A Population-Based Stochastic Coordinate Descent Method

Ana Maria A.C. Rocha[1,2], M. Fernanda P. Costa[3,4], and Edite M.G.P. Fernandes[5]

[1] ALGORITMI Center,
University of Minho, 4710-057 Braga, Portugal,
[2] Department of Production and Systems,
University of Minho
arocha@dps.uminho.pt
[3] Centre of Mathematics,
University of Minho, 4710-057 Braga, Portugal,
[4] Department of Mathematics,
4800-058 Guimarães, Portugal
mfc@math.uminho.pt
[5] ALGORITMI Center,
University of Minho, 4710-057 Braga, Portugal
emgpf@dps.uminho.pt

Abstract. This paper addresses the problem of solving a bound constrained global optimization problem by a population-based stochastic coordinate descent method. To improve efficiency, a small subpopulation of points is randomly selected from the original population, at each iteration. The coordinate descent directions are based on the gradient computed at a special point of the subpopulation. This point could be the best point, the center point or the point with highest score. Preliminary numerical experiments are carried out to compare the performance of the tested variants. Based on the results obtained with the selected problems, we may conclude that the variants based on the point with highest score are more robust and the variants based on the best point less robust, although they win on efficiency but only for the simpler and easy to solve problems.

Keywords: Global Optimization; Stochastic Coordinate Descent

1 Introduction

The optimization methods for solving problems that have big size of data, like large-scale machine learning, can make use of classical gradient-based methods, namely the full gradient, accelerated gradient and the conjugate gradient, classified as batch approaches [1]. Using intuitive schemes to reduce the information data, the stochastic gradient approaches have shown to be more efficient than the batch methods. An appropriate approach to solve this type of problems is through coordinate descent methods. Despite the fact that they were the first

optimization methods to appear in the literature, they have received much attention recently. Although the global optimization (GO) problem addressed in this paper has not a big size of data, the herein proposed solution method is iterative, stochastic and relies on a population of candidate solutions at each iteration. Thus, a large amount of calculations may be required at each iteration. To improve efficiency, we borough some of the ideas that are present in machine learning techniques and propose a population-based stochastic coordinate descent method. This paper comes in the sequence of the work presented in [2].

We consider the problem of finding a global solution of a bound constrained nonlinear optimization problem in the following form:

$$\begin{aligned} &\min \ f(x) \\ &\text{subject to } \ x \in \Omega, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear function and $\Omega = \{x \in \mathbb{R}^n : -\infty < l_i \leq x_i \leq u_i < \infty, i = 1, \ldots, n\}$ is a bounded feasible region. We assume that the objective function f is differentiable, nonconvex and may possess many local minima in the set Ω. We assume that the optimal set X^* of the problem (1) is nonempty and bounded, x^* is a global minimizer and f^* represents the global optimal value. To solve the GO problem (1), a stochastic or a deterministic method may be selected. A stochastic method provides a solution, in general in a short CPU time, although it may not be globally optimal. On the other hand, a deterministic method is able to compute an interval that contains the global optimal solution, but requires a much larger computational effort [3]. To generate good solutions with less computational effort and time, approximate methods or heuristics may be used. Some heuristics use random procedures to generate candidate solutions and perform a series of operations on those solutions in order to find different and hopefully better solutions. They are known as stochastic heuristics. A method for GO has two main goals. One intends to explore the search domain for the region where the global optimal solution lies, the other intensifies the search in a vicinity of a promising region in order to compute a high quality approximation.

This paper aims to present a practical study involving several variants of a population-based stochastic method for solving the GO problem (1). Since our goal is to make the method robust and as efficient as possible, a strategy based on coordinate descent directions is applied. Although a population of candidate solutions/points of large size is initially generated, only a very small subset of those points is randomly selected, at each iteration – henceforward denoted as subpopulation – to provide an appropriate approximation, at the end of each iteration. Since robustness of the method is to be privileged, the point of each subpopulation that is used to define the search direction to move each point of the subpopulation is carefully chosen in order to potentiate both exploration and exploitation abilities of the method. The point with the highest score of the subpopulation is proposed. A comparison with the best and the center points is also carried out.

This paper is organized as follows. Section 2 briefly presents the coordinate descent method and Sect. 3 describes the herein proposed stochastic coordinate descent method when applied to a population of points. Finally, Sect. 4 contains the results of our preliminary numerical experiments and we conclude the paper with the Sect. 5.

2 Coordinate Descent Method

This section briefly presents the coordinate descent method (CDM) and its stochastic variant. The CDM operates by taking steps along the coordinate directions [1,4]. Hence, the search direction for minimizing f from the iterate x_k, at iteration k, is defined as

$$d_k = -\nabla_{i_k} f(x_k) e_{i_k} \tag{2}$$

where $\nabla_{i_k} f(\cdot)$ represents the component i_k of the gradient of f, e_{i_k} represents the i_kth coordinate vector for some index i_k, usually chosen by cycling through $\{1, 2, \ldots, n\}$, and x_{i_k} is the i_kth component of the vector $x \in \mathbb{R}^n$. For a positive step size, α_k, the new approximation, x_{k+1}, differs from x_k only in the i_kth component and is computed by $x_{k+1} = x_k + \alpha_k d_k$. Note that the direction shown in (2) might not be a negative directional derivative for f at x_k. When the index i_k to define the search direction and the component of x_k to be adjusted is chosen randomly by Uniform distribution ($\mathbf{U}$) on $\{1, 2, \ldots, n\}$, *with* or *without replacement*, the CDM is known as a stochastic CDM. This type of method has attracted the attention of the scientific community because of their usefulness in data analysis and machine learning. Applications are varied, in particular in support vector machine problems [5].

3 A Population-Based Stochastic Coordinate Descent Method

At each iteration of a population-based algorithm, a set of points is generated aiming to explore the feasible region for a global optimum. Let $|P|$ denote the number of points in the population, where $x_i \in \mathbb{R}^n$ represents the point with index i of the population, where $i \in P = \{1, 2, \ldots, |P|\}$. The likelihood is that the greater the $|P|$ the better is the exploration feature of the algorithm. However, to handle and evaluate the objective f for a large number of points is time consuming.

In order to improve the efficiency of the method, the number of function evaluations must be reduced. Thus, the method is based on a random selection of points from the original population, at each iteration k – herein designated by the subpopulation k. Thus, at each iteration, a subpopulation of points (of small size) is selected to be evaluated and potentially moved in direction to the global optimum. This random selection uses the $\mathbf{U}$ either *with* or *without replacement* to select the indices for the subpopulation from the set $\{1, 2, \ldots, |P|\}$. Let

$P_1, P_2, \ldots, P_k, \ldots$ be the sets of indices of the subpopulation randomly chosen from P.

At each iteration k there is a special point that is maintained for the next iteration. This point is the best point of the current subpopulation. This way, for $k > 1$, the randomly selected set P_k does not include the index of the best point from the previous subpopulation and the size of the subpopulation k is $|P_k| + 1$. We note that the size of the subpopulation at the first iteration is $|P_1|$. The subsets of indices when generating the subpopulation satisfy the following conditions: (i) $P_1 \subset P$ and $P_{k+1} \subset P \backslash \{k_b\}$ for $k \geq 1$; (ii) $|P_1| \ll |P|$; (iii) $|P_2| + 1 \leq |P_1|$ and $|P_{k+1}| \leq |P_k|$ for $k > 1$; where k_b is the index of the best point of the subpopulation k. Onwards $P_1^+ = P_1$ and $P_{k+1}^+ = P_{k+1} \cup \{k_b\}$ for $k \geq 1$ are used for simplicity [2].

We now show how each point x_{k_j} $(j = 1, \ldots, |P_k^+|)$ of the subpopulation k is moved. For each point, a search direction is generated. Thus, the point x_{k_j} may be moved along the direction, d_{k_j}, as follows:

$$x_{k_j} = x_{k_j} + \alpha_{k_j} d_{k_j} \tag{3}$$

where $0 < \alpha_{k_j} \leq 1$ is the step length computed by a backtracking strategy.

The direction d_{k_j} used to move the point x_{k_j} is defined by

$$d_{k_j} = -\nabla_i f(x_{k_H}) e_i \tag{4}$$

where e_i represents the ith coordinate vector for some index i, randomly selected from the set $\{1, 2, \ldots, n\}$. We note that the search direction is along a component of the gradient computed at a special point of the subpopulation k, x_{k_H}, further on denoted by the point with the highest score. Since d_{k_j} might not be a descent direction for f at x_{k_j}, the movement according to (3) is applied only if d_{k_j} is descent for f at x_{k_j}. Otherwise, the point x_{k_j} is not moved. Whenever the new position of the point falls outside the bounds, a projection onto Ω is carried out.

The index of the point with highest score k_H, at iteration k, satisfies

$$k_H = \arg \max_{j=1,\ldots,|P_k^+|} s(x_{k_j}) \text{ where } s(x_{k_i}) = \hat{D}(x_{k_i}) - \hat{f}(x_{k_i}) \tag{5}$$

is the score of the point x_{k_i} [6]. The normalized distance $\hat{D}(x_{k_i})$, from x_{k_i} to the center point of the k subpopulation, and the normalized objective function value $\hat{f}(x_{k_i})$ at x_{k_i} are defined by

$$\hat{D}(x_{k_i}) = \frac{D(x_{k_i}) - \min_{j=1,\ldots,|P_k^+|} D(x_{k_j})}{\max_{j=1,\ldots,|P_k^+|} D(x_{k_j}) - \min_{j=1,\ldots,|P_k^+|} D(x_{k_j})} \tag{6}$$

and

$$\hat{f}(x_{k_i}) = \frac{f(x_{k_i}) - \min_{j=1,\ldots,|P_k^+|} f(x_{k_j})}{\max_{j=1,\ldots,|P_k^+|} f(x_{k_j}) - \min_{j=1,\ldots,|P_k^+|} f(x_{k_j})} \tag{7}$$

respectively. The distance function $D(x_{k_i})$ (to the center point $\bar{x}_k$) is measured by $\|x_{k_i} - \bar{x}_k\|_2$ and the center point is evaluated as follows:

$$\bar{x}_k = \frac{1}{|P_k^+|} \sum_{j=1}^{|P_k^+|} x_{k_j}. \tag{8}$$

We note here that the point with the highest score in each subpopulation is the point that lies far away from the center of the region defined by its points (translated by $\bar{x}$) that has the lowest function value. This way, looking for the largest distance to $\bar{x}$, the algorithm potentiates its exploration ability, and choosing the one with lowest f value, the algorithm reenforces its local exploitation capability. For each point with index $k_j, j = 1, \ldots, |P_k^+|$, the gradient coordinate index i may be randomly selected by **U** on the set $\{1, 2, \ldots, n\}$ one at a time for each k_j *with replacement*. However, the random choice may also be done using **U** on $\{1, 2, \ldots, n\}$ but *without replacement*. In this later case, when all indices have been chosen, the set $\{1, 2, \ldots, n\}$ is shuffled [5].

The stopping condition of our population-based stochastic coordinate descent algorithm aims to guarantee a solution in the vicinity of f^*. Thus, if

$$|f(x_{k_b}) - f^*| \leq \epsilon |f^*| + \epsilon^2, \tag{9}$$

where x_{k_b} is the best point of the subpopulation k and f^* is the known global optimum, is satisfied for a given tolerance $\epsilon > 0$, the algorithm stops. Otherwise, the algorithm runs until a specified number of function evaluations, $nf_{\max}$, is reached. The main steps of the algorithm are shown in Algorithm 1.

```
Randomly generate the population in Ω
repeat
    Randomly select a subpopulation for iteration k and select x_{k_H}
    for each point x_{k_j} in the subpopulation do
        Randomly select i ∈ {1, ..., n} to choose the component of ∇f at x_{k_H}
        Compute the search direction d_{k_j} according to (4)
        if d_{k_j} is descent for f at x_{k_j} then
            Move x_{k_j} according to (3)
    Select the best point x_{k_b} of the subpopulation
until (9) is satisfied or the number of function evaluations exceeds nf_max
```

Algorithm 1: Population-based stochastic coordinate descent algorithm

4 Numerical Experiments

During the preliminary numerical experiments, well-known benchmark problems are used: BO (Booth, $n = 2$), BP (Branin, $n = 2$), CB6 (Camel6, $n = 2$), DA (Dekkers & Aarts, $n = 2$), GP (Goldstein & Price, $n = 2$), HSK (Hosaki, $n = 2$), MT (Matyas, $n = 2$), MC (McCormick, $n = 2$), MHB (Modified Himmelblau, $n = 2$), NF2 (Neumaier2, $n = 4$), PWQ (Powell Quadratic, $n = 4$), RG-2, RG-5,

RG-10 (Rastrigin, $n = 2$, $n = 5$, $n = 10$) RB (Rosenbrock, $n = 2$), WF (Wood, $n = 4$), see the full description in [7]. The Matlab^{TM} (Matlab is a registered trademark of the MathWorks, Inc.) programming language is used to code the algorithm and the tested problems. The parameter values are set as follows: $|P| = 500$, $|P_1| = 0.01|P|$, $|P_k| = |P_1| - 1$ for all $k > 1$, $\epsilon = 1E - 04$ and $nf_{\max} = 50000$.

In our previous work [2], we have used the gradient computed at $\bar{x}$. Besides this variant, we have also tested a variant where the gradient is computed at the best point of the subpopulation. These variants are now compared with the new strategy based on the gradient computed at the point with highest score, summarized in the previous section. All the tested variants are termed as follows:

- best_w (best_wout): gradient computed at the best point and the coordinate index i (see (4)) is randomly selected by **U** *with (without) replacement*;
- center_w (center_wout): gradient computed at $\bar{x}$ and the coordinate index i is randomly selected by **U** *with (without) replacement*;
- hscore_w (hscore_wout): gradient computed at the point with highest score and the coordinate index i is randomly selected by **U** *with (without) replacement*;
- best_full_g (center_full_g / hscore_full_g): using the full gradient computed at the best point ($\bar{x}$ / the point with highest score) to define the search direction.

Each variant was run 30 times with each problem. Tables 1 and 2 show the average of the obtained f solution values over the 30 runs, f_{avg}, the minimum f solution value obtained after the 30 runs, f_{min}, the average number of function evaluations, nf_{avg}, and the percentage of successful runs, $\%s$, for the variants best_w, center_w, hscore_w and best_wout, center_wout, hscore_wout. A successful run is a run which stops with the stopping condition for the specified ϵ, see (9). The other statistics also reported in the tables are: (i) the % of problems with 100% of successful runs (% prob_100%); (ii) the average nf in problems with 100% of successful runs (nf_{avg}_100%); (iii) average nf in problems with 100% of successful runs simultaneously in the 3 tested variants (for each table) (nf_{avg}_all100%). A result printed in 'bold' refers to the best variant shown and compared in that particular table. From the results, we may conclude that using *with* or *without replacement* to choose the coordinate index i (see (4)) has no influence on the robustness and efficiency of the variant based on the gradient computed at the best point. Variants best_w and best_wout are the less robust and variants center_w, hscore_w and hscore_wout are the most robust.

When computing the average number of function evaluations for the problems that have 100% of successful runs in all the 3 tested variants, best_w wins, followed by hscore_w and then by center_w (same is true for best_wout, hscore_wout and center_wout). We remark that these average number of evaluations correspond to the simpler and easy to solve problems. For the most difficult problems and yet larger problems, the variants hscore_w (75% against 50% and 69%) and hscore_wout (69% against 50% and 63%) win as far as robustness is concerned. This justifies their larger nf_{avg}_100% values.

Table 1. Results based on the use of one coordinate of the gradient, randomly selected *with replacement*.

	best_w				center_w				hscore_w			
	f_{avg}	f_{min}	nf_{avg}	% s	f_{avg}	f_{min}	nf_{avg}	% s	f_{avg}	f_{min}	nf_{avg}	% s
BO	6.772E-09	4.354E-09	**495**	100	6.616E-09	1.496E-11	2082	100	6.543E-09	2.061E-09	1555	100
BP	3.979E-01	3.979E-01	**96**	100	3.979E-01	3.979E-01	681	100	3.979E-01	3.979E-01	239	100
CB6	-1.032E+00	-1.032E+00	935	100	-1.032E+00	-1.032E+00	**385**	100	-1.032E+00	-1.032E+00	512	100
DA	-2.478E+04	-2.478E+04	**786**	100	-2.478E+04	-2.478E+04	1251	100	-2.478E+04	-2.478E+04	1020	100
GP	3.000E+00	3.000E+00	**828**	100	3.000E+00	3.000E+00	1262	100	3.000E+00	3.000E+00	1564	100
HSK	-2.346E+00	-2.346E+00	**81**	100	-2.346E+00	-2.346E+00	305	100	-2.346E+00	-2.346E+00	110	100
MT	9.652E-09	9.006E-09	**1542**	100	8.650E-09	5.902E-09	2255	100	8.556E-09	1.157E-09	2159	100
MC	-1.913E+00	-1.913E+00	**93**	100	-1.913E+00	-1.913E+00	318	100	-1.913E+00	-1.913E+00	172	100
MHB	3.510E-01	2.662E-10	12144	77	5.525E-09	7.487E-10	1721	100	4.254E-09	5.141E-10	**1450**	100
NF2	3.728E-03	3.690E-05	50009	0	1.023E-02	6.221E-06	50020	0	4.403E-03	6.601E-05	50020	0
PWQ	6.514E-03	1.936E-07	50013	0	5.965E-03	5.386E-06	50019	0	6.655E-03	1.723E-05	50021	0
RG-2	4.643E-01	3.165E-09	18947	63	4.568E-09	1.868E-11	**1505**	100	4.160E-09	1.600E-10	2074	100
RG-5	3.283E+00	5.984E-09	43593	13	4.026E-09	8.058E-12	**5918**	100	3.855E-09	3.368E-12	6981	100
RG-10	5.373E+00	1.990E+00	50007	0	3.317E-02	1.994E-10	13911	97	3.157E-09	2.200E-11	**20202**	100
RB	5.346E-04	3.505E-08	50008	0	6.533E-03	1.224E-06	50027	0	5.449E-03	2.052E-07	50023	0
WF	8.587E-04	1.706E-05	50012	0	1.490E-01	6.966E-06	50018	0	2.425E-01	4.423E-03	50021	0
% prob_100%				50				69				**75**
nf_{avg}_100%			607				1607				3170	
nf_{avg}_all100%			**607**				1067				916	

Table 2. Results based on the use of one coordinate of the gradient, randomly selected *without replacement*.

	best_wout				center_wout				hscore_wout			
	f_{avg}	f_{min}	nf_{avg}	% s	f_{avg}	f_{min}	nf_{avg}	% s	f_{avg}	f_{min}	nf_{avg}	% s
BO	7.221E-09	4.119E-09	**521**	100	6.865E-09	3.723E-10	1723	100	6.306E-09	9.252E-10	1641	100
BP	3.979E-01	3.979E-01	**126**	100	3.979E-01	3.979E-01	794	100	3.979E-01	3.979E-01	196	100
CB6	-1.032E+00	-1.032E+00	883	100	-1.032E+00	-1.032E+00	**358**	100	-1.032E+00	-1.032E+00	487	100
DA	-2.478E+04	-2.478E+04	**753**	100	-2.478E+04	-2.478E+04	1272	100	-2.478E+04	-2.478E+04	1032	100
GP	3.000E+00	3.000E+00	**904**	100	3.000E+00	3.000E+00	1375	100	3.000E+00	3.000E+00	1518	100
HSK	-2.346E+00	-2.346E+00	**69**	100	-2.346E+00	-2.346E+00	295	100	-2.346E+00	-2.346E+00	113	100
MT	9.701E-09	9.103E-09	**1500**	100	8.286E-09	4.319E-09	2190	100	8.199E-09	2.762E-09	2127	100
MC	-1.913E+00	-1.913E+00	**103**	100	-1.913E+00	-1.913E+00	300	100	-1.913E+00	-1.913E+00	154	100
MHB	3.168E-01	8.286E-11	9682	83	5.692E-09	1.428E-10	1975	100	3.787E-09	2.452E-10	**1357**	100
NF2	3.016E-03	4.736E-05	50010	0	1.033E-02	7.602E-05	50017	0	5.700E-03	8.673E-05	50024	0
PWQ	6.014E-03	1.635E-05	50014	0	5.501E-03	9.561E-07	50025	0	5.293E-03	1.100E-06	50026	0
RG-2	4.975E-01	3.109E-09	22302	57	3.245E-09	4.320E-11	**1846**	100	4.137E-09	3.149E-11	2082	100
RG-5	1.957E+00	4.262E-09	41991	17	3.317E-02	3.006E-12	7943	97	3.762E-09	2.160E-11	**7759**	100
RG-10	6.567E+00	7.386E-09	46919	7	3.317E-02	3.264E-11	15331	97	5.592E-08	1.516E-11	22677	97
RB	2.924E-04	9.952E-09	49438	7	6.361E-03	9.378E-07	50028	0	5.017E-03	1.505E-08	50021	0
WF	9.078E-04	3.213E-06	50008	0	1.287E-01	7.751E-05	50019	0	1.794E-01	1.618E-04	50027	0
% prob_100%				50				63				**69**
nf_{avg}_100%			607				1213				1679	
nf_{avg}_all100%			**607**				1038				908	

The results reported in Table 3 aim to show that robustness has not been improved when the full gradient is used. All the values and statistics have the same meaning as in the previous tables. Similarly, the variant based on gradient computed at the best point reports the lowest nf_{avg}_all100% but also reaches the lowest % prob_100%. The use of the full gradient has deteriorated the results mostly on the variant center_full_g when compared with both center_w and center_wout.

Table 3. Results based on the use of the full gradient.

	best_full_g			center_full_g			hscore_full_g		
	f_{avg}	nf_{avg}	% s	f_{avg}	nf_{avg}	% s	f_{avg}	nf_{avg}	% s
BO	6.331E-09	**208**	100	5.862E-09	4841	100	5.397E-09	883	100
BP	3.979E-01	**185**	100	3.979E-01	2142	100	3.979E-01	294	100
CB6	-1.032E+00	**116**	100	-1.032E+00	465	100	-1.032E+00	558	100
DA	-2.477E+04	**3398**	100	-2.477E+04	33403	77	-2.477E+04	4220	100
GP	3.000E+00	**658**	100	3.000E+00	1701	100	3.000E+00	1235	100
HSK	-2.346E+00	**55**	100	-2.346E+00	636	100	-2.346E+00	72	100
MT	9.615E-09	756	100	5.753E-09	5153	100	4.442E-09	**709**	100
MC	-1.913E+00	**41**	100	-1.913E+00	1320	100	-1.913E+00	78	100
MHB	5.123E-01	10005	83	5.116E-09	2743	100	4.665E-09	**1526**	100
NF2	6.756E-03	50009	0	3.470E-02	50014	0	9.037E-03	50027	0
PWQ	1.082E-02	50011	0	6.890E-02	50020	0	5.892E-03	50020	0
RG-2	1.194E+00	39004	23	5.076E-09	6722	100	5.181E-09	**6118**	100
RG-5	1.718E+01	50013	0	4.245E+00	50018	0	4.669E+00	50015	0
RG-10	6.179E+01	50016	0	3.333E+01	50023	0	3.254E+01	50019	0
RB	5.162E-03	50006	0	2.809E-05	44037	47	3.643E-04	47933	17
WF	4.247E-01	50006	0	3.000E-01	50019	0	7.109E-01	50024	0
% prob_100%			50			56			**63**
nf_{avg}_100%		677			2858			1569	
nf_{avg}_all100%		**288**			2323			547	

Table 4 compares the results obtained with five of the above mentioned problems with those presented in [2]. The comparison involves the three tested variants center_w, hscore_w and hscore_wout, which provided the highest percentages of successful runs, 69%, 75% and 69% respectively. This table reports the values of f_{avg} and nf_{avg}, after 30 runs. We note that the herein stopping condition is the same as that of [2]. All reported variants have 100% of successful runs when solving GP, MHB, RG-2 and RG-5. However, only the variant hscore_w reaches 100% success when solving RG-10 (see last row in the table).

5 Conclusions

In this paper, we present a population-based stochastic coordinate descent method for bound constrained GO problems. Several variants are compared in order to

Table 4. Comparative results.

	results in [2]		center_w		hscore_w		hscore_wout	
	f_{avg}	nf_{avg}	f_{avg}	nf_{avg}	f_{avg}	nf_{avg}	f_{avg}	nf_{avg}
GP	3.00E+00	**833**	3.00E+00	1262	3.00E+00	1564	3.00E+00	1518
MHB	5.10E-09	**1229**	5.53E-09	1721	4.25E-09	1450	3.79E-09	1357
RG-2	3.40E-09	**1502**	4.57E-09	1505	4.16E-09	2074	4.14E-09	2082
RG-5	3.52E-09	13576	4.03E-09	**5918**	3.86E-09	6981	3.76E-09	7759
RG-10	2.65E-01	30104	3.32E-02	13911	3.16E-09	**20202**	5.59E-08	22677
(% s)		(77)		(97)		(**100**)		(97)

find the most robust, specially when difficult and larger problems are considered. The idea of using the point with highest score to generate the coordinate descent directions to move all the points of the subpopulation has shown to be more robust than the other tested ideas and worth pursuing.

Future work will be directed to include, in the set of tested problems, instances with varied dimensions to analyze the influence of the dimension n in the performance of the algorithm. Another matter is related to choosing a specified number (yet small) of gradient coordinate indices (rather than just one) by the uniform distribution on the set $\{1, 2, \ldots, n\}$, to move each point of the subpopulation.

Acknowledgments. This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Projects Scope: UID/CEC/00319/2019 and UID/MAT/00013/2013.

References

1. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. Technical Report arXiv:1606.04838v3, Computer Sciences Department, University of Wisconsin-Madison (2018)
2. Rocha, A.M.A.C., Costa, M.F.P., Fernandes, E.M.G.P.: A stochastic coordinate descent for bound constrained global optimization, AIP Conference Proceedings, **2070**, pp 020014 (2019)
3. Kvasov, D.E., Mukhametzhanov, M.S.: Metaheuristic vs. deterministic global optimization algorithms: The univariate case. Appl. Math. Comput. **318**, 245–259 (2018)
4. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**(2), 341–362 (2012)
5. Wright, S.J.: Coordinate descent algorithms. Math. Program. Series B, **151**(1), 3–34 (2015)
6. Liu, H., Xu, S., Chen, X., Wang, X., Ma, Q.: Constrained global optimization via a DIRECT-type constraint-handling technique and an adaptive metamodeling strategy. Struct. Multidisc. Optim. **55**(1), 155–177 (2017)
7. Ali, M.M., Khompatraporn, C., Zabinsky, Z.B.: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J. Glob. Optim. **31**(4), 635–672 (2005)