A finite-step convergent derivative-free method of unconstrained optimization

Yunqing Huang[a], Kai Jiang[a,*]

[a]*School of Mathematics and Computational Science,*
Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, P.R. China, 411105

Abstract

In our previous work [Y. Q. Huang and K. Jiang, Advances in Applied Mathematics and Mechanics, 2017, 9: 307-323], a useful derivative-free algorithm, the hill-climbing method with a stick (HiCS), has been proposed to treat unconstrained optimization. Numerical results have been demonstrated its wonderful performance. However, there are two issues required to be solved: convergent analysis and application to high dimensional problems. In this paper, we will give a rigorous theory to ensure finite-step convergence with mild conditions. Meanwhile, an economic sampling strategy using regular simplex of evaluating function values is proposed to treat high dimensional optimization. Finally, we use several benchmark numerical examples to demonstrate its efficiency.

Keywords: Hill-climbing method with a stick, Finite-step convergence, Suspected extreme point, Simplex, High-dimensional unconstrained optimization

1. Introduction

Derivative-free optimization is an area of long history and current rapid growth, fueled by a growing number of applications that range from science problems to medical problems to engineering design and facility location problems. In general, derivative-free optimization does not use derivative information to find optimal solution.

The derivative-free optimization algorithms can mainly be classified as direct and model-based. Direct algorithms usually determine search directions by evaluating the function f directly, whereas model-based algorithms construct and utilize a surrogate model of f to guide the search process. A detailed review about this kind of approaches was presented by Rios and Sahinidis [6]. Recently developed methods based trust-region using interpolation model belong to model-based methods [7, 8, 9, 10]. In practical implementation, heuristic algorithms, such as simulated annealing, genetic algorithm [11], have been also developed to solve derivative-free optimization. Here we focus our attention on the direct search algorithms.

In our previous work [1], we proposed a derivative-free optimization method, i.e., hill-climbing method with a stick (HiCS), to treat unconstrained optimization problems. The main idea of the

*kaijiang@xtu.edu.cn.

algorithm, at each iteration, is comparing function values on a surface surrounding the current point, rather than a neighbourhood of current node. It has many good properties, such as easily to implement, a unique parameter required to be modulated, and having capacity for find local and global maxima. However, it still lacks rigorous theoretical explanation. In this paper, we will give the convergence analysis and related properties of this algorithm. Meanwhile, a new strategy will be proposed to sample the search surface to deal with high dimension optimization problems.

In the following, we will briefly introduce the HiCS algorithm and prove its finite-step convergence in Sec. 2. The algorithm implementation is presented in Sec. 3. In particular, the new sampling strategy using regular simplex is also given in this section. The numerical experiments including high dimensional optimization problems are showcased in Sec. 4. Finally the conclusion and discussions are given in Sec. 5.

2. Algorithm description and convergence analysis

Before going further, a short introduction of the HiCS method is necessary. We consider an unconstrained optimization problem

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

where the objective function $f(x):\mathbb{R}^n\to\mathbb{R}$ is continuous. Let ρ be a search radius, $O(x_k,\rho)=\{x : |x-x_k|=\rho\}$ be the search surface in the k-th iteration with radius ρ. $U(x_k,\rho)$ is the neighbourhood of x_k with radius of ρ. To illustrate the algorithm to be more precision, an useful concept of suspected extreme point is introduced.

Definition 1. *For a given objective function $f(x)$ and a positive constant $\rho>0$, $\tilde{x}$ is a suspected extreme point if $f(\tilde{x})<f(x)$, for $\forall x\in O(\tilde{x},\rho)$. If $f(\tilde{x})<f(x)$ for all $x\in O(\tilde{x},\rho)$, $\tilde{x}$ is the suspected minimum point (SMP).*

Certainly, $\tilde{x}$ is a SMP if $\tilde{x}$ is a minimizer in the neighborhood of $U(\tilde{x},\rho)$. The opposite is not always true. The definition can be extended to describe suspected maximum point. With these notations, the HiCS algorithm can be presented more precisely as Algorithm 1.

Algorithm 1 Hill-Climbing method with a stick (HiCS)

1: **Initialization:** Choose x_0 and ρ.
2: **For** $k=0,1,2,\ldots$
3: Find $\bar{x}=\mathrm{argmin}_{y\in O(x_k,\,\rho)} f(y)$.
If $f(\bar{x})<f(x_k)$, then set $x_{k+1}=\bar{x}$.
Otherwise, declare that a SMP is found, and stop iteration.

It is evident that the approximation error of the HiCS algorithm is measured by the distance between a SMP and a minimum. When HiCS converges, its error is smaller than the search radius ρ. From our experience, the HiCS approach usually terminates in finite steps. It is an amazing property. In what follows, we will give the condition to ensure the finite-step convergence.

Theorem 1 (Finite-step convergence)**.** *Suppose that objective function $f(x)$ is continuous and the search domain Ω is a compact set. If there are not two SMPs x_* and x^* satisfying $|x_*-x^*|=\rho$ and $f(x^*)=f(x_*)=\alpha$. Then Algorithm 1 converges in finite steps.*

Proof. Assume that the HiCS method produces an infinite pair sequence $\{x_n, f(x_n)\}_{x=0}^{\infty}$. From assumption, it is obvious $f(x)$ is bounded. The decreasing sequence $\{f(x_n)\}_{n=0}^{\infty}$ converges, and the bounded $\{x_n\}_{n=0}^{\infty}$ has a convergent subsequence $\{x_{n_k}\}_{k=0}^{\infty}$. Assume that $f(x_n) \to \alpha$ and $x_{n_k} \to x^*$. Obviously x^* is a SMP.

In accordance with the subsequence $\{x_{n_k}\}_{k=0}^{\infty}$, we can always choose an another bounded subsequence $\{x_{n_k-1}\}_{k=0}^{\infty} \subset \{x_n\}$ satisfying $|x_{n_k-1} - x_{n_k}| = \rho$. Due to the boundedness of iteration sequence, $\{x_{n_k-1}\}_{k=0}^{\infty}$ has a convergent subsequence $\{x_{n_m}\}_{m=0}^{\infty}$. Let $x_{n_m} \to x_*$ when $m \to \infty$. x_* is also a SMP. From the $\{x_{n_m}\}$, we can find a subsequence $\{x_{n_m+1}\} \subset \{x_{n_k}\}$ satisfying $|x_{n_m} - x_{n_m+1}| = \rho$, and $x_{n_m+1} \to x^*$ $(m \to \infty)$. Obviously, $|x^* - x_*| = \rho$, and $f(x^*) = f(x_*) = \alpha$ which clearly contradicts the assumption. □

3. Algorithm implementation

As mentioned above, the HiCS algorithm can converge in finite steps with mild assumptions and has a unique parameter of search radius ρ to be chosen. In practice, the search surface $O(x_k, \rho)$ in each iteration shall be sampled in numerical implementation. The principle of discretization of sampling $O(x_k, \rho)$ includes symmetric and uniform distribution, and as few discretization points as possible when without a priori information of the objective function. Our previous work has demonstrated that uniformly distributed sampling points were useful to find the SMP when without a priori knowledge of objective functions [1]. However, the bisection sampling strategy based on spherical coordinates has been used in the previous work. The sampling points are as large as $2m^{n-1}$ in each iteration, m is the number of refinement, n is the dimensions of optimization problems. This significantly limits the application to high-dimensional problems. To overcome this limitation, it is required to develop a new strategy to sample $O(x_k, \rho)$ with a few sampling points. A reasonable requirement is that the number of sampling points will be linear or quasi-linear growth with the increases of problem dimensions. In this work, we will use the regular simplex and its rotations to sample search surface $O(x_k, \rho)$. As seen in the following, the computational complexity grows linearly as the dimension of optimization problems increases.

A n-dimension regular simplex is the congruent polytope of $\mathbb{R}^n$ with a set of points $\{a_1, \dots, a_n, a_{n+1}\}$, and all pairwise distances 1. Its Cartesian coordinates can be obtained from the following two properties:

1. For a regular simplex, the distances of its vertices $\{a_1, \dots, a_n, a_{n+1}\}$ to its center are equal.
2. The angle subtended by any two vertices of n-dimension simplex through its center is $\arccos(-1/n)$.

In particular, the above two properties can be implemented through Algorithm 2.

If the HiCS method has not find a better state on a regular simplex, we can add more points to refine $O(x_k, \rho)$ The new adding points of refining $O(x_k, \rho)$ should be distinct from existing samplings. Here we will refine $O(x_k, \rho)$ through rotating the regular simplex. For a given rotation

Algorithm 2 Generate n-D regular simplex coordinates

Give an $n \times (n+1)$-order zero matrix $x(1:n, 1:n+1)$
for $i = 1:1:n$ **do**
  $x(i,i) = \sqrt{1 - \sum_{k=1}^{i-1} [x(k,i)]^2}$
  for $j = i+1:1:n+1$ **do**
    $x(i,j) = \frac{1}{x(i,i)}\Big(-\frac{1}{n} - x(1:i-1,i)^T \cdot x(1:i-1,j)\Big)$
  end for
end for
Output the column vectors, and let $a_j = x(:,j),\ j = 1,2,\ldots,n+1$.

angle $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$, the rotation matrix $\boldsymbol{R}$ is given as

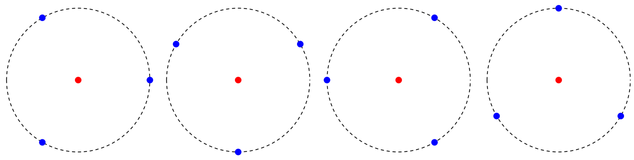
$$\boldsymbol{R} = \prod_{i=2}^{n-1} \begin{array}{c} \begin{array}{ccccccccc} & & & & i & & & & \end{array} \\ \begin{pmatrix} 1 & & & & \vdots & & & & \\ & \ddots & & & \vdots & & & & \\ & & 1 & & \vdots & & & & \\ & & & \cos\theta_i & 0 & -\sin\theta_i & & & \\ & & & 0 & 1 & 0 & & & \\ & & & \sin\theta_i & 0 & \cos\theta_i & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{pmatrix} \end{array} \tag{2}$$
$$\begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & & \\ \sin\theta_1 & \cos\theta_1 & 0 & & \\ 0 & 0 & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & 0 & 0 \\ & & 0 & \cos\theta_n & -\sin\theta_n \\ & & 0 & \sin\theta_n & \cos\theta_n \end{pmatrix}.$$

Then vertices of new simplex can be obtained by

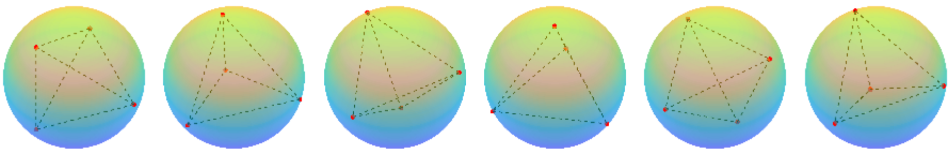
$$a_j = \boldsymbol{R} a_j + x_k \tag{3}$$

When without a priori knowledge of objective, the uniformly distributed principle is still a reasonable assumption to rotate regular simplex. A standard schematic plots of 2, and 3-D case are given in Fig. 1. It should be noted that there are also other strategies to rotate the regular simplex. For example, the additional simplexes can be dependent on the known information of objective functions.

To save computational amount, we choose a dynamic refinement strategy to sample the search surface and compare function values in practice. Based on the dynamic refinement strategy, we propose the method, see Algorithm 3. The computational amount is not larger than $m_{\max}(n+1)$ in each iteration which is linearly dependent on the dimension of optimization problems, $m_{\max}$ is



(a) 2D regular simplexes



(b) 3D regular simplexes

Figure 1: The first 2-, and 3-D regular simplexes of sampling the search set $O(x, \rho)$.

the maximum number of rotation. Whence, it is possible to treat high-dimensional optimization problems.

Algorithm 3 HiCS

Input x_0, ρ, and $m_{\max}$
for $k = 0, 1, 2, \cdots$ **do**
    Set $m = 0$
    if $m \leq m_{\max}$ **then**
        Discrete $O(x_k, \rho)$ to obtain $O_h^m(x_k, \rho)$
        if $\exists x_j \in O_h^m(x_k, \rho)$, s.t. $f(x_j) < f(x_k)$ **then**
            Set $x_{k+1} = x_j$, and $m = m_{\max} + 1$
        else
            Set $m = m + 1$
        end if
    else
        Declare that find a SMP, end program
    end if
end for

If the HiCS algorithm converges, the convergent result provides a good initial value for other optimization methods, including derivative-free approaches, and derivative-based algorithms if the objective function is differentiable. It is evident that the search space is shrunk to a ball with the radius ρ, and more significantly, the convergent ball contains a SMP. We will demonstrate this by several numerical experiments in Sec. 4.

We can also adjust the search radius ρ in HiCS method to improve the approximation precision as done in our previous work [1]. Algorithm 4 gives the process of narrowing down ρ when Algorithm 3 fails to find $f(\bar{x}) < f(x_k)$, $\bar{x} \in O(x_k, \rho)$ with a fixed ρ. The approximation distance between convergent point and a SMP is improved when Algorithm 4 converges. Certainly, the search surface can be expanded by setting control factor $\eta > 1$ if required. In fact, the Algorithm 4 can be restarted by fixed k iterations or by other criterions with different search radius ρ.

Algorithm 4 Adaptive HiCS: adjust ρ

Input $x_0, \rho, m_{\max}, \varepsilon$ and $\eta < 1$
if $\rho > \varepsilon$ **then**
  for $k = 0, 1, 2, \cdots$ **do**
    Set $m = 0$
    if $m \le m_{\max}$ **then**
      Discrete $O(x_k, \rho)$ to obtain $O_h^m(x_k, \rho)$
      if $\exists x_j \in O_h^m(x_k, \rho)$, s.t. $f(x_j) < f(x_k)$ **then**
        Set $x_{k+1} = x_j$, and $m = m_{\max} + 1$ (Jump out of IF statement)
      else
        Set $m = m + 1$
      end if
    else
      Set $\rho = \eta\rho$
    end if
    Set $k = k + 1$
  end for
end if

4. Numerical results

In this section, we choose three kinds of test functions, including a single extreme point function, high dimensional multi-extreme points functions, and a continuous but indifferentiable function, to demonstrate the performance of the HiCS algorithm. In Algorithm 3, the sampling points of search set in each iteration are $m(n+1)$, n is the dimension of objective function. If not specified, the maximum number of refinement $m = 32$.

4.1. The single extreme point problem: Gaussian function

The first example is a unimodal function, in particular, the Gaussian function

$$f(x) = -20 \exp\left(-\sum_{j=1}^{n} x_j^2\right), \tag{4}$$

who has a unique global minimum 0 with $f(0) = -20$. The objective function is differentiable in $\mathbb{R}^n$, however, it quickly diffuses out towards zero out of the upside-down “bell”.

We firstly investigate the convergent properties of HiCS method for 10-dimensional Gaussian function. The function satisfies the assumptions of Theorem 1, therefore, Algorithm 1 will be convergent in finite steps theoretically. To verify this fact, we use random initial values and carry out Algorithm 1 within 30 runs. In the set of numerical experiments, the search radius ρ is fixed as 0.3, start points are randomly generated in the space $[-1, 1]^{10}$. For each experiment, the HiCS method indeed converges and captures a neighbourhood of the peak 0 in finite iterations. Fig. 2 gives the required iterations for convergence in 30 numerical experiments. In these 30 runs, the average iterations of convergence is 20.5, while the maximum is 27, and the minimum is 9.

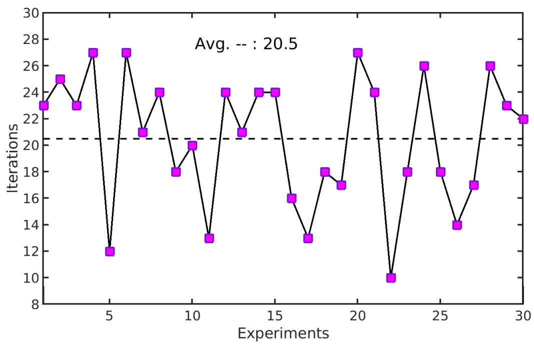


Figure 2: The iterations of convergence of the HiCS algorithm to the Gaussian function (4) in 30 runs. Start points are randomly generated in the space $[-1, 1]^{10}$, and $\rho = 0.3$. The flat dashed line shows the average.

Then we use a small search radius $\rho = 0.1$ to observe the behavior of HiCS method. The initial values are also randomly generated in 30 numerical tests. The required iterations for convergence is given in Fig. 3 In these 30 runs, the average iterations of convergence is 77.2, while the maximum is 121, and the minimum is 54. From these results, we can find that HiCS approach converges in finite iterations. Meanwhile, it is obvious that the value of ρ affects the number of iterations.

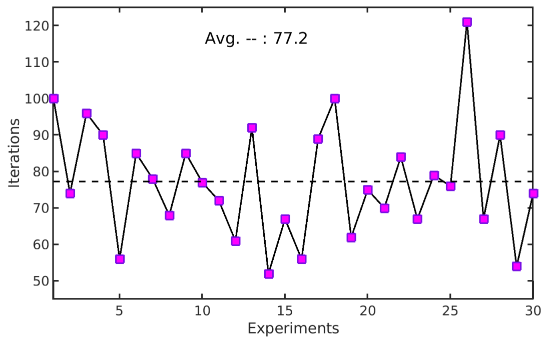
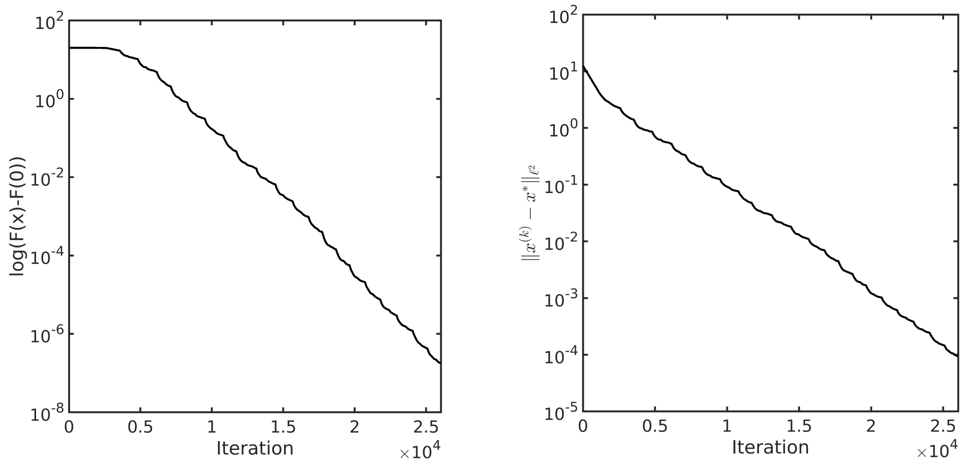


Figure 3: The iterations of convergence of the HiCS algorithm to the Gaussian function (4) in 30 runs. Start points are randomly generated in the space $[-1, 1]^{10}$, and $\rho = 0.1$. The flat dashed line shows the average.

In the following, we apply AHiCS algorithm to 1000 dimensional Gaussian function. The initial value is randomly generated in domain $[-1, 1]^{1000}$, the initial search radius $\rho_0 = 0.3$, and control factor $\eta = (\sqrt{5} - 1)/2$. Fig. 4 presents the iteration process. The difference between $f(x_k)$ and the global minimum $f(0) = -20$ is given in Fig. 4(a). The ℓ^2-distance between the iterator and the global minimizer 0, where $\|x\|_{\ell^2} = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$, is illustrated in Fig. 4(b). It can be found that the AHiCS method can approach to the global minimum through shrinking search radius ρ.



(a) (b)

Figure 4: The iteration process of the AHiCS method to 1000 dimensional Gaussian function. Start point is randomly generated in the space $[-1, 1]^{1000}$, $\rho = 0.3$ and control factor $\eta = (\sqrt{5} - 1)/2$.

4.2. The multi-minimizers problem: Ackley function

The second tested function is the Ackley function [19], a benchmark function, widely used for testing optimization algorithms. The expression of the Ackley function can be written as

$$f(x) = -20 \cdot \exp\left(-\frac{1}{5} \cdot \sqrt{\frac{1}{n}\sum_{i=1}^{2} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e, \tag{5}$$

where n is the dimension. Ackley function has many local minima and a unique global minimum of 0 with $f(0) = 0$, which poses a risk for optimization algorithms to be trapped into one of local minima, such as the traditional hill-climbing method [22]. In this subsection, we will apply the HiCS algorithm to higher dimensional Ackley function. In this subsection, the maximum rotation number $m_{\max} = 32$, the control factor $\eta = 0.5$.

We take 100 dimensional Ackley function as an example to test the performance of the HiCS algorithm. The first numerical test is taking $\rho = 2.0$ to observe the iteration process of the HiCS approach. The correspond iteration process is presented in Tab. 1 for 100 dimensional Ackley function. HiCS takes 365 iterations to obtain a SMP. In the first 353 steps, HiCS can efficiently find a better position merely using a regular simplex, i.e., within 101 function evaluations. After convergence, we can continue to apply the AHiCS method with $\eta = 0.5$. Fig. 5 gives the distance between function value and the function minimum, and the ℓ^2-distance between the position and the global minimizer. It is easy to find that the iterate indeed approximates the global minimizer as the iteration increases.

Then we try to test the ability of the proposed method for approximating different minimizer of the Ackley function. Since the unique parameter in the HiCS algorithm is the search radius ρ. We carry out the AHiCS method 30 times using different initial search radius ρ_0. The initial value is generated randomly in the region of $[-5, 5]^{100}$. The convergent criterion is the ℓ^2-distance between the convergent SMP and the global minimizer smaller than $\varepsilon = 10^{-14}$. Fig. 6 gives

Table 1: Iteration process of HiCS with constant $\rho_0 = 2.0$ to 100 dimensional Ackley function.

Iter.	ℓ^2-distance	Fun. Val.
1 (1-353)	43.76984 $\downarrow$ 5.67057	13.40276 $\downarrow$ 3.65003
3	5.76768	3.64961
5	5.84103	3.64579
5	5.75717	3.63706
1	5.72732	3.63154
1	5.68106	3.63021
6	5.67458	3.60655
12	5.76181	3.60569
5	5.84496	3.59674
1	5.76528	3.59377
1	5.71754	3.59277
2	5.90908	3.59054
32 (m_{max})	5.93767	3.58252

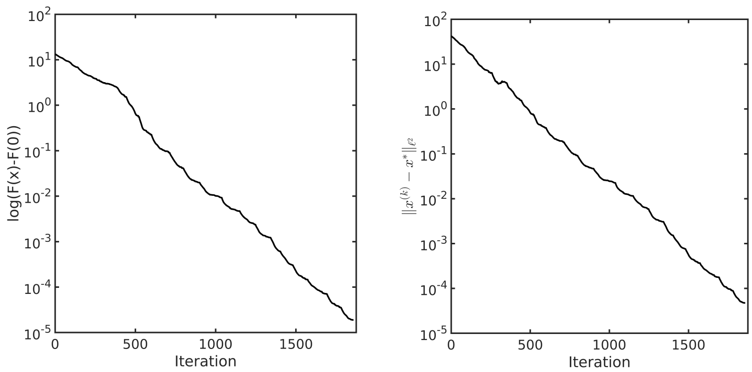
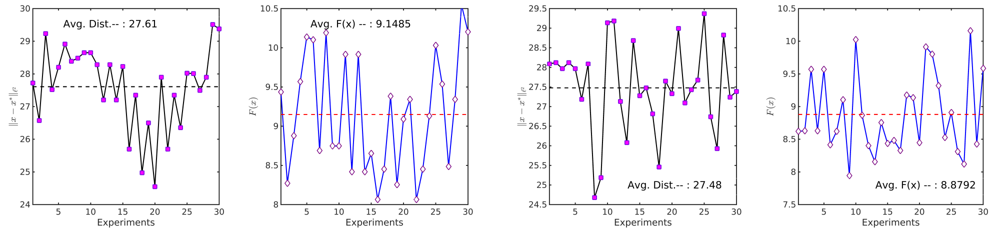
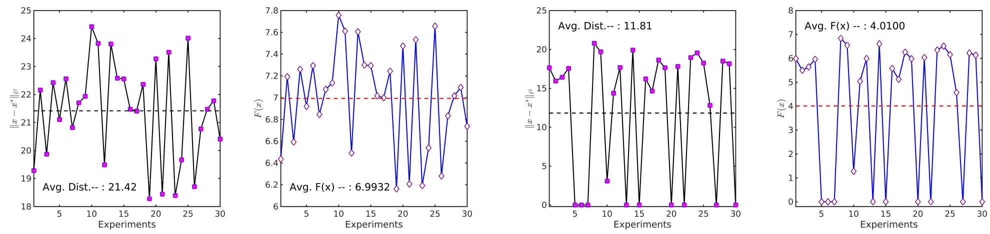


Figure 5: The iteration process of the AHiCS method to the 100 dimensional Ackley function when the HiCS algorithm is convergent with initial search radius $\rho = 2.0$ and the random initial value generated in $[-5, 5]^{100}$.

the ℓ^2-distance, and the function value after convergence when $\rho_0 = 0.05, 0.1, 0.5$, and 0.8, respectively. Obviously, with the increment of ρ_0, the convergent point is close to the global minimizer in the average sense. Correspondingly, the convergent function value becomes small. It can be found that when $\rho_0 = 0.8$, AHiCS method can find the neighbourhood with radius 10^{-14} of the global minimizer 0 for several times. It demonstrates that the AHiCS method can capture the global minimizer if ρ_0 is appropriately increased.



(a) $\rho_0 = 0.05$ (b) $\rho_0 = 0.1$



(c) $\rho_0 = 0.5$ (d) $\rho_0 = 0.8$

Figure 6: The iteration process of the AHiCS method to 100 dimensional Ackley function with different initial search radius ρ_0. Start point is randomly generated in the space $[-5, 5]^{100}$. The maximum rotation number $m_{\max} = 32$, the control factor $\eta = 0.5$ and the convergent criterion $\varepsilon = 10^{-14}$.

We continue to apply AHiCS method to 2500 dimensional Ackley function. The initial search radius is $\rho_0 = 3.5$, and initial position is generated randomly in $[-10, 10]^{2500}$. The control factor $\eta = 0.5$ to shrink the search radius. Due to such high dimension problem, the maximum number of rotation of simplexes is $m_{\max} = 16$ to save computational amount. The convergent criterion of AHiCS approach is that ρ is smaller than 10^{-5}. The iteration process is presented in Fig. 7. For the such high dimension optimization problem, the iteration behavior is similar to previous numerical experiments. When $\rho = 3.5$, the HiCS method costs 5415 steps to achieve convergence. We can further approximate the global minimizer 0 through shrinking ρ. As we can see from Fig 7, the AHiCS algorithm can arrive the global minimizer, as well as function value after 61755 iterations.

4.3. Other benchmark functions

Here we consider some benchmark functions in unconstrained optimization, including the Woods, ARWHEAD, CHROSEN functions [24, 25, 26].

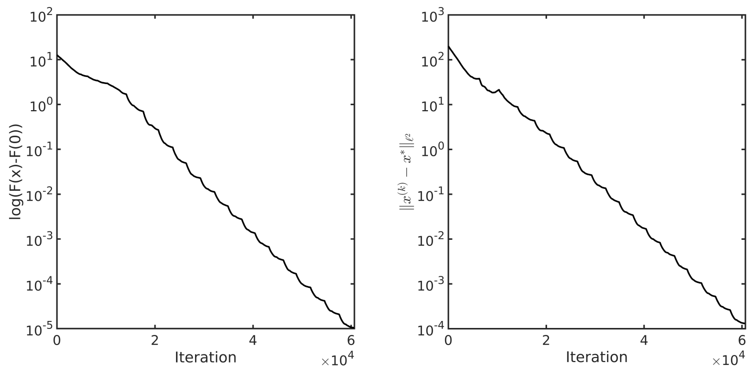


Figure 7: The iteration process of the AHiCS method to 2500 dimensional Ackley function with initial search radius $\rho_0 = 3.5$. Start point is randomly generated in the space $[-10, 10]^{2500}$. The maximum rotation number $m_{\max} = 32$, the control factor $\eta = 0.5$.

4.3.1. Woods function

The Woods function is a large and difficult problem in the CUTE test set [24]. Its specified expression is

$$F(x) = \sum_{i=1}^{n/4} \Big[100(x_{4i-2} - x_{4i-3}^2)^2 + (1 - x_{4i-3})^2 + 90(x_{4i} - x_{4i-1})^2 + (1 - x_{4i-1})^2 + 10(x_{4i-2} + x_{4i} - 2)^2 + 0.1(x_{4i-2} - x_{4i})^2 \Big]. \tag{6}$$

The global minimizer is $x^* = (1, 1, \ldots, 1)$ with $F(x^*) = 0$. Here we choose the hard initial value [24], $x_j^{(0)} = -3.0$ if j is even, and $x_j^{(0)} = -1.0$ if j odd, to test the HiCS method with $n = 320$ and $\rho = 5.0$. The HiCS approach spends 111 iterations to find a SMP. Then we can successively decrease the search radius by setting the control factor $\eta = 0.5$. Tab. 2 gives the iteration information when changing ρ four times, including the number of iterations, the ℓ^2-distance between convergent iterate and the global minimizer, and the convergent function value for each ρ. It can be found that the HiCS method can converge for each fixed search radius. Certainly, we can continue to apply the AHiCS approach with $\eta = 0.5$ to approximate the minimizer of Woods functions based on the above convergent results. The convergent procedure can be found in Fig. 8. It is easy to find that the AHiCS can further approximate the global

Table 2: Iteration information of applying the HiCS method to 320 dimensional Woods function through successively decreasing ρ.

ρ	Iter.	ℓ^2-distance	$F(x)$
5.0	111	2.0269797302e+01	1.9462281448e+04
2.5	21	2.1446697698e+01	1.7213410433e+04
1.25	38	2.3007762412e+01	1.4027762445e+04
0.625	49	2.1442322133e+01	9.2058823364e+03

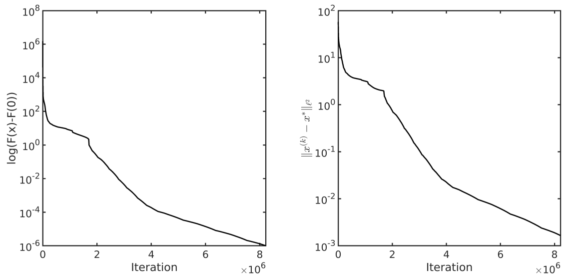


Figure 8: The iteration process of the AHiCS method ($\eta = 0.5$) to the 320 dimensional Woods function.

minimizer as the iterations increase and the search radius decreases.

The others benchmark examples, including ARWHEAD and CHROSEN functions, have been also used by Powell in 2006 to test his derivative-free method [25]. The ARWHEAD function is

$$F(x) = \sum_{i=1}^{n-1} [(x_i^2 + x_n^2)^2 - 4x_i + 3]. \tag{7}$$

The least value of F is zero, which occurs when the variables take the values $x_j = 1$, $j = 1, 2, \ldots, n-1$ and $x_n = 0$. We directly apply AHiCS method ($\eta = 0.5$) to 640 dimensional ARWHEAD function. The starting vector is given by $x_j^{(0)} = 1$, $j = 1, 2, \ldots, n$, as Powell done in Ref. [25] The initial search radius $\rho_0 = 3$ and the maximum number of refinement $m_{\max} = 32$ Fig. 9 gives the iteration process of applying the AHiCS algorithm to 640 dimension ARWHEAD

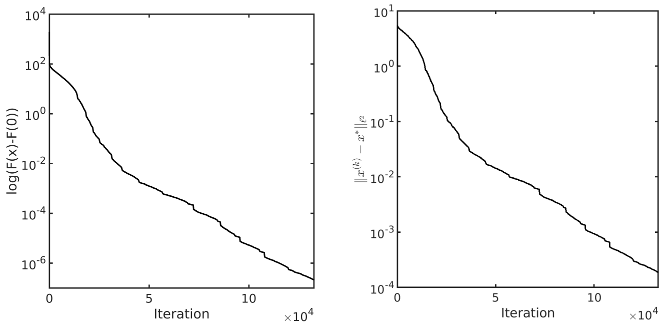


Figure 9: The iteration process of the AHiCS method ($\eta = 0.5$) to the 640 dimensional ARWHEAD function.

function.

The expression of CHROSEN function is

$$F(x) = \sum_{i=1}^{n-1} [(4(x_i - x_{i+1}^2)^2 + (1 - x_{i+1})^2]. \tag{8}$$

Its least value of F is zero, which occurs when the variables take the values $x_j = 1$, $j = 1, 2, \ldots, n$. We consider the 100 dimensional CHROSEN function with start point $x_j^0 = -1$, $j = 1, 2, \ldots, n$, as Powell done [25]. We also apply the AHiCS method to this problem with initial search radius $\rho_0 = 5$, $m_{\max} = 32$ and $\eta = 0.5$. The iteration process is presented in Fig. 10.

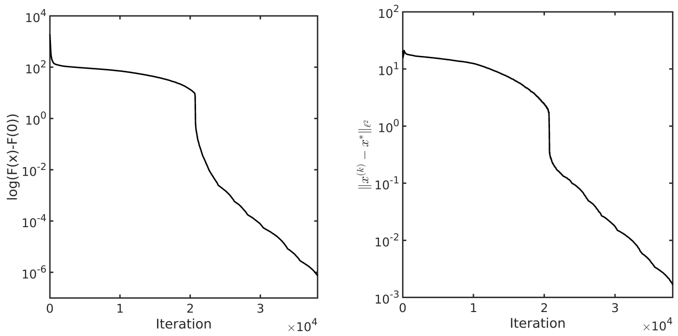


Figure 10: The iteration process of the AHiCS method ($\eta = 0.5$) to the 100 dimensional CHROSEN function.

From Fig. 9 and Fig. 10, one can find that the iteration processes are different for both ARWHEAD and CHROSEN functions which is attributed to the properties of objective functions. However, the AHiCS method is able to efficiently approximate the global minimizer for both functions.

5. Discussion

Inspired by the hill-climbing behavior of the blind, we has proposed a new derivative-free method to unconstrained optimization problems in our previous work [1]. In this paper, we establish a rigorous mathematical theory of HiCS algorithm which theoretically ensures finite-step convergence under mild conditions. Numerical results also have demonstrated this great property. In practice, the computational complexity of HiCS algorithm mainly depends on the sampling strategy on search boundary. In our previous work, the number of sampling points increases exponentially with the dimension of problems. It limits the application to the high-dimensional optimization. In this work, to deal with high-dimensional problems, we proposed a new strategy of simplex sampling method to save computational amount. Using the new sampling strategy, the number of function valuations is linear dependent on the dimension of problems. It allows us to solve high dimensional optimization problems. Finally we demonstrate the efficiency of our proposed algorithm through solving several benchmark problems.

Acknowledgments

[1] Huang, Yunqing and Jiang, Kai, Hill-Climbing Algorithm with a Stick for Unconstrained Optimization Problems, Advances in Applied Mathematics and Mechanics, 2017, 9: 307–323.

[2] W. Y. Sun and Y. Yuan, Optimization theory and methods: nonlinear programming, New York: Springer, 2006.

[3] A. R. Conn, N. I. M. Gould and P. L. Toint, Trust region methods, Philadelphia: SIAM, 2000.

[4] J. Nocedal and S. J. Wright, Numerical optimization, Berlin: Springer-Verlag, 2nd ed., 2006.

[5] A. R. Conn, K. Scheinberg and L. N. Vicente, Introduction to derivative-free optimization, Philadelphia: SIAM, 2009.

[6] L. M. Rios and N. V. Sahinidis, Derivative-free optimization: a review of algorithms and comparison of software implementations. J. Global Optim., 2013, 56: 1247–1293.

[7] M. J. D. Powell, UOBYQA: unconstrained optimization by quadratic approximation, Technical Report DAMTP NA2000/14, CMS, University of Cambridge, 2000.
[8] M. J. D. Powell, On trust region methods for unconstrained minimization without derivatives, Technical Report DAMTP NA2002/NA02, CMS, University of Cambridge, February 2002.
[9] T. Wu, Y. Yang, L. Sun, and H. Shao, A heuristic iterated-subspace minimization method with pattern search for unconstrained optimization, Comput. Math. Appl., 2009, 58: 2051-2059.
[10] Z. Zhang, Sobolev seminorm of quadratic functions with applications to derivative-free optimization, Math. Program., 2014, 146: 77-96.
[11] Z. Michalewicz and D. B. Fogel, How to solve it: modern heuristics, Springer, 2004.
[12] Y. LeCun, Bengio, Y. Hinton, G. Hinton, Deep learning, Nature, 2015, 521: 521-436.
[13] R. Hooke and T. A. Jeeves, "Direct search" solution of numerical and statistical problems, J. ACM, 1961, 8: 212–229.
[14] R. M. Lewis, V. Torczon and M. W. Trosset, Direct search methods: then and now, J. Comput. Appl. Math., 2000, 124: 191–207.
[15] J. A. Nelder and R. Mead, A simplex method for function minimization, Comput. J., 1965, 7: 308–313.
[16] V. Torczon, On the convergence of pattern search algorithms, SIAM J. Optim., 1997, 7: 1–25.
[17] T. G. Kolda, R. W. Lewis and V. Torczon, Optimization by direct search: new perspectives on some classical and modern methods, SIAM Rev., 2003, 45: 385–482.
[18] J. E. Jr Dennis and V. Torczon, Direct search methods on parallel machines, SIAM J. Optim., 1991, 1: 448–474.
[19] J. M. Dieterich and B. Hartke, Empirical review of standard benchmark functions using evolutionary global optimization, Appl. Math. 2012, 3: 1552–1564.
[20] S. Gratton, C. W. Royer, L. N. Vicente, Z. Zhang, Direct search based on probabilistic descent, SIAM J. Optim., 2015, 25:1515-1541.
[21] S. J. Russell and P. Norvig, Artificial intelligence: a modern approach, 3rd ed., Prentice Hall, 2010.
[22] T. Bäck, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Oxford University Press, 1996.
[23] J. E. Jr Dennis and D. J. Woods, Optimization on microcomputers: The Nelder-Mead simplex algorithm. In: New computing environments: microcomputers in large-scale computing, A. Wouk ed., Philadelphia: SIAM, 1987.
[24] L. Lukšan, C. Matonoha, J. Vlcek, Modified CUTE problems for sparse unconstrained optimization, Techical Report, 2010, 1081.
[25] M. J. D. Powell, The NEWUOA software for unconstrained optimization without derivative, in Large-Scale Nonlinear Optimization , eds. G. Di Pillo and M. Roma, Springer (New York), 2006, 255 297.
[26] N. Andrei, An unconstrained optimization test functions collection, Adv. Model. Optim., 2008, 10: 147–161.