A Stochastic Coordinate Descent for Bound Constrained Global Optimization

Ana Maria A.C. Rocha[1,a)], M. Fernanda P. Costa[2,b)] and Edite M.G.P. Fernandes[1,c)]

[1] *Algoritmi Research Centre, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal.*
[2] *Centre of Mathematics, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal.*

a) Corresponding author: arocha@dps.uminho.pt
b) mfc@math.uminho.pt
c) emgpf@dps.uminho.pt

Abstract. This paper presents a stochastic coordinate descent algorithm for solving bound constrained global optimization problems. The algorithm borrows ideas from some stochastic optimization methods available for the minimization of expected and empirical risks that arise in large-scale machine learning. Initially, the algorithm generates a population of points although only a small subpopulation of points is randomly selected and moved at each iteration towards the global optimal solution. Each point of the subpopulation is moved along one component only of the negative gradient direction. Preliminary experiments show that the algorithm is effective in reaching the required solution.

INTRODUCTION

In this paper, we consider the problem of finding a global solution of a bound constrained nonlinear optimization problem in the following form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega, \end{array} \tag{1}$$

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, is 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 global optimization (GO) problem shown in (1), a stochastic or a deterministic method may be selected. A stochastic method is able to provide a solution that may not be globally optimal. On the other hand, a deterministic method provides an interval that contains the global optimal solution, but requires in general a much larger computational effort [1]. Techniques to find a global optimal solution rely on two search procedures. The exploration procedure aims to diversify the search so that a global optimal solution is located; the exploitation procedure is devoted to intensify the search in a vicinity of a promising region so that a good approximation is computed. Approximate methods or heuristics are designed to generate good solutions with less computational effort and time than the more classical algorithms. Stochastic 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.

Recent and promising optimization methods for large-scale machine learning make use of classical gradient-based methods, like the full gradient, accelerated gradient, conjugate gradient and quasi-Newton, classified as batch approaches. On the other hand, stochastic gradient approaches have relied on intuitive schemes to employ data information more efficiently than the batch methods. It has been recognized that working with small data samples can avoid redundancy and be quite appealing [2, 3, 4]. The coordinate descent method (CDM) is one of the oldest methods in optimization. It is very popular for its simplicity and it has been applied in a variety of practical situations. Cyclic coordinate search, block coordinate descent and random coordinate search are known variants of the CDM and differ mainly on the choice of the component(s) of the gradient for the update of the variables. Convergence issues for convex objective f are clear and well defined for some variants of the CDM [5].

This paper is devoted to the use of a variant of the CDM in the context of a stochastic population-based method for solving bound constrained GO problems that may have a large number of variables. The new stochastic CDM has started to be tested and the numerical results seem promising.

STOCHASTIC COORDINATE DESCENT FRAMEWORK

This section presents the motivation for the use of the stochastic coordinate descent method in the context of a population-based method for solving the problem (1).

Coordinate Descent Method The CDM operates by taking steps along the coordinate directions [2, 5]. Hence, the search direction for minimizing f from the iterate $x^{(k)}$, at iteration k, is defined as

$$d^{(k)} = -\nabla_{i_k} f(x^{(k)}) e_{i_k} \text{ with } \nabla_{i_k} f(x) = \frac{\partial f}{\partial x_{i_k}}(x) \tag{2}$$

where 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, $\alpha^{(k)}$, the new approximation, $x^{(k+1)}$, is computed as shown below and differ from $x^{(k)}$ only in the i_kth component:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}. \tag{3}$$

Stochastic Coordinate Descent Method Stochastic gradient methods and stochastic CDM (S-CDM) have been attracting the attention of the scientific community because of their usefulness in data analysis and machine learning. Applications to practical problems of the CDM are varied being the support vector machine problem the most known application of S-CDM, see [6] and references therein. In the S-CDM, the index i_k to define the search direction and the component of $x^{(k)}$ to be adjusted (see (2) and (3)) is chosen randomly by uniform distribution on $\{1, 2, \ldots, n\}$ *with replacement* in each iteration. If the random choice is made *without replacement*, a set of n consecutive iterations defines an "epoch" and at the start of each epoch, the set $\{1, 2, \ldots, n\}$ is shuffled [6]. As shown above the approximation $x^{(k)}$ is moved along the direction corresponding to the component of the gradient with index i_k. We note here that the direction shown in (2) might not be a negative directional derivative for f from $x^{(k)}$.

Population-Based Stochastic Coordinate Descent Method At each iteration of a population-based algorithm, a set of candidate solutions/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 ith point ($i \in P = \{1, 2, \ldots, |P|\}$). 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. We now describe an S-CDM adapted to a population-based framework, herein coined by P-S-CDM. To prevent the computational burden of moving and evaluating the full population, without spoiling the exploration capability of the algorithm, a subpopulation of points is randomly selected to be potentially moved in direction to the global optimum. In practical terms, the indices of the subpopulation are randomly selected by a uniform distribution on $\{1, 2, \ldots, |P|\}$ *without replacement*. Let $P_1, P_2, \ldots, P_k, \ldots$ be the sets of indices of the subpopulation randomly chosen from P. After movement, the best point of each subpopulation will be maintained for the next iteration. We note that, for $k = 1$, P_1 includes the index of the best point, while for the subsequent iterations, P_k does not include the index of the best point of the previous subpopulation. Hence, for $k = 1$ the size of the population is $|P_1|$, and for $k > 1$, the size is $|P_k| + 1$. The rules to generate the subsets of indices, from the set P, for the subpopulations are the following:

r1: $P_1 \subset P$ and $P_k \subset P \backslash \{j_{k-1}\}$ for $k > 1$;
r2: $|P_1| \ll |P|$;
r3: $|P_2| + 1 \leq |P_1|$ and $|P_{k+1}| \leq |P_k|$ for $k > 1$;

where j_{k-1} is the index of the best point of the subpopulation of iteration $k-1$. We use $\bar{x}$, the central point, to represent the region defined by the points of a subpopulation, which in the context of our algorithm is defined as follows:

$$\bar{x}^{(1)} = \frac{1}{|P_1|} \sum_{j \in P_1} x^j \text{ and } \bar{x}^{(k)} = \frac{1}{|P_k| + 1} \left(\sum_{j \in P_k} x^j + x^{j_{k-1}} \right) \text{ for } k > 1. \tag{4}$$

We define $P_1^+ = P_1$ and $P_k^+ = P_k \cup \{j_{k-1}\}$ for $k > 1$. Each point of the subpopulation, x^j ($j \in P_k^+$), is moved by $x^j = x^j + \alpha^{kj} d^{(k)}$ if $d^{(k)}$ is a descent direction for f from x^j, where α^{kj} is a positive step size computed by a backtracking strategy; otherwise, the point x^j is not moved. Whenever the new position of the point falls outside the bounds, a projection onto Ω is carried out. For each $j \in P_k^+$, the search direction is along a component of the gradient at the central point, with the index i_k randomly selected by uniform distribution on $\{1, 2, \ldots, n\}$ one at a time for each *j with replacement*:

$$d^{(k)} = -\nabla_{i_k} f(\bar{x}^{(k)}) e_{i_k}. \tag{5}$$

```
Input: $\varepsilon$, $N.F._{\max}$, $\mu$
Output: $x^{j_k}$, $f(x^{j_k})$
Set $k = 1$ and $N.F. = 0$
Randomly generate a set of $|P|$ points in $\Omega$, $\{x^1, x^2, \ldots, x^{|P|}\}$
while stopping condition is not satisfied  and  $N.F. \leq N.F._{\max}$ do
    Randomly select a subset of indices $P_k$ from $P$ following rules r1, r2 and r3 above
    if $k = 1$ then
        Set $P_k^+ = P_k$
    else
        Set $P_k^+ = P_k \cup \{j_{k-1}\}$
    Compute $\bar{x}$  according to (4)
    for all $j \in P_k^+$ do
        Randomly choose $i_k$ by uniform distribution on $\{1, 2, \ldots, n\}$
        Compute the search direction $d^{(k)}$ according to (5)
        if $\nabla_{i_k} f(x^j) d_{i_k}^{(k)} < 0$ then
            if a descent direction for $f$ from $x^j$ has not been computed before then
                Compute $f(x^j)$; Set $N.F. = N.F. + 1$
            Set $m = 0$ and $N.F. = N.F. + 1$; Compute $f(x^j + (0.5)^m d^{(k)})$
            while $f(x^j + (0.5)^m d^{(k)}) > f(x^j) + \mu (0.5)^m \nabla_{i_k} f(x^j) d_{i_k}^{(k)}$  and  $(0.5)^m >$ 1e-8 do
                Set $m = m + 1$ and $N.F. = N.F. + 1$
            if $(0.5)^m >$ 1e-8 then
                Set $x^j = x^j + (0.5)^m d^{(k)}$ and $f(x^j) = f(x^j + (0.5)^m d^{(k)})$
    Set $j_k = \arg\min\{f(x^j), j \in P_k^+\}$
    Set $k = k + 1$
```

Algorithm 1: Population-based stochastic coordinate descent algorithm

Although using only one component of the gradient to adjust each point x^j of the population seems a weakness in contrast with using the full gradient, it is in fact a strength since in a population-based environment (with a variety of candidate solutions at each iteration) the likelihood is that all components of the gradient end up to be used to move the points. The smaller the number of variables the greater is the probability that all components are selected. We also note that, even if the direction with the full gradient is not a descent direction for f from x^j, the selected coordinate direction could be and some progress is achieved along this direction. Algorithm 1 shows the main steps of the algorithm. The stopping condition to guarantee a solution in the vicinity of f^* is

$$|f(x^{j_k}) - f^*| \leq \varepsilon |f^*| + \varepsilon^2 \tag{6}$$

where x^{j_k} is the best point of the subpopulation, $j_k = \arg\min\{f(x^j), j \in P_k^+\}$. However, if (6) is not satisfied for a given tolerance, $\varepsilon > 0$, the algorithm stops after a specified number of function evaluations, $N.F._{\max}$.

PRELIMINARY EXPERIMENTS AND CONCLUSION

For a preliminary practical validation of the proposed P-S-CDM five well-known benchmark problems are used: GP (Goldstein and Price), MHB (Modified Himmelblau), RA-2, RA-5, RA-10 (Rastrigin $n = 2$, $n = 5$, $n = 10$ respectively), described in Table 1 of [7]. The algorithm is implemented in the Matlab$^{\text{TM}}$ (Matlab is a registered trademark of the MathWorks, Inc.) programming language. The selected parameter values for the algorithm are $|P| = 500$, $|P_1| = 0.01|P|$, $|P_k| = |P_1| - 1$ for all $k > 1$, $\varepsilon = 10^{-4}$, $N.F._{\max} = 50000$ and $\mu =$1e-3.

We compare our results with other population-based methods from the literature using the same stopping condition, mEM (modified electromagnetism-like mechanism based on memory force vector), mAFS-P (priority-based modified artificial fish swarm), mDE (modified differential evolution with mixing mutation), mGA (modified genetic algorithm with diversity preserving mechanism) and the results are taken from [7]. Table 1 shows the average of the obtained minimum f values, f_{avg}, after 30 runs, and the average number of function evaluations, $N.F._{avg}$.

TABLE 1. Comparative results.

	Algorithm 1		**mEM**		**mAFS-P**		**mDE**		**mGA**	
	f_{avg}	$N.F._{avg}$	f_{avg}	$N.F._{avg}$	f_{avg}	$N.F._{avg}$	f_{avg}	$N.F._{avg}$	f_{avg}	$N.F._{avg}$
GP	3.00e+00	833	3.00e+00	357	3.00e+00	1760	3.00e+00	509	3.00e+00	795
MHB	5.10e-09	1229	1.35e-10	855	2.93e-10	1882	4.29e-09	1010	3.14e-09	7448
RA-2	3.40e-09	1502	1.03e-09	3490	1.22e-09	4017	4.14e-09	1057	1.50e-09	7965
RA-5	3.52e-09	13576	6.74e-09	13582	1.85e-09	8890	6.41e-09	8130	6.01e-09	35458
RA-10	2.65e-01	30104	2.69e-10	14143	6.30e-01	36198	7.94e-09	43560	2.39e-06	50000

The preliminary experiments show that the proposed P-S-CDM is effective in reaching the required solution. In the comparisons with other stochastic population-based methods, our algorithm has 100% of successful runs (according to (6)) in four of five tested problems. It wins in efficiency over mAFS-P on GP, over mAFS-P and mGA on MHB, over mEM, mAFS-P and mGA on RA-2 and over mEM and mGA on RA-5. When solving RA-10, our algorithm reaches a solution with the required accuracy (according to (6)) on 77% of the runs (mEM reports 100%, mAFS-P reports 38%, mDE 100% and mGA 0%). We also note that both mEM and mAFS-P implement a local search starting from the best solution at each iteration aiming to enhance the quality of the obtained solutions.

Future developments will focus on improving the convergence of our proposal by using a self-adaptive strategy to define the size of each subpopulation and a new strategy to sample a block of components of the gradient to move points of the subpopulation. Classes of randomly generated test problems [8] will be used for a comparison between deterministic descent methods, our improved algorithm and other stochastic metaheuristics via operational zones [9, 10]. The convergence for nonconvex objective functions will be also analyzed.

ACKNOWLEDGMENTS

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT - Fundação para a Ciência e Tecnologia within the projects UID/CEC/00319/2013 and UID/MAT/00013/2013.

REFERENCES

[1] D. E. Kvasov, M. S. Mukhametzhanov, Appl. Math. Comput. **318**, 245–259 (2018).

[2] L. Bottou, F. E. Curtis and J. Nocedal, “Optimization methods for large-scale machine learning”, Technical Report arXiv:1606.04838v3, Computer Sciences Department, University of Wisconsin-Madison 2018.

[3] A. Beck and L. Tetruashvili, SIAM J. Optim. **23**, 2037–2060 (2013).

[4] C.-p. Lee and S.J. Wright, “Random permutations fix a worst case for cyclic coordinate descent”, Technical Report arXiv:1607.08320v4, Computer Sciences Department, University of Wisconsin-Madison 2018.

[5] Y. Nesterov, SIAM J. Optim. **22**, 341–362 (2012).

[6] S. J. Wright, Math. Program. Series B, **151**, 3–34 (2015).

[7] I. A. C. P. Espírito Santo, L. Costa, A. M. A. C. Rocha, M. A. K. Azad and E. M. G. P. Fernandes, “On challenging techniques for constrained global optimization”, in Handbook of Optimization edited by I. Zelinka, V. Snášel and A. Abraham, ISRL vol 38, 641–671, Springer-Verlag, Berlin Heidelberg, 2013.

[8] M. Gaviano, D. E. Kvasov, D. Lera, Y. D. Sergeyev, ACM Trans. Math. Software **29**, 469–480 (2003).

[9] Y. D. Sergeyev, D. E. Kvasov, M. S. Mukhametzhanov, Math. Comput. Simul. **141**, 96–109 (2017).

[10] Y. D. Sergeyev, D. E. Kvasov, M. S. Mukhametzhanov, Sci. Rep-UK, **8**, art. n. 453 (2018).