CSMA 2017
13ème Colloque National en Calcul des Structures
15-19 Mai 2017, Presqu'île de Giens (Var)

Continuous Global Optimization Using A Guide Function Approach

H. Gao[1,2], P. Breitkopf[2], M. Xiao[3]

[1] *Université Libre de Bruxelles, Huanhuan.Gao@ulb.ac.be*
[2] *Université de Technologie de Compiègne, piotr.breitkopf@gmail.com*
[3] *Northwestern Polytechnical University, manyuxiao@nwpu.edu.cn*

Résumé — Nowadays, gradient-based methods have achieved big success in continuous optimization problems, but most of them focus on the local searching process, thus the global solutions of the problems may be lost. The main difficulty of global searching is how to find a solution which is better than the current local solution obtained by gradient-based methods. In this paper, we propose a guide function approach which can help find better solutions when a local solution has already been found. A classical numerical test is executed using the proposed approach to show its validity, and in further development, this approach will be extended for categorical optimization.
Mots clés — Global Optimization, Continuous Problem, Guide Function.

1 Introduction

Continuous optimization problems are always be tackled with local searching methods, especially gradient-based algorithms (e.g. steepest descent, conjugate gradient methods), when both objective and constraint functions are continuous and differentiable. In this case, local solutions can always be found. To purse higher goals, global optimization remains a tough issue. General evolutionary methods, including genetic algorithms [**?**], evolutionary strategy [**?**], are always used to obtain the global solution. But they may confront the optimization efficiency issue. Different from evolutionary algorithms, the filled function method proposed in [**?**, **?**, **?**] is applied to construct a path which can help optimize the objective function when a local solution has been found by using gradient-based searching. In their papers, the authors recommend that a delicate parameter setting for the filled function is the critical for successfully obtaining the global solution. Another method for global optimization is called the tunnelling methods [**?**, **?**, **?**]. The tunnelling methods focus on digging a horizontal tunnel out of the function valley when the gradient-based methods have found a local solution, thus may help find a better solution step by step. In this paper, the main context are as follows : in section 2, we formulate the problem definition ; in section 3, we propose our approach for global optimization ; then In section 4, a numerical example is tested ; and in the last, we draw some conclusions.

2 Problem formulation

This paper deals with finding a global minimizer of a at least twice continuously differentiable function

$$f(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, x_2, \cdots, x_n). \tag{1}$$

We also assume

$$f(\boldsymbol{x}) \to +\infty, \quad \parallel \boldsymbol{x} \parallel \to +\infty. \tag{2}$$

So there exists a subspace of the whole design space which contains all the local minimizers. We name this closed boundary subspace as Ω in which every design variable x_i satisfies

$$x_i \in [x_i^{min}, x_i^{max}], \quad i = 1, 2, \cdots, n. \tag{3}$$

So $f(\boldsymbol{x})$ has no minimizer along its design boundaries. In this paper, we assume this subspace Ω is known in advance, and we handle the global optimization process in Ω.

3 Guide function approach

In this section, the guide function approach for finding the global solution is introduced. The guide function is a combination of base function, Heaviside function, compress function and background function. They will be explained step by step.

3.1 The guide function

Before we put forward the guide function approach, we assume one has found a local minimizer by local searching method, and we mark this local minimizer as $\boldsymbol{x}_1^*$. Then the base function is written as :

$$BF(\boldsymbol{x}) = (f(\boldsymbol{x}_1^*) - f(\boldsymbol{x}) - \varepsilon) \tag{4}$$

where ε is a small positive value and it makes $BF(\boldsymbol{x}_1^*) < 0$. For any given $\boldsymbol{x}$, if $f(\boldsymbol{x})$ is larger than $f(\boldsymbol{x}_1^*) - \varepsilon$, then $BF(\boldsymbol{x})$ is negative, and vice versa.
Then the continuous Heaviside function is introduced to penalize the base function

$$Hs(BF(\boldsymbol{x})) = k_1 arctan(BF(\boldsymbol{x}))/\pi. \tag{5}$$

We can see that if $BF(\boldsymbol{x}) < 0$, $Hs(BF(\boldsymbol{x}))$ will be about $-k_1/2$; while if $BF(\boldsymbol{x}) > 0$, $Hs(BF(\boldsymbol{x}))$ will be about $k_1/2$. k_1 is an adjusting parameter, and is given as positive.
In the third step, the compress function is used to compress the smaller peak regions of $Hs(BF(\boldsymbol{x}))$

$$CF(Hs(BF(\boldsymbol{x}))) = k_2 N^{Hs(BF(\boldsymbol{x}))}(Hs(BF(\boldsymbol{x}))+1). \tag{6}$$

where N denotes a big positive number, in our test, N is set as 50. k_2 is used to adjust the magnitude of the compress function. Finally, the guide function is written as

$$GF(\boldsymbol{x}, \boldsymbol{x}_1^*) = P(\boldsymbol{x}, \boldsymbol{x}_1^*) - CF(Hs(BF(\boldsymbol{x}))) \tag{7}$$

where $P(\boldsymbol{x})$ is the background function which is written as

$$P(\boldsymbol{x}, \boldsymbol{x}_1^*) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}_1^*\|^2}{2\rho^2}} \tag{8}$$

where ρ is a positive value. The guide function holds the advantage that only the local minimizers which are smaller than $f(\boldsymbol{x}_1^*)$ are remained in this function, the other local minimizers are vanished due to the compress function.

3.2 Optimization strategy

The global optimization process can be listed as follows :
Step 1 : Set l and m as 1, n is the number of design variables.
Step 2 : Use gradient-based methods to find the first local minimizer $\boldsymbol{x}_m^*$;
Step 3 : If $l = n$, go to Step 6; else we make $\boldsymbol{x}_0 = \boldsymbol{x}_m^* + t\boldsymbol{e}_l$. t is the step length and $\boldsymbol{e}_l$ indicates the unit vector in which $\boldsymbol{e}(l) = 1$.
Step 4 : Optimize $GF(\boldsymbol{x}, \boldsymbol{x}_m^*)$ from the initial point $\boldsymbol{x}_0$ until find a solution $\boldsymbol{x}_t$ which satisfies $f(\boldsymbol{x}_t) < f(\boldsymbol{x}_m^*)$ or $\| f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t-1} \| < \tau$, τ is a small positive value.
Step 5 : If $f(\boldsymbol{x}_t) < f(\boldsymbol{x}_m^*)$, then $m = m+1$, use $\boldsymbol{x}_t$ as the initial design and go to step 2; if $\| f(\boldsymbol{x}_t) - f(\boldsymbol{x}_m^* \| < \tau$, $l = l+1$, go to step 3.
Step 6 : Global optimization converges and output optimization results.

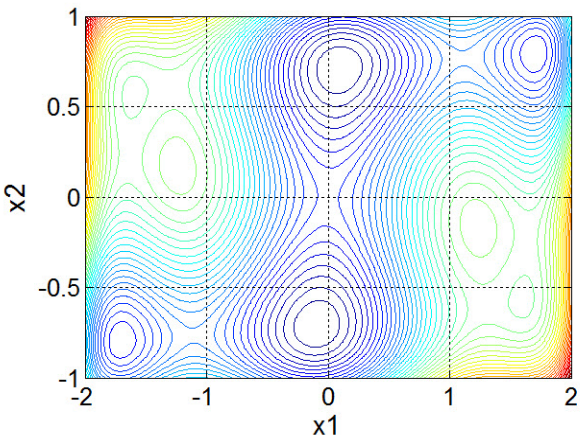

FIGURE 1 – The six-hump camel back function

4 Numerical test

In this section the classical six-hump camel function is tested for the guide function approach. The six-hump camel function is written as :

$$f(x_1,x_2) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 - x_1x_2 - 4x_2^2 + 4x_2^4 \tag{9}$$

As shown in Fig. **??**, the camel function contains six local minimizers. Among them the global minimizer is $\boldsymbol{x}^* = (0.0898, -0.7126)$ or $\boldsymbol{x}^* = (-0.0898, 0.7126)$ with the global minimum $f(\boldsymbol{x}^*) = -1.0316$. We start the searching from an initial design point $\boldsymbol{x}_0 = (-2,-1)$. By using BFGS methods, we can find the first local minimizer $\boldsymbol{x}_1^* = (-1.7037, -0.7957)$ with its objective value $f(\boldsymbol{x}_1^*) = -0.21546$. In the next step we construct the Guide function. Firstly, the base function is shown as in Fig. **??**.

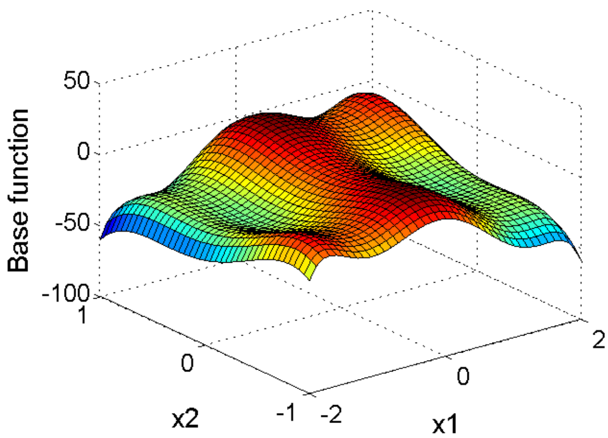

FIGURE 2 – The base function

Then the base function is penalized by Heaviside function, as displayed in Fig. **??**. In Fig. **??**, we can see that the original minimizers become maximizers.
In order filter the unobtrusive minimizers, we introduce the compress function. Its shape is shown in Fig. **??**.
We use the difference of the background function and the compress function to obtain the guide function (Fig. **??**). We continue to optimize on the surface of the guide function. By using the BFGS method, it gives a descending line as shown in Fig. **??**, and then we obtains the global minimizer finally, as shown in Fig. **??**. After we have found a new better minimizer, we continue using the global approach, trying to find a better one. We reconstruct the guide function, and it is illustrated in Fig. **??**. After the trying

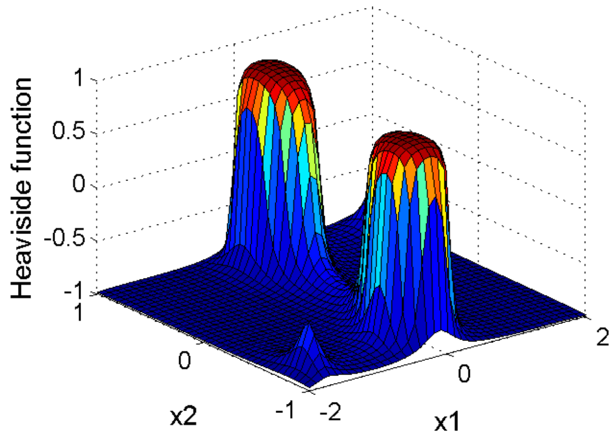

FIGURE 3 – The Heaviside function

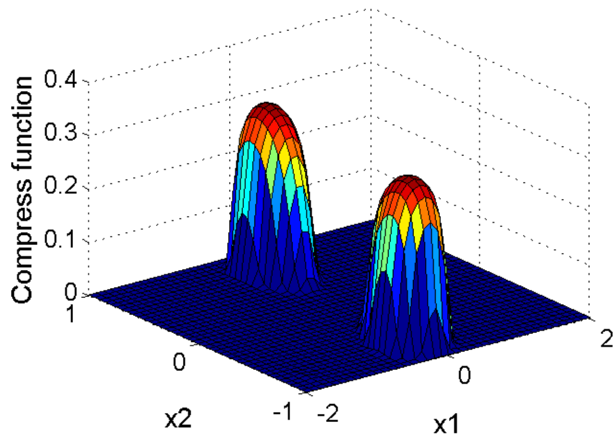

FIGURE 4 – The compress function

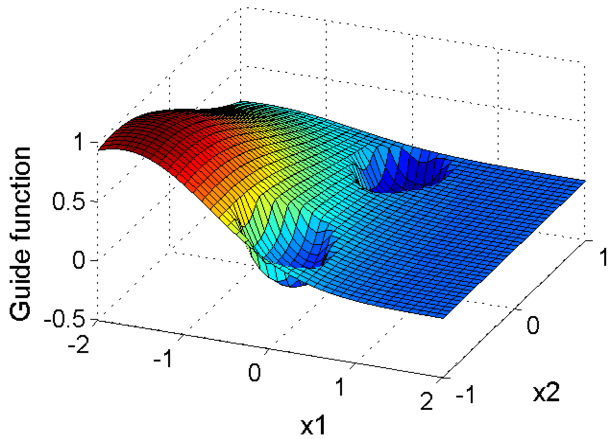

FIGURE 5 – The guide function

searches, no better solution can be obtained, proving that we probably have found the global minimizer. The objective history versus iteration number is shown in Fig. **??**.

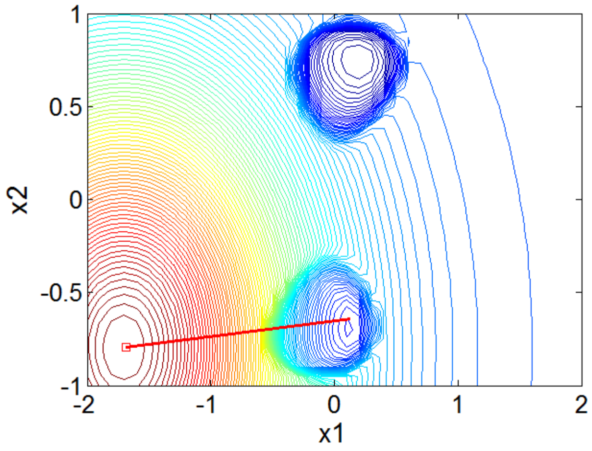

FIGURE 6 – The guide search line in the guide function

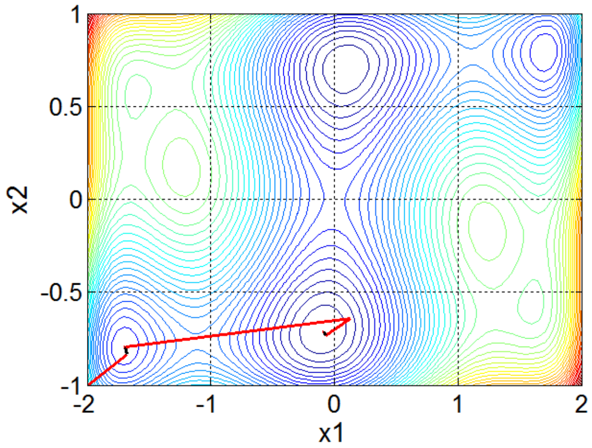

FIGURE 7 – The optimization process

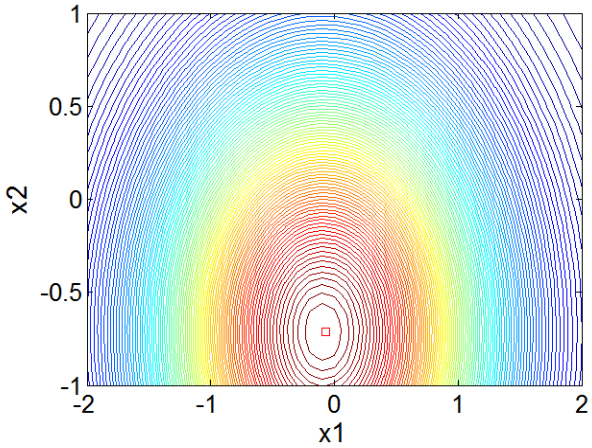

FIGURE 8 – The 2nd guide function

5 Conclusion

In this paper, a guide function approach, consisting of the base function, the Heaviside function, the compress function and the background function, is proposed to solve continuous global optimization pro-

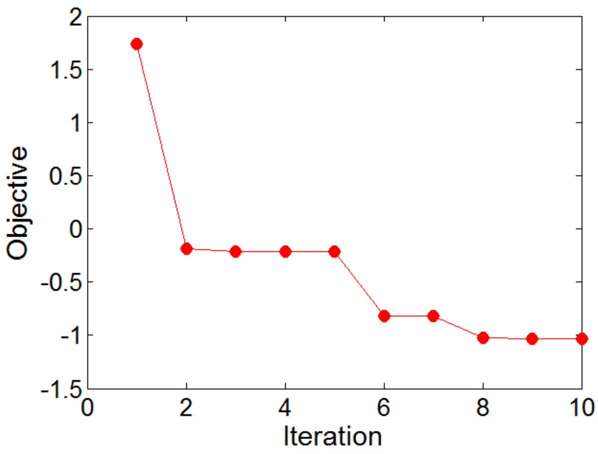

FIGURE 9 – The objective history

blems. The six-hump camel function proves that the guide function can automatically filter the designs, eliminating those local minimizers which are worse than previous minimizer.
In further development, this proposed approach will be compared with the filled function method and other global optimization methods. Then it will be tested for categorical optimization problems.

Références

[1] L. Davis. *Handbook of genetic algorithms*, 1991.

[2] D. Mester, Y. Ronin, D. Minkov, E. Nevo, A. Korol. *Constructing large-scale genetic maps using an evolutionary strategy algorithm*, Genetics, 165(4), 2269-2282, 2003.

[3] G. Renpu. *A filled function method for finding a global minimizer of a function of several variables*. Mathematical Programming, 46(1-3), 191-204, 1990.

[4] L. Zhang, C. Ng, D. Li, W. Tian. *A new filled function method for global optimization. Journal of Global optimization*, 28(1), 17-43, 2004.

[5] Y. Yang, Y. Shang. *A new filled function method for unconstrained global optimization*. Applied Mathematics and Computation, 173(1), 501-512, 2006.

[6] Y. Yao. *Dynamic tunneling algorithm for global optimization*. IEEE Transactions on Systems, Man, and Cybernetics, 19(5), 1222-1230, 1989.

[7] B. Cetin, J. Barhen, J. Burdick. *Terminal repeller unconstrained subenergy tunneling (TRUST) for fast global optimization*. Journal of Optimization Theory and Applications, 77(1), 97-126, 1993.

[8] J. Barhen, V. Protopopescu, D. Reister. *TRUST : A deterministic algorithm for global optimization*. Science, 276(5315), 1094-1097, 1997.