

Nonsmooth spectral gradient methods for unconstrained optimization

Milagros Loreto* Hugo Aponte† Debora Cores‡ Marcos Raydan §

November 8, 2015

Abstract

To solve nonsmooth unconstrained minimization problems, we combine the spectral choice of step length with two well-established subdifferential-type schemes: the gradient sampling method and the simplex gradient method. We focus on the interesting case in which the objective function is continuously differentiable almost everywhere, and it is often not differentiable at minimizers. In the case of the gradient sampling method, we also present a simple differentiability test that allows us to use the exact gradient direction as frequently as possible, and to build a stochastic subdifferential direction only if the test fails. The proposed spectral gradient sampling method is combined with a monotone line search globalization strategy. On the other hand, the simplex gradient method is a direct search method that only requires function evaluations to build an approximation to the gradient direction. In this case, the proposed spectral simplex gradient method is combined with a suitable nonmonotone line search strategy. For both scenarios, we present preliminary numerical results on a set of nonsmooth test functions. These numerical results indicate that using a spectral step length can improve the practical performance of both methods.

AMS Subject Classification: 49M05, 90C56, 90C26, 65K10.

Keywords: Nonsmooth optimization, sampling gradients, simplex gradients, spectral gradient method, derivative free optimization, nonmonotone line search.

1 Introduction

We consider the nonsmooth unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is locally Lipschitz continuous in $\mathbb{R}^n$ and differentiable on an open dense subset $W \subset \mathbb{R}^n$. We focus our attention on the practical case in which f is continuously differentiable almost everywhere, but it is often not differentiable at minimizers.

*School of Science, Technology, Engineering, and Mathematics (STEM), University of Washington Bothell, 18115 Campus Way NE, Bothell, WA 90011-8246, USA (mloreto@uw.edu).

†Senior Software Development Engineer at Microsoft, Seattle, WA, USA (haponte@gmail.com).

‡Departamento de C  mputo Cient  fico y Estad  stica, Universidad Sim  n Bol  var, Ap. 89000, Caracas 1080-A, Venezuela (cores@usb.ve). Partially supported by CESMA at USB.

  Departamento de C  mputo Cient  fico y Estad  stica, Universidad Sim  n Bol  var, Ap. 89000, Caracas 1080-A, Venezuela (mraydan@usb.ve). Partially supported by CCCT Center at UCV.

For solving (1), the gradient sampling method introduced and analyzed by Burke *et al.* [5, 6] has emerged in the last decade as a robust and practical option. Roughly speaking, at a current iterate x_k the method computes the user-supplied gradient of the function f on a set of randomly generated points, in a suitable neighborhood of x_k , to build a descent direction. The next iterate x_{k+1} is then obtained performing a standard monotone line search along the stochastically computed direction. Several variations and extensions of the gradient sampling method have been recently proposed; see, e.g., [10, 18, 20]. We would like to emphasize that the direction built by the gradient sampling method can be viewed as a Clarke-Generalized Subdifferential; see [6, 7].

Another approach for solving (1), that requires no derivative information, is based on the simplex gradient method originally proposed by Bortz and Kelly [4], and further analyzed and enhanced by Custodio *et al.* [11, 12]. We concentrate our attention in the SID-PSM method (simplex derivative - pattern search method) fully described and analyzed in [11]. Roughly speaking, at each iteration k the method uses some previously computed and some new evaluations of the objective function at $n + 1$ conveniently chosen points (vertices of a simplex), solves a linear system that comes from imposing a set of linear polynomial interpolation conditions to obtain the simplex gradient vector, which is then properly used to obtain a better iterate x_{k+1} such that $f(x_{k+1})$ guarantees sufficient decrease with respect to $f(x_k)$. The simplex gradient built by the SID-PSM method is also related to the Clarke-generalized subdifferential; see [11]. For additional properties of simplex gradients see [25], and for a combination of simplex gradient methods with a nonmonotone line search strategy see [17]. The SID-PSM method has been successfully applied to solve nonsmooth optimization problems emerging from practical applications for which derivatives are not available; see [8].

The purpose of our work is to combine the spectral choice of step length with subdifferential directions built either by the gradient sampling method or by the SID-PSM method. The spectral step length, originally proposed by Barzilai and Borwein [1], analyzed by Raydan for convex quadratics [23], and later extended to nonquadratic unconstrained minimization problems [24], is remarkably superior to the classical steepest descent method; see, e.g., [2, 3, 14] and references therein. This choice of step length, which is related to the quasi-Newton family of methods through an approximated secant equation, requires little computational work and can be easily adapted to different optimization scenarios. Our main motivation for extending the spectral approach to the nonsmooth case is the wide variety of practical problems for which the smooth spectral versions have been successfully applied; see [2, 3].

It is worth mentioning that Loreto *et al.* [9] developed the so-called spectral projected subgradient method for the minimization of non-differentiable functions on convex sets, specifically solving the Lagrangean dual problem that appears when solving integer programming problems, for which subgradients could be obtained in a rather natural way. They presented encouraging numerical results on two different type of problems: set covering problems and generalized assignment problems. Therefore, our work can also be viewed as an generalization of the work presented in [9], for which some convergence properties were later analyzed by Loreto and Crema [22].

This paper is organized as follows. In Section 2 we present the nonsmooth spectral approach. In Sections 3 we describe the combination of the spectral approach with the gradient sampling method, including a simple differentiability test. In Section 4 we describe the combination of the spectral approach with the simplex gradient used in derivative-free methods. In Section 5 we present numerical results to give further insight into the nonsmooth spectral approach.

2 Nonsmooth spectral approach

Let us consider the nonsmooth optimization problem (1), and let us assume that at every iterate x_k we obtain a subdifferential vector, denoted by g_k , using the same subdifferential-type scheme. In this work, to compute g_k , we will use either the gradient sampling method for every k , or the SID-PSM method for every k . Nevertheless, we are proposing a general family of methods called nonsmooth spectral gradient methods (NSG), for which an iteration to move from x_k to x_{k+1} can be described as follows:

1. Compute a subdifferential g_k
2. Set $d_k = -\alpha_k g_k$, where α_k is the spectral step length
3. Set $x_+ = x_k + d_k$ and $\tau = 1$
4. While a line search condition is not satisfied reduce τ (backtracking), and set $x_+ = x_k + \tau d_k$.
5. Set $x_{k+1} = x_+$, and compute g_{k+1} and α_{k+1} .

The structure of each iteration of the NSG method is very attractive because of its simplicity and low memory requirements. The search direction is $-g_k$, the backtracking procedure only requires a few additional function evaluations per iteration, and the calculation of the spectral step length requires a low computational cost. Indeed, in the smooth case, the basic idea to obtain the spectral step length is to regard the matrix $(1/\alpha_k)I$ as an approximation to the Hessian $\nabla^2 f(x_k)$ and impose a quasi-Newton secant equation

$$\frac{1}{\alpha_k} s_{k-1} = y_{k-1},$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$. In general, this equation cannot be solved. However, accepting the least-squares solution that minimizes $\|\frac{1}{\alpha_k} s_{k-1} - y_{k-1}\|_2^2$, we obtain the so-called spectral step length:

$$\alpha_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}.$$

By the Mean-Value Theorem of integral calculus, it follows that:

$$y_{k-1} = \left(\int_0^1 \nabla^2 f(x_{k-1} + t s_{k-1}) dt \right) s_{k-1}.$$

Hence, α_k is the inverse of a Rayleigh quotient relative to the average Hessian matrix

$$\int_0^1 \nabla^2 f(x_{k-1} + t s_{k-1}) dt,$$

and so it is between the minimum and the maximum eigenvalue of the average Hessian, which has been the main motivation for using the word *spectral* in its name. Additional properties and variations of the spectral step length can be found in [2, 3].

Writing the secant equation as $(1/\alpha_k)y_{k-1} = s_{k-1}$, which is also standard in the quasi-Newton tradition, we arrive at a different spectral coefficient:

$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}.$$

Now the formulas for α_k are used provided that for all iterations they are bounded away from zero on the positive side, and bounded away from infinity. In other words, we use some safeguard fixed parameters $0 < \alpha_{\min} < \alpha_{\max} < \infty$ and define at each iteration:

$$\alpha_k = \min\{\alpha_{\max}, \max\{\alpha_{\min}, \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}\}\}, \quad (2)$$

which will be throughout the rest of this work our spectral choice of step length.

3 SpecSampling NSG

The SpecSampling method is one of the proposed NSG methods, which shows the following general characteristics:

- Uses the user-supplied gradient function if it passes a differentiability test. Otherwise, it builds the subdifferential direction using the gradient sampling method; see [5, 6].
- Adjusts the sampling radius used by the gradient sampling method dynamically during the convergence process.
- Stops the process when the search direction is small enough, and the sampling radius cannot be reduced anymore.
- Uses a standard monotone Armijo line search as a globalization strategy.

3.1 Differentiability test

The differentiability test is based on a finite difference gradient estimation being close enough to the user-supplied gradient $\nabla f(x)$. We approximate each partial derivative $\partial f/\partial x_i$ at the current point x_k , by the commonly used formula of finite difference and verify that it is close enough to $(\nabla f(x_k))_i$, as follows:

$$\text{if } |\frac{\widehat{\partial f}}{\partial x_i} - (\nabla f(x_k))_i| < \delta, \text{ for all } 1 \leq i \leq n,$$

and for a very small $\delta > 0$, then g_k satisfies the differentiability test, where:

$$\frac{\widehat{\partial f}}{\partial x_i} = \frac{f(x + he_i) - f(x)}{h}, \quad (3)$$

and e_i is the i -th canonical vector. This test requires evaluating f at the point x_k as well as the n perturbed points $x + he_i$. Therefore, f will be evaluated a total of $n + 1$ times in the worst possible case.

Notice that the differentiability test can be verified as soon as each $(\nabla f(x_k))_i$ component is calculated, in order to determine non-differentiability as soon as possible, avoiding unnecessary additional function evaluations. Notice also that if the test fails, the gradient sampling scheme described below is used to build the subdifferential g_k . If the test succeeds the sampling gradient is not computed, and the user-supplied gradient will be used, i.e., we set $g_k = \nabla f(x_k)$ in the NSG method. In that case, only $n + 1$ evaluations of the function are required, avoiding the $2n$ evaluations required by the gradient sampling method. In addition, this simple procedure provides information about the points where the function is differentiable.

3.2 Calculating the search direction d_k

The central idea behind the gradient sampling scheme is to approximate the subdifferential g_k through a random sampling of user-supplied gradients near the current iteration point. The distance between the points used to calculate g_k and the current iteration point x_k is determined by the sampling radius $\epsilon > 0$, which is typically adjusted as the algorithm progresses. If the differentiability test fails, the vector g_k is obtained as follows: for a given ϵ , consider the map $G_\epsilon : \mathbb{R}^n \rightarrow \mathbb{R}^n$ given by

$$G_\epsilon = clconv \nabla f((x + \epsilon B) \cap W),$$

where $B = \{x : \|x\| \leq 1\}$ is the closed unit ball, and $\|\cdot\|$ is the 2-norm. Here $clconv$ denotes the closure of the convex hull set. The sets G_ϵ can be used to give the following representation of the Clarke-subdifferential of f at a point x (see [7]):

$$\tilde{\partial}f(x) = \bigcap_{\epsilon > 0} G_\epsilon(x).$$

Using now the ϵ -subdifferential introduced by Goldstein [15], for each $\epsilon > 0$, the Clarke-subdifferential is defined as:

$$\tilde{\partial}_\epsilon f(x) = clconv \tilde{\partial}f(x + \epsilon B),$$

where $G_\epsilon \subset \tilde{\partial}_\epsilon f(x)$. A point x is Clark- ϵ -stationary point for f if $0 \in \tilde{\partial}_\epsilon f(x)$. This notion of ϵ stationary is fundamental to the gradient sampling scheme, since it is designed to locate Clarke-stationary points. Therefore, a measure of proximity to Clarke ϵ -stationarity is given by

$$\rho_\epsilon = dist(0|G_\epsilon). \quad (4)$$

The vector g_k will be the element in G_ϵ for which the distance in (4) is attained. We now present the SpecSampling NSG procedure to obtain g_k and the search direction d_k .

Procedure (3.1):

1. Compute the user-supplied gradient $\nabla f(x_k)$
2. Check the differentiability test using (3), if the test is passed then $g_k = \nabla f(x_k)$, otherwise compute g_k as follows:
 - Let $u_{k1}, \dots, u_{km}$ be sampled independently and uniformly in B , where $m = 2n$.
 - Calculate the sample set by:

Setting $x_{k0} = x_k$ and

$x_{kj} = x_k + \epsilon u_{kj}$, $j = 1, \dots, m$ where $\epsilon > 0$ is the sampling radius.

- Calculate $G_k = \text{conv}\{\nabla f(x_{k0}), \nabla f(x_{k1}), \dots, \nabla f(x_{km})\}$
- Solve the quadratic program

$$\min_{g \in G_k} \|g\|^2,$$

which means, find $g_k \in G_k$ such that $\|g_k\| = \text{dist}(0|G_k)$, which is equivalent to (4).

3. Set $d_k = -\alpha_k g_k$

Typically this procedure is executed initially with a large radius ϵ , and the radius is reduced dynamically under some conditions to be discussed below.

3.3 Line search globalization strategy

The globalization strategy used at every k , in the SpecSampling NSG method, is the following standard monotone Armijo line search: find the maximum $\tau_k \in \{1, \beta, \beta^2, \dots\}$ such that:

$$f(x_k + \tau_k d_k) \leq f(x_k) - \gamma \alpha_k \tau_k g_k^T g_k, \quad (5)$$

where $0 < \beta < 1$ and $0 < \gamma < 1$ are fixed for all iterations. In practice the typical values are $\beta = 0.5$, and $\gamma = 10^{-4}$. To obtain τ_k , a standard quadratic interpolation backtracking scheme can also be used; see, e.g., [3].

3.4 Adjusting the sampling radius

As the algorithm progresses, in the same fashion as presented by [6], the sampling radius ϵ is reduced by a factor of 10^{-1} when one of the following conditions holds:

- (i) $\|d_k\|$ is less than a prefixed tolerance,
- (ii) a limit of 100 iterations per ϵ is reached,
- (iii) the maximum number of line-search backtrackings is reached.

A comprehensive explanation of the sampling radius role on gradient sampling approximations is presented by Burke *et al.* in [5].

3.5 SpecSampling Algorithm

Given x_0 in the open dense subset $W \subset \mathbb{R}^n$, an initial radius $\epsilon > 0$, a maximum number of iterations $maxiter$, $dnTol$ a tolerance for $\|d_k\|$, $\alpha_0 > 0$, and $m = 2n$. Set $k = 0$.

While $k < maxiter$ and the optimality condition is not satisfied

1. Obtain d_k using Procedure (3.1) with spectral step length α_k and sampling radius ϵ
2. Obtain τ_k for which the condition (5) is satisfied, and set $x_{k+1} = x_k + \tau_k d_k$
3. Compute α_{k+1} using (2), reduce ϵ if needed, and set $k = k + 1$

End While

The SpecSampling Algorithm stops when k reaches the maximum number of iterations, or when the optimality condition is satisfied. Our optimality condition is satisfied if $\|d_k\|$ is less

than a prefixed tolerance, and the sampling radius cannot be reduced anymore. In any case, when the algorithm stops it returns the best solution found so far.

We close this section commenting on the theoretical properties of the SpecSampling Algorithm. First we note that, in the gradient sampling method and its variants [6, 20], the subdifferential g_k is approximated at all k using a random sampling as described in Procedure (3.1), whereas the SpecSampling Algorithm uses the same Procedure (3.1) to build g_k at some iterations, and the user-supplied gradient to set $g_k = \nabla f(x_k)$ for all other iterations. Clearly, when the differentiability test succeeds, $\nabla f(x_k)$ has all the required properties satisfied by the approximated subdifferential built by the gradient sampling schemes at x_k . Moreover, based on (2), the spectral step length $\alpha_k > 0$ is uniformly bounded from above by $\alpha_{\max}$, and hence the search direction d_k built in Procedure (3.1) can be seen as a special case of the variant developed in [20, Section 4.1]. Finally, we remark that the monotone line search strategy imposed in (5) coincides with the one used for gradient sampling schemes [6, 20]. Therefore, if in addition of being locally Lipschitz continuous, we assume that f is bounded below, then from the results presented in [20, Section 4.1] it follows that with probability 1 every accumulation point of the sequence $\{x_k\}$ generated by the SpecSampling Algorithm is a Clarke-stationary point.

4 SpecSimplex NSG

The SpecSimplex method is the second NSG method proposed in this work, which shows the following general characteristics:

- Does not need gradient evaluations to compute the search direction d_k .
- Computes the subdifferential vector g_k using the SID-PSM method discussed in Custodio *et al.* [11, 12].
- Adjusts the sampling radius dynamically during the convergence process.
- Stops the process when the search direction is small enough, and the sampling radius cannot be reduced anymore.
- Uses a nonmonotone line search globalization strategy.

4.1 Calculating the search direction d_k

Based on the SID-PSM method, a finite set of directions can be generated which must positively span $\mathbb{R}^n$. Let us define D as a positive spanning set (see [12]), then the sample set Y_k is defined at the point x_k as:

$$y_k^0 = x_k$$

$$Y_k = \{y_k^i = x_k + \epsilon D(:, i) \cup \{y_k^0\}\},$$

where $D(:, i)$ is the i -th column of D and ϵ is the simplex radius.

In the determined case, the simplex gradient is computed by first sampling the objective function at $n + 1$ points (vertices). The convex hull of a set of $n + 1$ points $Y_k = \{y_k^0, y_k^1, \dots, y_k^n\}$ is called a simplex. The simplex is said to be non-singular if the matrix $S = [y_k^1 - y_k^0 \dots y_k^n - y_k^0]$ is not singular.

Given a nonsingular simplex of vertices Y_k , the simplex gradient at y_k^0 is defined as:

$$\nabla_S f(y_k^0) = S^{-T} \delta(f; S),$$

where $\delta(f; S) = [f(y_k^1) - f(y_k^0) \dots f(y_k^n) - f(y_k^0)]^T$. When the number of $q + 1$ points is not necessarily equal to $n + 1$, the simplex gradient can be also represented as a solution of the linear system:

$$S^T \nabla_S f(y_k^0) = \delta(f; S),$$

where $S = [y_k^1 - y_k^0 \dots y_k^q - y_k^0]$ and $\delta = [f(y_k^1) - f(y_k^0) \dots f(y_k^n) - f(y_k^0)]^T$. Roughly speaking, $\nabla_S f(y_k^0)$ is an approximation to the Clarke-subdifferential vector. In our setting, $g_k = \nabla_S f(y_k^0)$.

We now present the SpecSimplex NSG Procedure to generate d_k :

Procedure (4.1)

1. Generate D choosing one positive spanning set among:
 - The minimal positive basis $D = [-e \ I]$, where e is the vector of ones and I is the identity matrix
 - The maximal positive basis $D = [I \ -I]$
 - The positive spanning set $D = [e \ -e \ I \ -I]$
2. Compute a simplex gradient:
 - Generate the sampling set Y_k :
$$y_k^0 = x_k$$

$$Y_k = \{y_k^i = x_k + \epsilon D(:, i) \cup \{y_k^0\}\}, \text{ where } \epsilon \text{ is the simplex radius and } i \text{ depends on the size of } D.$$
 - Solve $S^T \nabla_S f(y_k^0) = \delta(f; S)$ or calculate $\nabla_S f(y_k^0) = S^{-T} \delta(f; S)$, where:
$$\delta(f; S) = [f(y_k^1) - f(y_k^0) \dots f(y_k^n) - f(y_k^0)]^T$$

$$S = [y_k^1 - y_k^0 \dots y_k^q - y_k^0]$$
3. Set $g_k = \nabla_S f(y_k^0)$ and $d_k = -\alpha_k g_k$

4.2 Line search globalization strategy

The globalization strategy used at every k , in the SpecSimplex method, combines and extends the Grippo *et al.* [16] line search scheme with the globalization strategy used by Diniz-Ehrhardt *et al.* [13] for a derivative-free unconstrained optimization scheme, originally introduced by La Cruz *et al.* [21] for the globalization of nonlinear systems of equations. Our acceptance condition for the next iterate is:

$$f(x_k + \tau_k d_k) \leq \max_{0 \leq j \leq \min\{k, M\}} f(x_{k-j}) - \gamma \alpha_k \tau_k^2 g_k^T g_k + \eta_k, \quad (6)$$

where $M \geq 0$ is a fixed integer, $0 < \gamma < 1$ is a small number, α_k is the spectral step length, $\tau_k > 0$ is obtained after a backtracking process that starts at 1, and $\eta_k > 0$ is chosen such that:

$$0 < \sum_k \eta_k = \eta < \infty. \quad (7)$$

The first and third terms on the right hand side of (6) are responsible for the sufficiently non-monotone behavior of the sequence $\{f(x_k)\}$. In practice, the parameters associated with the line search strategy are chosen to reduce the number of backtracking as much as possible, while keeping the convergence properties of the method. For example the parameter η_0 is chosen as a large number, e.g., $\eta_0 = \max(f(x_0), \|g(x_0)\|)$, and then it is reduced as slow as possible making sure that (7) holds.

4.3 Adjusting the simplex radius

As the algorithm progresses, the simplex radius is reduced by a factor of 0.5 when one of the following conditions holds:

- (i) $\|d_k\|$ is less than a prefixed tolerance,
- (ii) the maximum number of line-search backtrackings is reached.

4.4 SpecSimplex Algorithm

Given x_0 in the open dense subset $W \subset \mathbb{R}^n$, an initial radius $\epsilon > 0$, a maximum number of iterations *maxiter*, *dnTol* a tolerance for $\|d_k\|$, $\alpha_0 > 0$, and $M \geq 0$. Set $k = 0$.

While $k < \text{maxiter}$ and the optimality condition is not satisfied

1. Obtain d_k using Procedure (4.1) with spectral step length α_k and simplex radius ϵ
2. Obtain τ_k for which the condition (6) is satisfied, and set $x_{k+1} = x_k + \tau_k d_k$
3. Compute α_{k+1} using (2), reduce ϵ if needed, and set $k = k + 1$

End While

The SpecSimplex Algorithm stops when k reaches the maximum number of iterations, or when the optimality condition is satisfied. Our optimality condition is satisfied if $\|d_k\|$ is less than a prefixed tolerance, and the simplex radius cannot be reduced anymore. In any case, when the algorithm stops it returns the best solution found so far.

We close this section commenting on the theoretical properties of the simplex gradient vectors g_k and the SpecSimplex Algorithm. In order to apply the theoretical results established by Diniz-Ehrhardt *et al.* [13], we need to assume that there exists a subsequence for which d_k , obtained using Procedure (4.1), is a descent direction for f . In general the negative of the simplex gradient, $-g_k$, might not be a descent direction for f at every k , but as established by Regis [25] it is a descent direction at every k for the linear interpolating map that approximates f . Moreover, if f is Lipschitz continuous near a limit point (say x^*), and since in the SpecSimplex Algorithm the simplex radius ϵ is dynamically reduced, then as established by Custodio *et al.* [11] the negative simplex gradient is an ϵ -approximation to the exact negative gradient; and in particular if f is differentiable at x^* then a subsequence of simplex gradients converges to the exact gradient of f at x^* . Clearly, in the differentiable case, the exact negative gradient is a descent direction for f at every k . Therefore, under the mild and reasonable assumption that there exists a subsequence for which d_k is a descent direction for f , and under the additional assumption that f is continuously differentiable and bounded below, and that the closed set $\{x : f(x) \leq f(x_0) + \eta\}$ is bounded, it follows from the results presented in [13] that every accumulation point of the sequence $\{x_k\}$ generated by the SpecSimplex Algorithm is a stationary point.

5 Numerical Results

To give further insight into the nonsmooth spectral approach we present the results of some numerical experiments. For that we use 10 nonsmooth unconstrained minimization problems taken from [19]. Table 1 and 2 describe these problems. Table 2 shows the optimal value $f(x^*)$, whether the function is convex or not, the dimension n , and the initial point x_0 .

SpecSampling and SpecSimplex were implemented using Matlab 2013. The experiments were executed in a laptop with CPU Intel Core i5 2.60 GHz, RAM memory of 16 GB, running Windows 7 64-bits.

Table 1: Problems

Prob.	$f(x)$
P1:Generalization of MAXQ	$\max_{1 \leq i \leq n} x_i^2$
P2:Generalization of MXHILB	$\max_{1 \leq i \leq n} \left \sum_{j=1}^n \frac{x_j}{i+j-1} \right $
P3:Chained LQ	$\sum_{i=1}^{n-1} \max\{-x_i - x_{i+1}, -x_i - x_{i+1} + (x_i^2 + x_{i+1}^2 - 1)\}$
P4:Chained CB3 I	$\sum_{i=1}^{n-1} \max\{x_i^4 + x_{i+1}^2, (2 - x_i)^2 + (2 - x_{i+1})^2, 2e^{-x_i + x_{i+1}}\}$
P5:Chained CB3 II	$\max\{\sum_{i=1}^{n-1} (x_i^4 + x_{i+1}^2), \sum_{i=1}^n ((2 - x_i)^2 + (2 - x_{i+1})^2), \sum_{i=1}^n (2e^{-x_i + x_{i+1}})\}$
P6:Number of active faces (Nactive)	$\max_{1 \leq i \leq n} \{g(-\sum_{i=1}^n (x_i)), g(x_i)\}$ where $g(y) = \ln(y + 1)$
P7:Nonsmooth generalization of Brown 2	$\sum_{i=1}^{n-1} (x_i ^{x_{i+1}^2 + 1 + x_{i+1} } x_{i+1} ^{x_i^2 + 1})$
P8:Chained Mifflin 2	$\sum_{i=1}^{n-1} (-x_i + 2(x_i^2 + x_{i+1}^2 - 1) + 1.75 x_i^2 + x_{i+1}^2 - 1)$
P9:Chained crescent I	$\max\{\sum_{i=1}^{n-1} (x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1), \sum_{i=1}^{n-1} (-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1)\}$
P10:Chained crescent II	$\{\sum_{i=1}^{n-1} \max\{(x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1), (-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1)\}$

Table 2: Problems

Prob	f^*	Convex	n	Initial Points
P1	0.0	yes	20	$x_i^0 = i, i = 1, \dots, n/2$ $x_i^0 = -i, i = n/2 + 1, \dots, n$
P2	0.0	yes	50	$x_i^0 = 1.0, i = 1, \dots, n$
P3	$-(n-1)2^{1/2}$	yes	$n = 2$	$x_i^0 = -0.5, i = 1, \dots, n$
P4	$2(n-1)$	yes	$n = 2$	$x_i^0 = 2.0, i = 1, \dots, n$
P5	$2(n-1)$	yes	$n = 2$	$x_i^0 = 2.0, i = 1, \dots, n$
P6	0.0	no	$n = 2$	$x_i^0 = 1.0, i = 1, \dots, n$
P7	0.0	no	$n = 2$	$x_i^0 = 1$ when $\text{mod}(i, 2) = 0$ $x_i^0 = -1$ when $\text{mod}(i, 2) = 1$
P8	-34.795	no	$n = 50$	$x_i^0 = -1.0, i = 1, \dots, n$
P9	0.0	no	$n = 2$	$x_i^0 = 2$ when $\text{mod}(i, 2) = 0$ $x_i^0 = -1.5$ when $\text{mod}(i, 2) = 1$
P10	0.0	no	$n = 2$	$x_i^0 = 2$ when $\text{mod}(i, 2) = 0$ $x_i^0 = -1.5$ when $\text{mod}(i, 2) = 1$

5.1 Numerical results for the SpecSampling method

We solved the problems with the original gradient sampling proposed in [6] and with the SpecSampling algorithm in order to compare their results.

For the gradient sampling method, the parameters were set in the following way: the sampling radius is initialized at $\epsilon = 0.1$, and the maximum number of iterations for each sampling radius is 100. If the maximum number of iterations is reached or $\|d_k\|$ is less than a tolerance, then the sampling radius is reduced by 0.1. The smallest sampling radius allowed is 10^{-6} . Thus, the total number of iterations is at most 600. Given the stochastic nature of the gradient sampling, we ran it 20 times; The results shown are for the run achieving the lowest value of f . It is important to say that for the rest of the parameters, we used the same values used in [6].

We report in Table 3: f_{SG} as the optimal obtained value, IT as the number of iterations, $\|d_{min}\|$ as the norm of the direction at the obtained optimal value, $ExeT$ as the execution time in seconds, ϵ as the sampling radius value when the optimal was reached, $feval$ as the number of evaluations of the function, and $\%suc$ as the percentage of success finding an acceptable solution out of the 20 executions. A solution is considered acceptable if the relative error is less than 10^{-4} .

Table 3: Gradient sampling results

Prob.	f_{SG}	IT	$\ d_{min}\ $	$ExeT$	ϵ	$feval$	$\%suc$
P1	4.0311e-13	169	5.16e-07	4.43	1.00e-06	7177	1.00
P2	2.9566e-09	40	5.43e-17	4.99	1.00e-06	4286	1.00
P3	-1.4142e+00	30	7.91e-12	0.16	1.00e-06	404	1.00
P4	2.0000e+00	43	4.83e-16	0.12	1.00e-06	719	1.00
P5	2.0000e+00	33	3.51e-16	0.11	1.00e-06	538	1.00
P6	2.7000e-07	18	3.93e-17	0.06	1.00e-06	191	0.35
P7	7.3107e-07	93	1.53e-15	3.11	1.00e-06	4983	1.00
P8	-3.4795e+01	543	3.98e+00	130.57	1.00e-06	64610	1.00
P9	3.3636e-11	43	9.31e-08	0.13	1.00e-06	722	1.00
P10	1.1279e-09	44	6.43e-07	0.13	1.00e-06	577	1.00

For the proposed SpecSampling method, the parameters were set following the algorithmic description in Section 3, as follows: for the sampling set we use $m = 2n$. The sampling radius ϵ starts as 10^{-1} and it is reduced up to 10^{-6} . The maximum number of iterations for each sampling radius is 100. If the maximum number of iterations is reached or $\|d_k\|$ is less than the tolerance ($dnTol = 10^{-6}$) then the sampling radius is reduced. For the line search we use $\gamma = 0$ and the maximum number of backtrackings per iteration is 20. The spectral step length is kept between $\alpha_{min} = 10^{-3}$ and $\alpha_{max} = 10^3$. The maximum number of allowed iterations is $maxiter = 600$.

As for the experiments with gradient sampling, we ran the SpecSampling algorithm 20 times. The results shown are for the run achieving the lowest value of f . In table 4 the results for the SpecSampling method are reported as follows: $SpecSam$ is the optimal estimate obtained by the algorithm, pos is the iteration in which the best estimated was found, $ExeT$ is the execution time in seconds, $diftest$ is the number of times that the differentiability test was satisfied, which also indicates the number of times the user-supplied gradient function was used, $\|d_{min}\|$ is the norm of the direction at the best obtained value, $TIter$ is the total number of iterations executed by the algorithms, $feval$ as the number of function evaluations, and $\%suc$ as the percentage of executions when an acceptable solution was found.

Table 4: SpecSampling results

Prob	<i>SpecSam</i>	<i>pos</i>	<i>diftest</i>	$\ d_{min}\ $	<i>TIter</i>	<i>ExeT</i>	<i>feval</i>	<i>%suc</i>
P1	2.2373e-14	101	94	1.45e-07	102	0.25	2287	1.00
P2	7.4167e-07	49	1	5.31e-20	50	3.46	4572	0.05
P3	-1.4142e+00	20	21	2.93e-01	28	0.06	355	1.00
P4	2.0000e+00	39	7	4.47e-03	45	0.23	709	1.00
P5	2.0000e+00	57	8	2.83e-03	58	0.28	1058	1.00
P6	4.0172e-10	18	23	1.41e-03	24	0.03	347	1.00
P7	1.3400e-08	51	42	7.94e-04	55	0.33	2563	0.80
P8	-3.4793e+01	227	0	1.65e-01	228	16.75	24122	1.00
P9	2.8559e-09	17	22	5.00e-01	29	0.06	420	1.00
P10	7.3144e-12	27	12	8.88e-09	28	0.11	330	0.95

Concerning quality of the solutions, the results reported in tables 3 and 4 are similar. Nevertheless, in general, the SpecSampling requires less iterations, shows less time of execution, and less evaluations of the function than the gradient sampling method. We believe that this attractive behavior of the SpecSampling comes from the incorporation of the spectral step length, and also from the use of the differentiability test. For example, for Problems P1 and P7, where the number of times that the differentiability test was satisfied turned out to be high, the SpecSampling method showed significantly better results.

On the other hand, the *%Suc* of gradient sampling is 100% for 9 out of 10 problems while for the SpecSampling 100% is obtained at 7 out of 10 problems, although two of the other problems show 80% and 95%. In our opinion, this is happening because of the wide free-movement of the spectral step length. In fact, in order to achieve convergence in some problems, it was important to maintain the spectral step length between the bounds $\alpha_{min} = 10^{-3}$ and $\alpha_{max} = 10^3$. This is the first time the spectral approach is combined with a stochastic method such as the gradient sampling for general functions, so more numerical experimentation should be performed in order to obtain more insight about the spectral step behavior for nonsmooth problems.

5.2 Numerical results for the SpecSimplex method

In this section, we present numerical results to compare the behavior of SpecSimplex method with the SID-PSM method. For the SID-PSM method, the parameters were set to their default values; see [11, 12]. The most relevant parameters for the numerical experiments are: the sampling set D which is generated using the maximal positive basis as the set $[I - I]$, which means $m = 2n$, and the first mesh size ϵ_0 was set to 1.2. The SID-PSM algorithm stops when the norm of the scaled simplex gradient is less than 10^{-6} .

In table 5, we report the results for the SID-PSM method. Table 5 shows: $f_{SID-PSM}$ as the optimal value obtained by the algorithm, IT as the number of iterations, $ExeT$ as the execution time in seconds, ϵ_{mesh} as the final mesh size, $SSGrad$ as the norm of the scaled simplex gradient, and $feval$ as the number of evaluations of the function.

For the proposed SpecSimplex method, the parameters were set following the algorithmic description in Section 4, as follows: the sampling set D is generated using the maximal positive basis $[I - I]$, which means $m = 2n$. The initial simplex radius ϵ is set to 1.2 and it is reduced by half if the norm of the gradient reaches a tolerance of 10^{-2} . For the nonmonotone line search strategy, we set $\gamma = 0$, $M = 1$, and the maximum number of backtracking per iteration is 20. The spectral step length is kept between $\alpha_{min} = 10^{-10}$ and $\alpha_{max} = 10^{10}$. The maximum number

Table 5: SID-PSM Results

Prob	$f_{SID-PSM}$	IT	$ExeT$	ϵ_{mesh}	$SSGrad$	$feval$
P1	2.6524e-07	297	23.89	5.8594e-04	6.9674e-07	3113
P2	7.7363e-02	111	984	1.4305e-07	6.3175e-08	3450
P3	-1.4142e+00	42	0.062	2.8610e-07	5.7220e-07	135
P4	2.0000e+00	65	0.094	2.8610e-07	8.7066e-07	177
P4	2.0000e+00	65	0.093	2.8610e-07	8.7066e-07	177
P6	7.4071e-08	40	0.062	2.8610e-07	5.2142e-07	136
P7	5.9434e-01	175	11.97	7.1526e-08	4.6099e-07	2218
P8	3.4424e+01	416	4286	1.4305e-07	9.6241e-07	5851
P9	5.5631e-05	43	0.078	2.8610e-07	5.7041e-07	171
P10	2.7529e-07	79	0.125	2.8610e-07	6.1544e-07	258

of allowed iterations is $maxiter = 600$, and $dnTol = 10^{-6}$.

In table 6 we report: $f_{SpecSim}$ as the optimal value obtained by the algorithm, pos as the iteration in which the best estimated was found, $ExeT$ as the execution time in seconds, $TIter$ as the total number of iterations executed by the method, ϵ as final simplex radius, $\|d_{min}\|$ as the direction norm at the best estimated, and $feval$ as the number of function evaluations.

Table 6: SpecSimplex results

Prob	$f_{SpecSimp}$	pos	$ExeT$	ϵ	$TIter$	$\ d_{min}\ $	$feval$
P1	3.7547e-44	36	0.1090	5.7e-07	36	3.92e-22	1517
P2	1.4745e-17	126	0.9850	5.7e-07	140	3.50e-16	14276
P3	-1.4142e+00	66	0.0160	5.7e-07	66	4.94e-14	335
P4	2.0000e+00	85	0.0310	5.7e-07	85	4.49e-13	439
P5	2.0000e+00	85	0.0470	5.7e-07	85	4.49e-13	439
P6	3.7210e-12	26	0.0150	5.7e-07	26	3.95e-12	138
P7	9.7830e-13	25	0.0470	5.7e-07	25	2.58e-13	1067
P8	-3.4779e+01	548	1.1250	2.3e-03	600	3.24e-01	60778
P9	5.7220e-07	68	0.0160	5.7e-07	68	8.26e-14	345
P10	5.0312e-07	352	0.1560	1.1e-06	600	2.65e-05	3893

Concerning quality of the solutions, tables 5 and 6 show that results obtained by SpecSimplex are better, in particular for problems P2, P7 and P8. In some cases the SpecSimplex requires more iterations than the SID-PSM, but an iteration of SpecSimplex is less expensive than an iteration of SID-PSM. This can be observed in the execution time reported for both methods. Indeed, an iteration of the SID-PSM is more expensive because of the diagonal approximation of the Hessian matrix that it performs. The number of evaluations of the function is higher for the SpecSimplex because of the nonmonotone line search, while SID-PSM does not have a globalization strategy. For Problem P8, SID-PSM does not converge and SpecSimplex stops after reaching the maximum number of iterations. However, SpecSimplex produces a better estimate of the solution than SID-PSM.

In our numerical experimentation, we observed that the spectral step works very well in combination with simplex gradients. Moreover, it was not necessary to narrow its bounds α_{max} or α_{min} to improve convergence. For the line search condition, using $\gamma = 0$ or $\gamma = 10^{-4}$ did not make any difference. The parameter η was needed in order to observe a faster convergence.

Acknowledgments. The authors thank Professor Ana Luisa Custodio who kindly provided us with the SID-PSM Matlab implementation, and Professors J.V. Burke, A.S. Lewis and M.L.

Overton for making the Matlab implementation of the gradient sampling algorithm publicly available.

References

- [1] J. Barzilai and J. M. Borwein, Two point step size gradient methods, *IMA Journal of Numerical Analysis* 8, pp. 141–148, 1988.
- [2] E. G. Birgin, J. M. Martínez, and M. Raydan, *Spectral Projected Gradient Methods*, in Encyclopedia of Optimization, Second Ed., Editors: C. A. Floudas and P. M. Pardalos, Part 19, pp. 3652–3659, 2009.
- [3] E. G. Birgin, J. M. Martínez, and M. Raydan, Spectral Projected Gradient methods: Review and Perspectives, *Journal of Statistical Software*, 60(3), pp. 1–21, 2014.
- [4] D. M. Bortz and C. T. Kelley, The simplex gradient and noisy optimization problems, *Computational Methods in Optimal Design and Control, Progress in Systems and Control Theory*, 24, pp. 77–90, 1998.
- [5] J. V. Burke, A. S. Lewis, and M. L. Overton, Approximating subdifferentials by random sampling of gradients, *Mathematics of Operation Research* 27, pp. 567–584, 2002.
- [6] J. V. Burke, A. S. Lewis, and M. L. Overton, A robust gradient sampling algorithm for nonsmooth, nonconvex optimization, *SIAM J. Optim.* 15, pp. 751–779, 2005.
- [7] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, 1990.
- [8] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, Philadelphia, 2009.
- [9] A. Crema, M. Loreto, and M. Raydan, Spectral projected subgradient with a momentum term for the Lagrangean dual approach, *Computers and Operations Research* 34, pp. 3174–3186, 2007.
- [10] F. E. Curtis and X. Que, An adaptive gradient sampling algorithm for non-smooth optimization, *Optimization Methods and Software* 28, pp. 1302–1324, 2013.
- [11] A. L. Custodio, J. E. Dennis Jr., and L. N. Vicente, Using simplex gradients of nonsmooth functions in direct search methods, *IMA Journal of Numerical Analysis* 28, pp. 770–784, 2008.
- [12] A. L. Custodio and L. N. Vicente, Using sampling and simplex derivative in pattern search methods, *SIAM J. Optim.* 18, pp. 537–555, 2007.
- [13] M. A. Diniz-Ehrhardt, J. M. Martínez, and M. Raydan, A derivative-free nonmonotone line search technique for unconstrained optimization, *Journal of Computational and Applied Mathematics*, 219, pp. 383–397, 2008.
- [14] R. Fletcher, On the Barzilai-Borwein method, in *Optimization and Control with Applications* (L. Qi, K. teo, and X. Yang eds.), Series in Applied Optimization Volume 96, Kluwer, pp. 235–256, 2005.

- [15] A. A. Goldstein, Optimization of Lipschitz continuous functions, *Math. Program.* 3, pp. 14–22, 1977.
- [16] L. Grippo, F. Lampariello, and S. Lucidi, A nonmonotone line search technique for Newton’s method, *SIAM Journal on Numerical Analysis* 23, pp. 707–716, 1986.
- [17] L. Grippo and F. Rinaldi, A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations, *Computational Optimization and Applications* 60, pp. 1–33, 2015.
- [18] E. S. Helou, S. A. Santos, and L. E. A. Simões, A nonmonotone approach without differentiability test for gradient sampling methods, *Optimization Online Digest*, March, 2015.
- [19] N. Karmitsa, *Test Problems for Large-Scale Nonsmooth Minimization*, Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing, No. B. 4/2007, University of Jyväskylä, 2007.
- [20] K. C. Kiwiel, Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization, *SIAM Journal on Optimization* 18, pp. 379–388, 2007.
- [21] W. La Cruz, J. M. Martínez, and M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Mathematics of Computation* 75, pp. 1429–1448, 2006.
- [22] M. Loreto and A. Crema, Convergence analysis for the modified spectral projected sub-gradient method, *Optimization Letters* 9, pp. 915–929, 2015.
- [23] M. Raydan, On the Barzilai and Borwein choice of steplength for the gradient method, *IMA Journal of Numerical Analysis* 13, pp. 321–326, 1993.
- [24] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal on Optimization* 7, pp. 26–33, 1997.
- [25] R. G. Regis, The calculus of simplex gradients, *Optimization Letters* 9, pp. 845–865, 2015.