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

Inspired by the behavior of the blind for hill-climbing using a stick to detect a higher place by drawing a circle, in this talk, we will present a new derivative-free method, i.e., the hill-climbing method with a stick (HiCS), to treat unconstrained optimization. At a given point, the new algorithm can obtain a better state by searching a surface with the length of the stick. This algorithm can capture a neighbourhood of a minimizer of the objective function rather than directly approximating it. A simple but rigorous theory can guarantee the finite-step convergence of the proposed algorithm without convexity assumption. Only one parameter is required to be input in this method which makes it easy for coding. Meanwhile, an economic sampling strategy with the regular simplex of evaluating function values is given to optimize high dimensional problems. Finally, several standard numerical examples have been used to demonstrate its efficiency. HiCS shows potential to find the global minimizer by choosing proper algorithm parameters.

Keywords: Stick hill-climbing algorithm, Finite-step convergence, Suspected extreme point, Simplex sampling, 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.

*kaijiang@xtu.edu.cn.

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 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 $f(x):\mathbb{R}^n\rightarrow\mathbb{R}$ 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 $f(\tilde{x})<f(x)$ or $f(\tilde{x})>f(x)$, 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: Try to find $\bar{x}\in O(x_k,\rho)$, s.t. $f(\bar{x})<f(x_k)$.
If such a point is found, then set $x_{k+1}=\bar{x}$.
Otherwise, a SMP is found, and declare the iteration successful.

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)**.** *Assume 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 $x_{n_k} \to x^*$.

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 $m \to \infty$. 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. $\square$

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**
    $x(i,j) = \dfrac{1}{x(i,i)}\Big(-\dfrac{1}{n} - x(1:i-1,i)^T \cdot x(1:i-1,j)\Big)$
  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

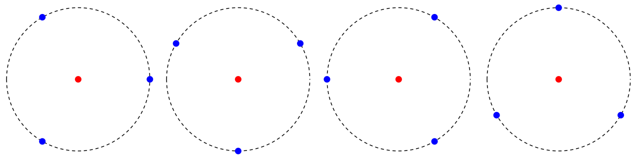
$$\boldsymbol{R} = \prod_{i=2}^{n-1} \begin{array}{c} \begin{array}{ccccccccc} & & & & i & & & & \end{array} \\ \begin{pmatrix} 1 & & & & \vdots & & & & \\ & \ddots & & & \vdots & & & & \\ & & 1 & & \vdots & & & & \\ & & & \cos\theta_i & 0 & -\sin\theta_i & & & \\ & & & 0 & 1 & 0 & & & \\ & & & \sin\theta_i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

Then vertices of new simplex can be obtained by

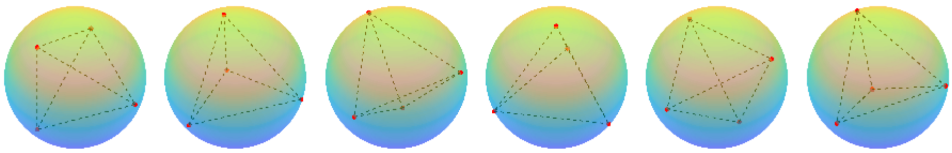
$$a_j = \boldsymbol{R} a_j + x_k \tag{3}$$

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 computable HiCS method, see Algorithm 3. The computational amount is not



(a) 2D regular simplexes



(b) 3D regular simplexes

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

larger than $m_{\max}(n+1)$ in each iteration which is linearly dependent on the dimension of optimization problems, $m_{\max}$ is 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 $m \le m_{\max}$ **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 $m = m_{\max} + 1$
        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 ρ can be expanded by setting control factor $\eta > 1$ if required. In fact, Algorithm 4 provides a restart technique by fixed the k-iterate with different search radius ρ.

Algorithm 4 HiCS: adjust ρ

Input x_0, ρ, $m_{\max}$, ε and $\eta < 1$
if $\rho > \varepsilon$ **then**
  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 $m = m_{\max} + 1$ (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. A single extreme point problem

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.

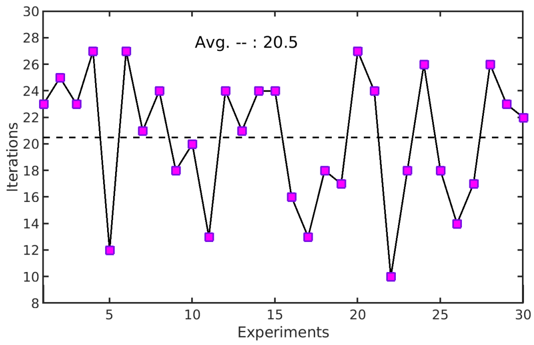


Figure 2: The iterations of convergence of the HiCS algorithm to the Gaussian function (4) in 30 runs. Start points are randomly generated in the space $[-1, 1]^{10}$, and $\rho = 0.3$. The flat dashed line shows the average.

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.

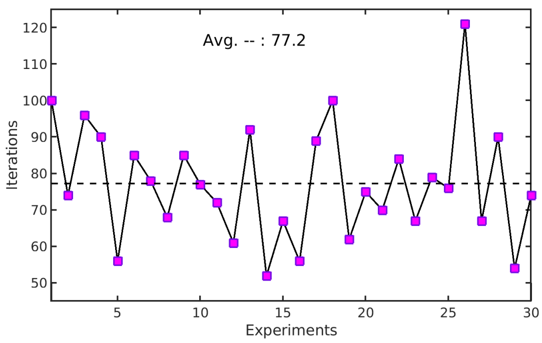
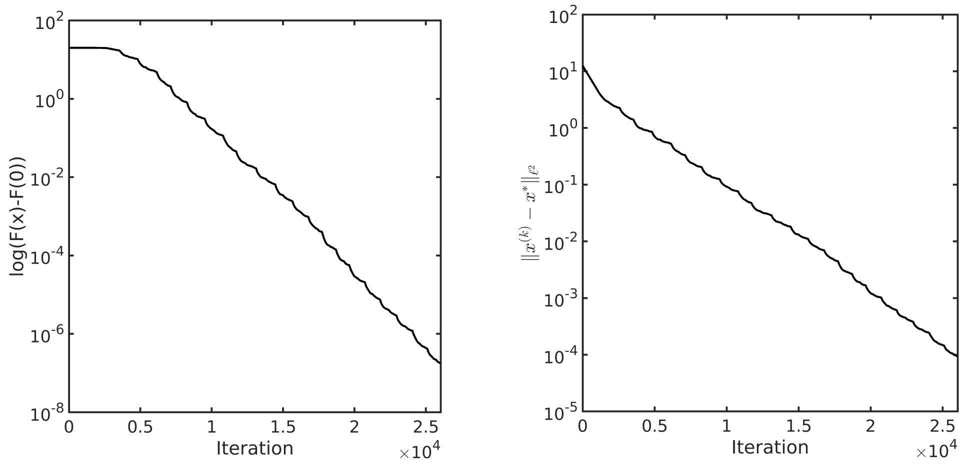


Figure 3: The iterations of convergence of the HiCS algorithm to the Gaussian function (4) in 30 runs. Start points are randomly generated in the space $[-1, 1]^{10}$, and $\rho = 0.1$. The flat dashed line shows the average.

In the following, we apply HiCSa 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 HiCSa method can approach to the global minimum through shrinking search radius ρ.



(a) (b)

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

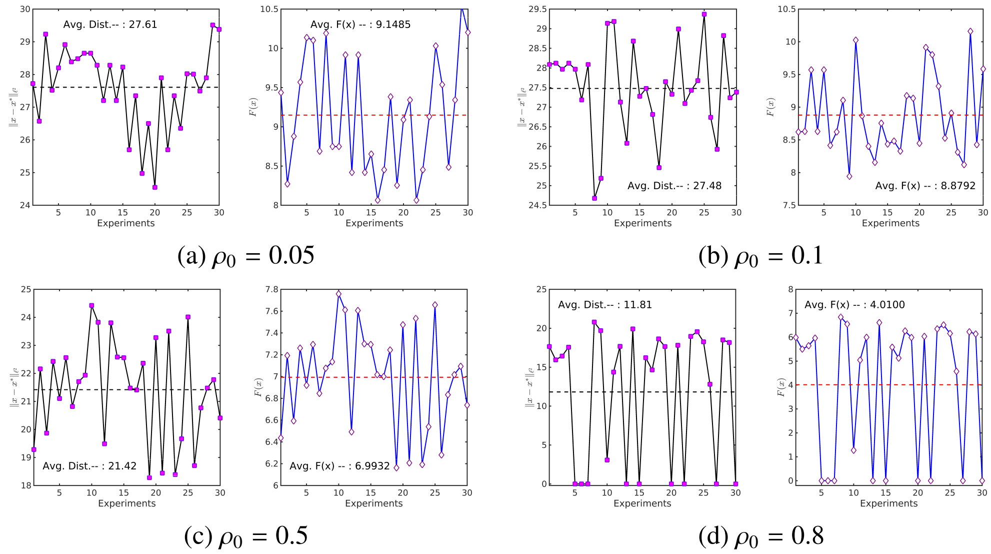
4.2. Ackley function: multi-minimizers problems

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 our proposed algorithms to higher dimensional Ackley function. In this subsection, the maximum rotation number $m_{\max} = 32$, the control factor $\eta = 0.5$.

Firstly, taking 100 dimensional Ackley function as an example, we will test the ability of HiCSa to find the different minimizer. Since the main parameter in our method is the search radius, we implement HiCSa method 30 times for different initial search radius ρ_0. The initial value is generated randomly in the region of $[-5, 5]^{100}$. The convergent criterion $\varepsilon = 10^{-14}$. Fig. 5 gives the ℓ^2-distance between the convergent SMP and the global minimizer, and the function value after convergence when $\rho_0 = 0.05, 0.1, 0.5, 0.8$, 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 $\rho_0 = 0.8$, HiCSa method can find the neighbourhood with radius 10^{-14} of the global minimizer 0 for several times. We also find that the HiCSa can capture the global minimizer when further increasing ρ_0.



(a) $\rho_0 = 0.05$

(b) $\rho_0 = 0.1$

(c) $\rho_0 = 0.5$

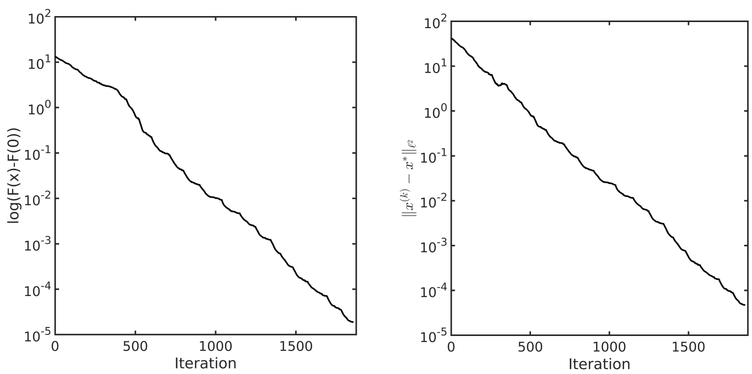
(d) $\rho_0 = 0.8$

Figure 5: The iteration process of the HiCSa 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 $\varepsilon = 10^{-14}$.

Then we use $\rho_0 = 2.0$ to observe the iteration process of HiCS. Tab. 1 gives the iteration process of HiCS with constant $\rho_0 = 2.0$ to 100 dimensional Ackley function.

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)	43.769843 ↓ 5.670566	13.402764 ↓ 3.650028
3	5.767675	3.649609
5	5.841026	3.645791
5	5.757173	3.637059
1	5.727323	3.631540
1	5.681064	3.630205
6	5.674577	3.606546
12	5.761805	3.605687
5	5.844962	3.596742
1	5.765280	3.593773
1	5.717544	3.592769
2	5.909084	3.590538
32 (m_{max})	5.937673	3.5825200



We continue to apply HiCSa method to 2500 dimensional Ackley function. The initial search radius is $\rho_0 = 3.5$, and initial position is generated randomly in $[-5, 5]^{2500}$. The control factor $\eta = 0.5$ to shrink the search radius. The iteration process is presented in Fig. 6. As we can see from this figure, the iterator approximates the global minimizer, and the function value decreases to 0.

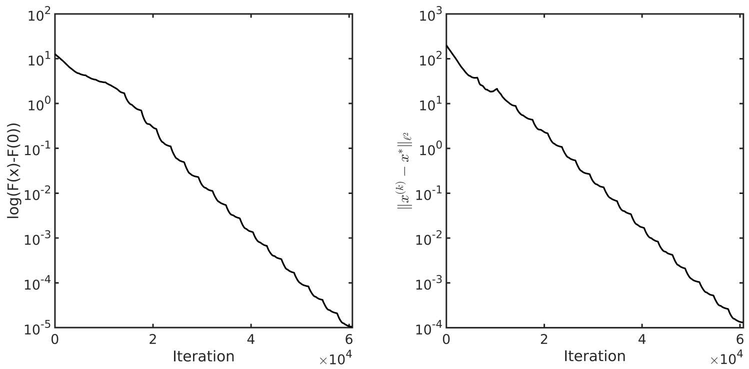


Figure 6: The iteration process of the HiCSa method to 2500 dimensional Ackley function with initial search radius $\rho_0 = 3.5$. Start point is randomly generated in the space $[-5, 5]^{2500}$. The maximum rotation number $m_{\max} = 32$, the control factor $\eta = 0.5$.

Firstly we consider the 2 dimension Ackley function whose morphology can be found everywhere, such as [1]. In our previous work, we have found that the HiCS method is able to obtain a neighbourhood of the global minimum with an appropriate search radius ρ using the bisection sampling approach. In the current work, we still to examine the convergent behavior of

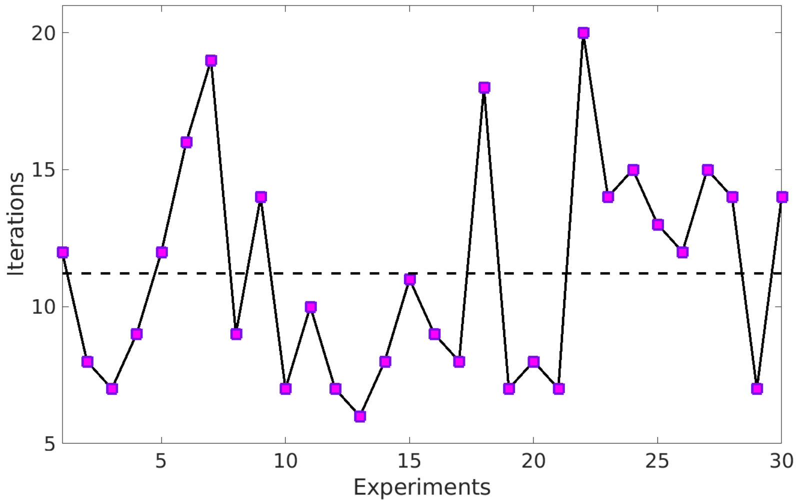


Figure 7: The required iterations of convergence by HiCS algorithm 1 for 2 dimension Ackley function (5) in 30 numerical experiments. Start points are randomly generated in the space $[-10, 10]^2$ with fixed $\rho = 1.0$. The flat dashed line shows the average.

Algorithm 1 to approximate the global minimum using the simplex sampling strategy. The first 30 numerical experiments are performed with random initial values generated in the space of $[-10, 10]^2$ when $\rho = 1.0$. The final convergent domain in each test contains the global minimizer $(0, 0)$. Fig. 7 shows the required iterations of convergence. In the 30 numerical tests, the average iterations of convergence is about 11, while the maximum number of iterations is 20 and the minimum is 4. The initial positions also only affect the speed of convergence, but not the ability of catching the neighbourhood of the global minimum.

The above tests have demonstrated that the HiCS algorithm can capture the neighbourhood of the global minimum of the Ackley function. Subsequently we plan to manifest the ability of the HiCS algorithm to obtain local minimizers. As discussed above, the unique regulatable parameter in the HiCS scheme is the search radius ρ. Therefore we will test the numerical behaviors with different size of ρ. In the set of tests, the initial value is always $x_0 = (4.1, 3.4)$. Tab. 2 gives the convergent results with different ρ. Fig. 8 marks corresponding local minima.

From these numerical experiments, one can find that the search radius ρ plays a filter role in catching different local minimum by setting different values. When ρ is greater than 0.6, the HiCS approach can approximate the global minimum, otherwise, the method can converge to different local minima. If starting from other initial position, the HiCS method can find other minima.

Next we consider the 100 dimension Ackley function. The initial value is randomly generated in $[-10, 10]^{100}$ and the search radius is set as $\rho = 2$. The iteration detail has been listed in Tab. 3. The first and second columns show the number of iterations and rotation simplexes when

Table 2: The convergent results and required iterations of the HiCS algorithm with different ρ when the initial value is $x_0 = (4.1, 3.4)$. The locations of different local minima are marked in Fig. 8.

ρ	0.1	0.3	0.5	0.55	0.58	0.6	1.0
Iterations	7	3	2	6	7	15	7
Min.	Loc_1	Loc_1	Loc_1	Loc_2	Loc_3	Global	Global

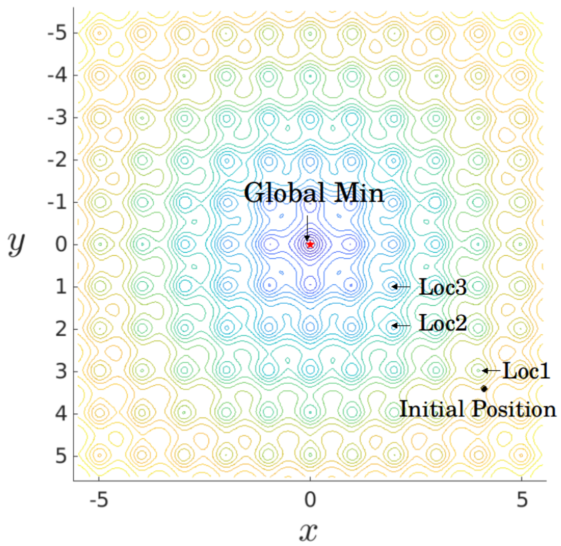


Figure 8: The locations of some local minima and the global minimum of 2 dimension Ackley function.

applying HiCS method. The third column gives the ℓ^2-distance between the current iterator and the global minimizer 0. The fourth column is the function value on the current iterator. Obviously, HiCS algorithm converges within 365 steps and the global minimizer is contained in the convergent neighbourhood.

Table 3: The iterative procedure of HiCS algorithm with fixed $\rho = 2$ and random initial value when optimizing 100 dimension Ackley function.

Iteration	m	ℓ^2-distance	Function value
1 ↓ 353	1	43.769842839 ↓ 5.6705662388	13.402763950 ↓ 3.6500283195
354	3	5.7676753732	3.6496090883
355	5	5.8410256060	3.6457912911
356	5	5.7571725375	3.6370586909
357	1	5.7273233589	3.6315399254
358	1	5.6810640400	3.6302051882
359	6	5.6745774447	3.6065456752
360	12	5.7618050286	3.6056866282
361	5	5.8449616655	3.5967423037
362	1	5.7652804943	3.5937731951
363	1	5.7175444967	3.5927685452
364	2	5.9090844843	3.5905379762
365	32 ($m_{\max}$)	5.9376731371	3.5825199326

The convergent result by HiCS method efficiently shrinks the search region. Then we can further approximate the unique minimizer by repeatedly using HiCS with different ρ, i.e., Algorithm 4. The convergent criterion of HiCSa is $\rho < 10^{-5}$. The detail of iteration procedure is given in Tab. 4. In this table, the first and second columns give the values of search radius ρ and corresponding iterations. The third and fourth columns are the same as the third and fourth ones in Tab. 3. Obviously, for each fixed ρ, the algorithm is convergent in finite steps. The pair sequence produced by HiCSa method approximates to both the unique global minimizer and corresponding function value.

Finally, we apply HiCS and HiCSa algorithms to 2500 dimension Ackley function. The random initial value is randomly generated in $[-10, 10]^{2500}$, and initial search radius is $\rho = 3.5$. 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 HiCSa approach is that ρ is smaller than 10^{-5}. The iterative information can be found in Tab. 5. The table is the same as Tab. 4 except that 2500 dimension Ackley function is optimized. For the high dimension optimization problem, the iteration behavior is similar to previous numerical experiments. When $\rho = 3.5$, the scheme, which is actual the HiCS method, converges within 5415 steps. Keeping carrying out the HiCS algorithm by shrinking ρ, we can further approximate the global minimizer 0. After 61755 iterations, the HiCSa algorithm achieves the convergent point as shown in the last row in Tab. 5. It costs roughly 3.7 hours of real time using one Inter 3.60 GHz i7-4790 processor.

Table 4: The iterative procedure of HiCSa algorithm for optimizing 100 dimension Ackley function based on the convergent result as given in Tab. 3. The control factor $\eta = 0.5$.

ρ	Iterations	ℓ^2-distance	Function value
2.0	431	5.9376731371	3.5825199326
1.0	152	2.5717428366	2.2979795353
0.5	171	1.2147220195	1.1029637432
0.25	145	0.60161985055	0.42190311564
0.125	152	0.30400991430	0.17007648336
0.0625	105	0.14644406013	0.069957199791
0.03125	175	0.076076391231	0.033509781782
↓	↓	↓	↓
1.525879e-05	135	3.5663302607e-05	1.4265998399e-05

Table 5: The iterative procedure of optimizing 2500 dimension Ackley function using HiCSa algorithm with initial $\rho = 3.5$ and random initial value. The control factor $\eta = 0.5$.

ρ	Iteration	ℓ^2-distance	Function value
3.5	5415	188.54368262 ↓ 43.980512458	12.310134936 ↓ 4.7600200861
1.75	5064	↓ 24.909250731	↓ 3.3115891477
0.875	4099	↓ 10.893125123	↓ 2.0271413426
0.4375	4869	↓ 5.1289809138	↓ 0.88082043719
0.21875	3269	↓ 2.5698591962	↓ 0.34009221628
↓	↓	↓	↓
1.335144e-05	3340	1.5546421188e-04	1.2437651812e-05

4.3. Sphere function

$$f(x) = \sum_{i=1}^{n} x_i^2 \tag{6}$$

The Sphere function has n local minima except for the global one $x = (0, 0, \ldots, 0)$ with $f = 0$. It is continuous, convex and unimodal.

4.4. Powell function

$$F(x) = \sum_{i=1}^{n/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4] \tag{7}$$

The function is usually evaluated on the hypercube $x_i \in [-4, 5]$ for all $i = 1, \ldots, n$.
available

4.5. ARWHEAD function

$$F(x) = \sum_{i=1}^{n-1} [(x_i^2 + x_n^2)^2 - 4x_i + 3] \tag{8}$$

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 $x_n = 0$. The starting vector is given by $x_j^{(0)} = 1$, $j = 1, 2, \ldots, n$, as Powell done in Ref. [26].
available

4.6. CHROSEN function

$$F(x) = \sum_{i=1}^{n-1} [(4(x_i - x_{i+1}^2)^2 + (1 - x_{i+1})^2] \tag{9}$$

The least value of F is zero, which occurs when the variables take the values $x_j = 1$, $j = 1, 2, \ldots, n$. The starting vector is given by $x_j^{(0)} = -1$, $j = 1, 2, \ldots, n$, as Powell done in Ref. [26].
Hard

Table 6: : Iteration information

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
0.3125	558	1.8950544937e+01	2.3646230729e+03
0.15625	634	1.6684359010e+01	1.2099516621e+03
0.078125	2502	1.2212940331e+01	2.9382990538e+02
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
1.907349e-05	114180	4.4027298178e-02	7.0921802110e-04

It costs 48665367 function evaluations.

4.7. *Woods function [24, 25]*

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

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

The global minimizer is $(1, 1, \dots, 1)$ with $f = 0$. Here we choose the hard initial value [], i.e., $x_j^{(0)} = -3.0$ if j is even, and $x_j^{(0)} = -1.0$ if j odd, to test our proposed method HiCS with n variables. The choices of n are 4, 20, 80, 320, 1280, 2000.

We can further apply HiCSa approach ($\eta = 0.5$) to approximating the minimizer of Woods functions based on the above convergent results. Here we take $n = 80$ as an example to demonstrate the iteration procedure.

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 built 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 which determines the function valuations. In our previous work, the number of sampling points increases exponentially with the dimension of problems. It limits the application to high-dimensional optimization. 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. Taking Ackley function as an example, it allows us to solve up to 2500 dimension function within a few hours.

Table 7: : Iteration information

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
0.3125	558	1.8950544937e+01	2.3646230729e+03
0.15625	634	1.6684359010e+01	1.2099516621e+03
0.078125	2502	1.2212940331e+01	2.9382990538e+02
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
1.907349e-05	114180	4.4027298178e-02	7.0921802110e-04

It costs 48665367 function evaluations.

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] https://pythonhosted.org/algopy/index.html
[26] 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.
[27] N. Andrei, An unconstrained optimization test functions collection, Adv. Model. Optim., 2008, 10: 147–161.