Proceedings of GOW 2012, pp. 1 – 4.

A global derivative-free optimization method for expensive functions with bound constraints

Eugenio Echagüe[1], Frédéric Delbos[2] and Laurent Dumas [3]

[1]*IFPEN, 1 et 4 Avenue du Bois Préau, 92500 Rueil-Malmaison, France,* eugenio.echague@ifpen.fr

[2]*IFPEN, 1 et 4 Avenue du Bois Préau, 92500 Rueil-Malmaison, France,* frederic.delbos@ifpen.fr

[3]*Université de Versailles S.Q. , 45 Avenue des Etats-Unis, 78035 Versailles, France,* laurent.dumas@uvsq.fr

Abstract In this article we propose a new method in order to solve a general black-box global optimization problem where function evaluations are expensive. Our work was motivated by many problems in the oil industry, coming from several domains like reservoir engineering, molecular modeling, engine calibration and inverse problems in geosciences. Even if evolutionary algorithms are often a good tool to solve these problems, they sometimes need too many function evaluations, especially in high-dimension cases. To overcome this difficulty, we propose here a new approach, called GOSGrid, using as surrogate model the Sparse Grid interpolation method with a refinement process .

Keywords: Global optimization, Expensive functions, Surrogate models, Sparse Grid interpolation.

1. Introduction

In the context of oil industry, many problems consist in a global minimization of a computationally expensive function with bound constraints ([1]):

$$\begin{gathered}\text{Find } x^* = \operatorname{argmin} f(x) \\ x_l \leq x \leq x_u \\ x \in \mathbb{R}^n\end{gathered}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is the computationally expensive function and x_l, $x_u \in \mathbb{R}^n$.

The values of f are in general the output of a complex simulator for which we don't have an explicit expression. The absence of any information on the function gradient narrows the resolution field to algorithms using no first or second order derivatives. There exists many different approaches in derivative free optimization, among which the most popular are direct search methods like Nelder Mead or MADS ([2]) and evolutionary algorithms like genetic algorithms ([3]), evolution strategies or particle swarm optimization (see [4] for a review of DFO methods). Unfortunately, all these approaches may exhibit a slow convergence behavior and thus be very expensive.

The use of a surrogate model is well suited for the type of optimization considered here. A surrogate model is a framework used to minimize a function by sequentially building and minimizing a simpler model (surrogate) of the original function. A widely used form of surrogate models consists of linear combinations of basis functions, for instance Radial Basis Functions ([5]) or Kriging. In general, the more points used when creating an interpolation model, the more accurate is the approximation.

In this work, we construct a new surrogate model by using the Sparse Grid interpolation method. Basically, the Sparse Grid approach is a hierarchical Lagrange approximation method

which neglects the basis functions with the smallest supports. This approach was introduced in 1963 by Smolyak ([6]) in order to approximate integrals in high dimensions. It was applied for PDE approximations and more recently for sensitivity analysis ([7]) and optimization ([8]). Compared to the approach in [8], a local refinement is constructed here in order to explore the more promising regions. The Sparse Grid interpolation method is recalled in section 2 whereas the new global optimization method is presented and applied for analytical test functions in section 3.

2. The Sparse Grid interpolation method

The Sparse Grid interpolation method uses Lagrange polynomials on the Chebyshev points as basis functions in dimension one. The extension to dimension d is done by simply tensoring the formulas obtained in dimension one. The hierarchical approach and the sparsity principle are respectively presented in the first two subsections. The refinement process is then described in subsection 2.3.

2.1 The 1-D case

For $i \in \mathbb{N}$, we call X^i the set of the Chebyshev points in the interval $[0,1]$ of level i. These sets have the property that $X^i \subset X^{i+1}$. If we denote by a_j^i the Lagrange polynomial associated to each $x_j^i \in X^i$, the interpolation model of level i of f, called $m_i(f)$, is equal to

$$m_i(f) = \sum_{x_j^i \in X^i} f(x_j^i) a_j^i. \tag{1}$$

Define Δ^k as the difference between two consecutive models, then:

$$\Delta^k = m_k(f) - m_{k-1}(f) = \textstyle\sum_{x_j^k \in X^k} (f(x_j^k) - m_{k-1}(f)(x_j^k)) \ . \ a_j^k.$$

If we set $X_\Delta^k = X^k \backslash X^{k-1}$, as $X^{k-1} \subset X^k$ we get

$$\Delta^k = \sum_{x_j^k \in X_\Delta^k} \underbrace{(f(x_j^k) - m_{k-1}(f)(x_j^k))}_{w_j^k} \ . \ a_j^k.$$

It means that for computing Δ^k we only evaluate the function on the points that don't belong to the previous level sets. The telescopic sum principle and $m_0(f) = 0$ give us:

$$m_i(f) = \sum_{k=1}^{i} \Delta^k.$$

Thus, in order to get the approximation of level $i+1$ we only need to compute the function values at the new interpolation points X_Δ^{i+1}.

2.2 The general case

For $k = 1, \ldots, d$ let X^{i_k} be a set of Chebyshev points of some level i_k. By simply tensoring (1) we get the Lagrange interpolation formula on the set $\prod_{k=1}^d X^{i_k}$ as:

$$m_{(i_1,i_2,\ldots i_d)}(f) = \sum_{x_{j_1}^{i_1} \in X^{i_1}} \cdots \sum_{x_{j_d}^{i_d} \in X^{i_d}} f(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i_d})(a_{j_1}^{i_1} \otimes \ldots \otimes a_{j_d}^{i_d}).$$

With the same hierarchical approach done in dimension 1, we get

$$m_{(i_1,i_2,\ldots i_d)}(f) = \sum_{k_1=1}^{i_1} \ldots \sum_{k_d=1}^{i_d} (\Delta^{k_1} \otimes \ldots \otimes \Delta^{k_d}).$$

If we only apply the sum on the indexes $k = (k_1, \ldots, k_n)$ such that $|k|_1 \leq d + N - 1$ we get the Sparse Grid interpolation formula of level N, which neglects the smallest support basis functions:

$$SG_N(f) = \sum_{|k|_1 \leq d+N} (\Delta^{k_1} \otimes \ldots \otimes \Delta^{k_d}).$$

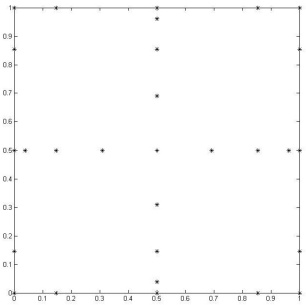

Figure 1. Sparse grid in 2-D for N=4. Number of points: 29

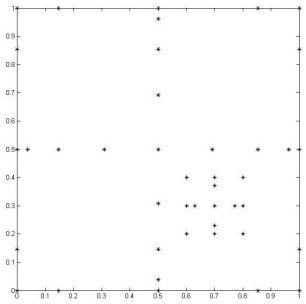

Figure 2. Sparse grid in 2-D for N=4 and a refinement process of order 3. Number of points: 42

The interpolation points of a Sparse Grid of level $N = 4$ in dimension 2 is depicted on Figure 1. Compared to a full grid of the same level which would contain 81 points, it is only made of 29 points. More generally, it can be proven that for a sufficently smooth function f, the approximation of f by a Sparse Grid model is of order $O(N^{-2}(\log N)^{d-1})$ with only $O(N(\log N)^{d-1})$ points.

2.3 The refinement process

In a global optimization problem, we need to explore the whole domain but we also need sometimes to focus our attention to special promising zones (the exploitation phase). To do so, a refinement process of the Sparse Grid interpolation model is used to construct a new and more precise model around a promising point. A local Sparse Grid model interpolates the error function in this area (see Figure 2) and can thus locally improve the current global model.

3. Global Optimization with a Sparse Grid model (GOSGrid)

Given a maximal number of function evaluations and bound constraints, the new optimization method, called GOSGrid, sequentially minimizes Sparse Grid models of the objective function f. In particular, during the process, it refines some zones in order to get a better solution. The hierarchical principle allows us to improve the global model without throwing away the previous one whereas the sparsity greatly reduces the number of exact evaluations, especially in high dimensions.

Starting from a given hierarchical level, the algorithm constructs a global model in the area described by the bound constraints. As the evaluation of the model is computationally inexpensive, a local multistart algorithm is run in order to find a global minimizer of the model. We then compare the function value at this point with the lowest value of the function at the grid and we keep the best of them. Then, we iteratively refine the global model in a hypercube centered at this point, and minimize with the same multistart algorithm the improved model. The construction of the next Sparse Grid level is done when a criteria based on the number of evaluations needed to perform the refinement or a relative rate of decrease is fulfilled.

Figure 3 gives a comparison between the GOSGrid method (continuous line) and a evolutionary algorithm, namely an evolution strategy with a cumulative step length adaptation, for the Michalewicz function in dimension $d = 5$.

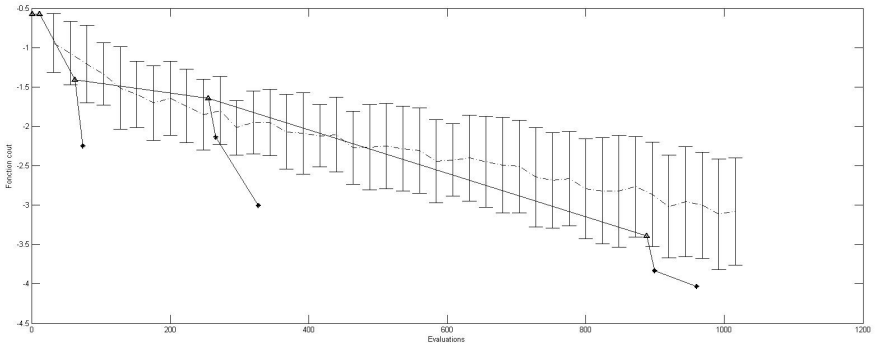

Figure 3. Comparison between GOSGrid (cont.) and an evolutionary algorithm (mean and standard deviation) for the Michalewicz function.

The refinement process corresponds to the small decreasing branches starting from the continuous line (here for the levels $N = 3$ to $N = 5$). For a given number of evaluations, the cost function value for the evolutionary algorithm is higher than the corresponding one for the GOSGrid approach. The figure shows the importance of the refinement step as the best points are found after the refinement process. Other tests in higher dimension, which couldn't be included in the text due to space limitation, lead to the same conclusion.

4. Conclusions and perspectives

We present here a new global optimization tool for expensive functions, called GOSGrid. It is based on the Sparse Grid interpolation method with a Sparse Grid refinement process. The hierarchical construction of the surrogate model and its sparsity allows to reach a faster decrease, in terms of cost function evaluations, compared to a classical evolutionary algorithm. The next step will consist to improve the refinement strategy by making it more adaptive and efficient (taking into account more promising points at the same time, for example) and to apply GOSGrid on real reservoir engineering cases.

References

[1] Oliver, Dean; Reynolds, Albert; Liu, Ning (2008). Inverse theory for petroleum characterization and history matching . Cambridge: Cambridge University Press.

[2] Audet, Charles; J.E. Dennis Jr (2006). Mesh adaptive direct search algorithms for constraint optimization. Philadelphia: SIAM J.Optim, Vol 17, pp. 188-217.

[3] Goldberg, Dave (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Adison Wesley.

[4] Conn, Andrew; Scheinberg, Katya; Vicente, Luis (2009). Introduction to derivative-free optimization. Philadelphia: MPS-SIAM Series on Optimization.

[5] Dumas, Laurent; Herbert, Vincent; Muyl, Frédérique (2005). Comparison of global optimization methods for drag reduction in the automotive industry, Lecture Notes in Computer Science, Vol. 3483, pp. 948-957.

[6] Smolyak, S.A. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions, Soviet. Math. Dokl., Vol.4, pp. 240-243.

[7] Causse, Mathieu (2010). Sparse Grid: de l'approximation des EDP au calcul de sensibilité. Toulouse: Ph.D.

[8] Buzzard, G.; Donahue, M.; Rundell, A. (2009). Parameter Identification with Adaptive Sparse Grid-based Optimization for Models of Cellular Processes, Methods in Bioengineering: Systems Analysis of Biological Networks, A. Jayaraman and J. Hahn, Eds. Boston/London: Artech House, pp. 211-232.