single starting point):



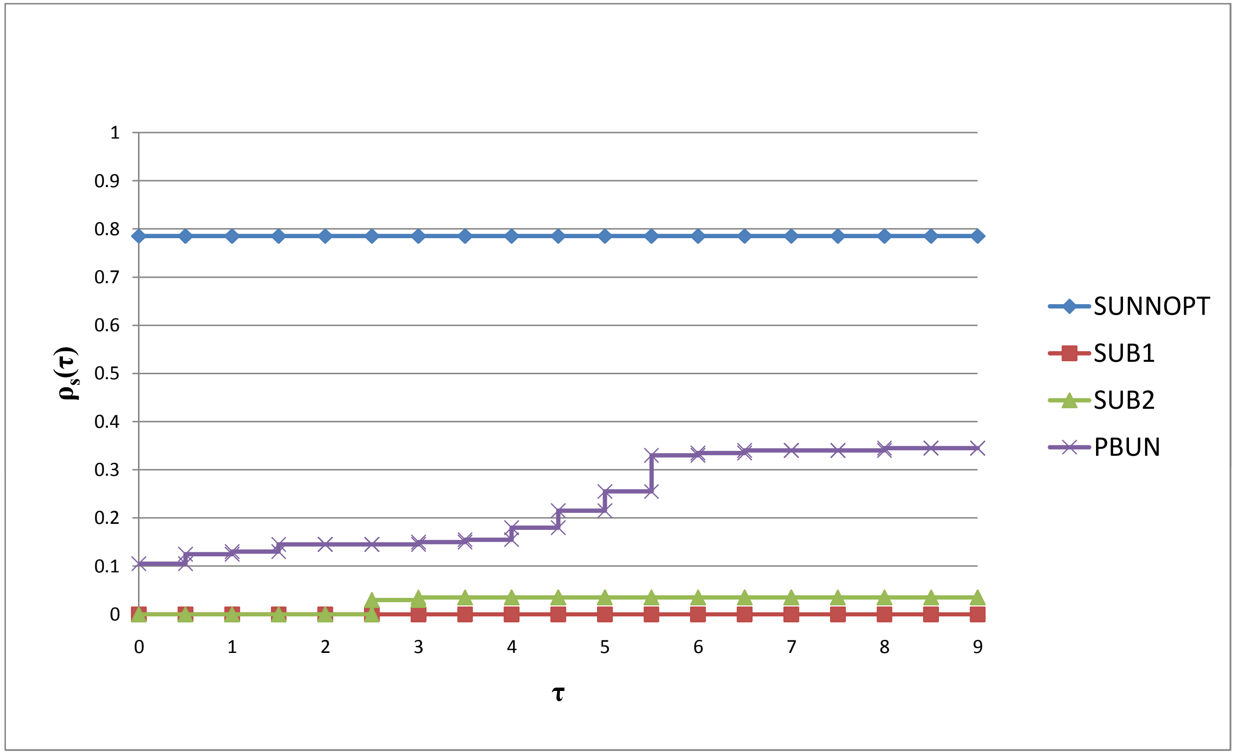
Numerical results

Large scale problems (function evaluations, 1000 variables, single starting point):



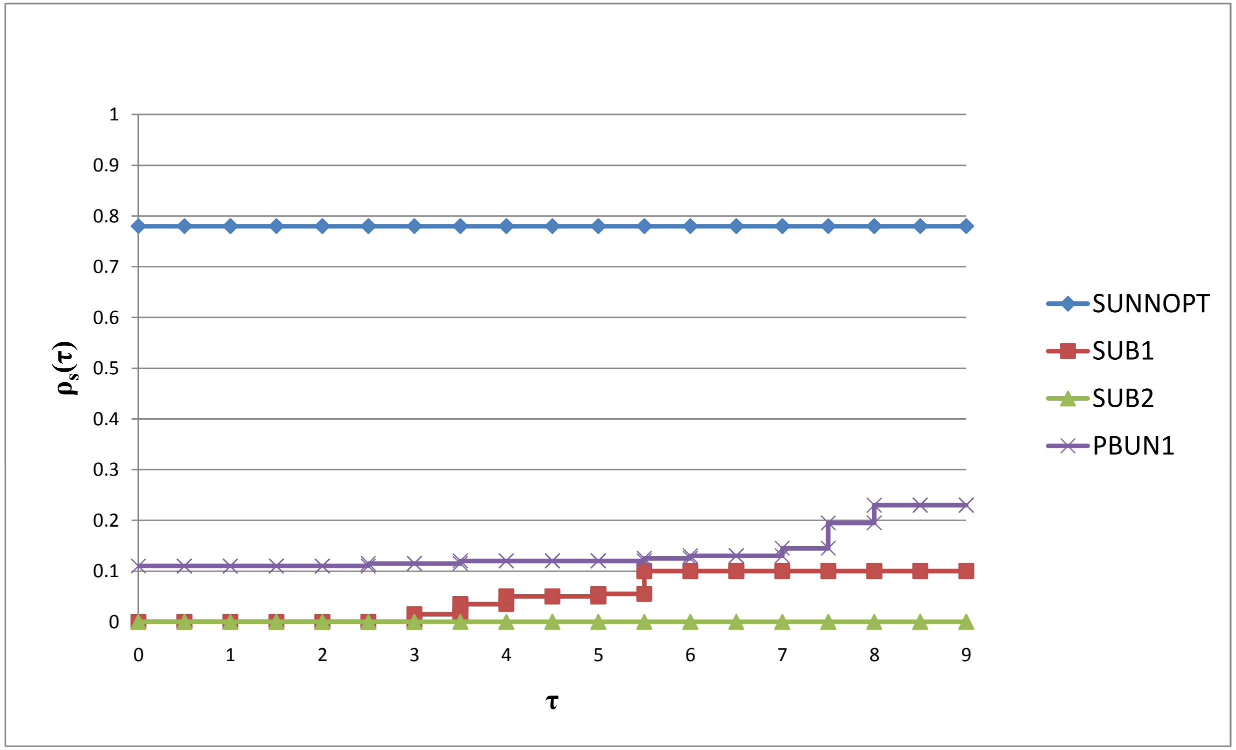
Numerical results

Large scale problems (CPU time, 500 variables, 20 random starting points):



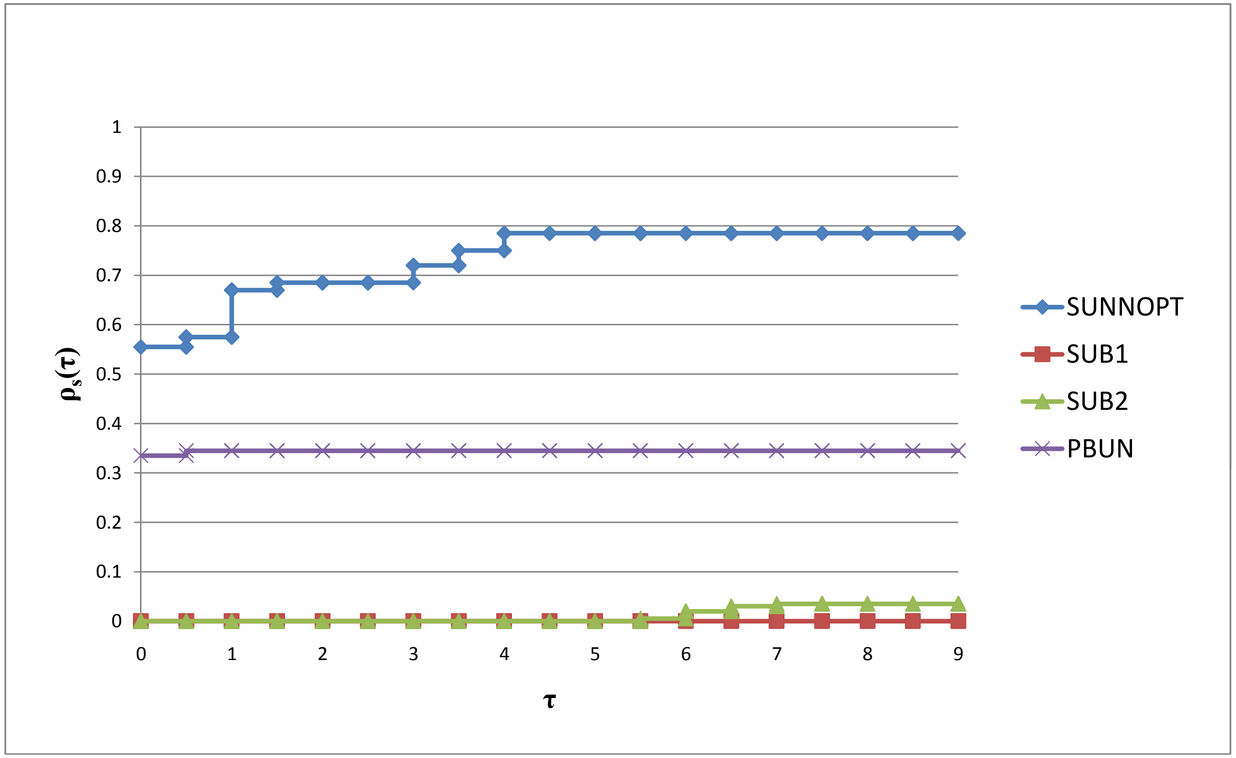
Numerical results

Large scale problems (CPU time, 1000 variables, 20 random starting points):



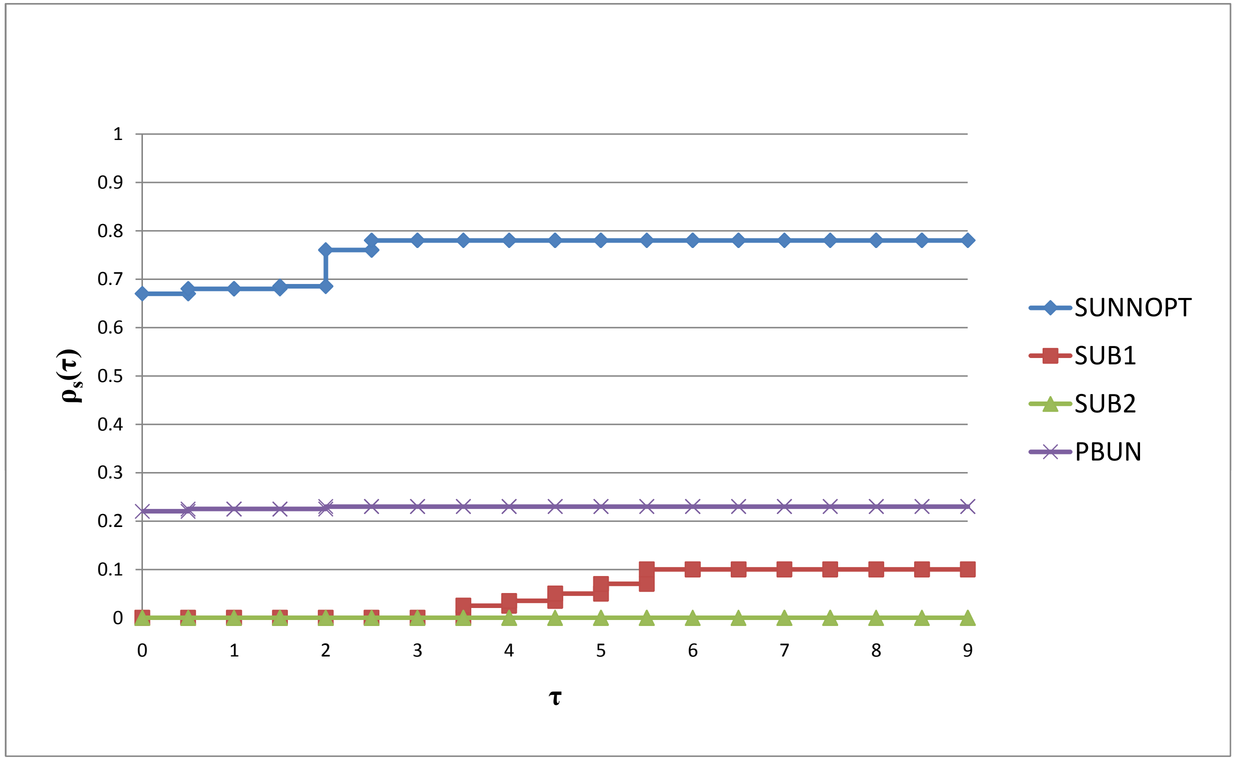
Numerical results

Large scale problems (function evaluations, 500 variables, 20 random starting points):



Numerical results

Large scale problems (function evaluations, 1000 variables, 20 random starting points):



Conclusion

- We developed a version of the subgradient for general nonsmooth and in particular nonconvex optimization problems.

- We proved its convergence for a broad class of nonsmooth nonconvex optimization problems.

- The proposed algorithm is easy to implement.

- Numerical results clearly demonstrate that the proposed algorithm is a significant improvement of the subgradient method.

- It outperforms the proximal bundle method for large scale problems.