CSIAM Trans. Appl. Math.
doi:

Vol. **x**, No. 1, pp. 1-13
February 202x

A finite-step convergent method of finding a suspected extreme point

Yunqing Huang[1]* and Kai Jiang[1]

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

Abstract. In our previous work [Adv. Appl. Math. Mech., 2017, 9: 307-323], we proposed a novel optimization algorithm, the hill-climbing method with a stick (HiCS), to address the unconstrained optimization. HiCS can find a suspected extreme point rather than the minimizer which indicates the minimizers exist in a small domain with mild conditions. In this paper, we give a rigorous theory to guarantee that the HiCS has a finite-step convergence property. Meanwhile, an efficient discretization strategy based on the regular simplex is developed to treat high-dimensional problems. The power of the improved HiCS is demonstrated by several high-dimensional benchmarks.

AMS subject classifications: 90C56, 90C59, 65K05.

Key words: Hill-climbing method with a stick, Suspected extreme point, Finite-step convergence, Simplex, High-dimensional problems

1 Introduction

Classic optimization approaches are developed to directly find the extreme point which can be divided into two classes, directional search and model-based [1–3]. Directional search algorithms first determine the search direction and then the step length along with the search direction. Model-based approaches construct and utilize a related simple model to approximate the original problem in a trust region to guide the search process. Before finding the extreme point, an important thing is determining that there exist minimizers in the search domain. Inspired by the behavior of the blind for climbing hill, we proposed a new approach, the HiCS, to greatly shrink the search domain which includes minimizers [4]. The main idea of the HiCS, at each search step, is comparing function values on a surface surrounding the current iterator. It requires a comparison of function values, and does not need the search direction or construct a surrogate model in a

*Corresponding author. *Email addresses:* huangyq@xtu.edu.cn

trust region. The HiCS is different from the direct search methods in derivative-free optimization. The direct search methods choose a set of nonzero vectors deterministically or stochastically as search directions, then try to find a replaceable point along search directions with a step size until a certain stopping criterion is met. The HiCS can find a neighbourhood that contains extreme value points rather than directly finding them. Numerical results have been demonstrated that the HiCS has many satisfactory properties, including being easy to implement, a unique parameter to be modulated, and having the capacity of finding a small domain of the local and global minimizer. However, there are two unsolved problems in the previous work, including a rigorous theoretical explanation and the treatment of high-dimensional problems. This paper will give the convergence analysis and related properties of this algorithm by introducing a new concept, the suspected extreme point. Meanwhile, a new strategy will be proposed to discretize the search surface for high-dimensional objective functions.

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

2 HiCS and convergence analysis

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

$$\min_{x\in\Omega\subset\mathbb{R}^d} f(x), \tag{2.1}$$

where the objective function $f(x):\mathbb{R}^d\to\mathbb{R}$. Let ρ be the search radius, $O(x_k,\rho)=\{x:\|x-x_k\|=\rho\}$ be the search surface at the k-th iteration with radius ρ. $\|\cdot\|$ is the common norm in $\mathbb{R}^d$ space. Denote $U(x_k,\rho)$ is the neighbourhood of x_k with radius of ρ. $\overline{U}(\tilde{x},\rho)$ is the closure of $U(\tilde{x},\rho)$, i.e., $\overline{U}(\tilde{x},\rho)=U(\tilde{x},\rho)\cup O(\tilde{x},\rho)$. To illustrate the algorithm more accurately, an useful concept of the *suspected extreme point* is proposed.

Definition 2.1. For a given objective function $f(x)$ and a positive constant $\rho>0$, $\tilde{x}$ is a *suspected extreme point* (SEP) if $f(\tilde{x})\le f(x)$ or $f(\tilde{x})\ge f(x)$, for each $x\in O(\tilde{x},\rho)$. If $f(\tilde{x})\le f(x)$ for all $x\in O(\tilde{x},\rho)$, $\tilde{x}$ is the *suspected minimum point (SMP)*.

Remark 2.1. Note that the SMP might be not a minimizer. If the SMP is found which means that the minimizer point is close to it. If $f(x)$ is continuous in $\overline{U}(\tilde{x},\rho)$, $\tilde{x}$ is a SMP, then there exists at least a minimizer in $U(\tilde{x},\rho)$ or $f(x)$ is constant on $\overline{U}(\tilde{x},\rho)$.

With these notations, the HiCS can be presented as the 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 $\bar{x}=\operatorname{argmin}_{x\in O(x_k,\,\rho)} f(x)$.
4: If $f(\bar{x})<f(x_k)$, then set $x_{k+1}=\bar{x}$.
5: Otherwise, declare that a SMP is found, and end the iteration.

In what follows, we will discuss the convergent property of the Algorithm 1.

Theorem 2.1 (Finite-step convergence)**.** *Suppose that the 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 $f(x^*)=f(x_*)$. Then Algorithm 1 converges in finite steps.*

Proof. Assume that the HiCS generates an infinite pair sequence $\{x_n,f(x_n)\}_{x=0}^{\infty}$. From these assumptions, 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.

According to 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 $\{x_{n_m+1}\}\subset\{x_{n_k}\}$ which satisfies $\|x_{n_m}-x_{n_m+1}\|=\rho$, 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. □

Theorem 2.1 demonstrates that the HiCS can converge to a SMP x^k in finite steps under reasonable conditions. If the stopping criterion is met, it implies that there exist minimizers in $U(x^k,\rho)$. The distance between the SMP x^k and minimizers is smaller than ρ. Therefore, the HiCS can efficiently shrink search domain Ω to the $U(x^k,\rho)$. At each iteration, the HiCS minimizes a $(d-1)$-dimensional subproblem of the d-dimensional objective function. In practice, HiCS requires to numerically solve the subproblem in the Algorithm 1. Next, we will discuss the implementation detailedly.

3 Implementation

As mentioned above, at each iteration, the HiCS has to discretize the $(d-1)$-dimensional search surface $O(x_k,\rho)$ to numerically imitate the minimization process of finding $\tilde{x}$. How to design an efficient method to discretize $O(x_k,\rho)$ is skillful. When without a priori information of the objective function, the discretization principles for $O(x_k,\rho)$ should include symmetric and uniform distribution and as few discretization points as possible. Previous results have shown that the uniformly distributed discretization method based on the spherical coordinate can be used to find the SMP [4]. The generated discretization points are as large as $2^d m$ in each iteration, m is the number of refinement, d is the dimension of the objective problem. The power increasing discretization points as the dimension d restricts the application to high-dimensional problems. To overcome this limitation, it needs to develop a new strategy to discretize the search surface $O(x_k,\rho)$ with fewer discretization points but still satisfying these properties. A reasonable requirement for discretization points should be linear or quasi-linear growth as the dimension of the objective problem increases. In this work, we will use the regular simplex and its rotations to discretize $O(x_k,\rho)$. The computational complexity grows linearly as d increases.

The d-dimensional regular simplex is a congruent polytope of $\mathbb{R}^d$ with a set of points $\{a_1,\ldots,a_d,a_{d+1}\}$, 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 $\{a_1,\ldots,a_d,a_{d+1}\}$ to its center are equal.

2. The angle subtended by any two vertices of the d-dimension simplex through its center is $\arccos(-1/d)$.

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

Algorithm 2 Generate d-D regular simplex coordinates

Given a $d\times(d+1)$-order zero matrix $x(1:d,1:d+1)$
Let $x(:,1)=(1,0,\ldots,0)$
for $i=2:1:d$ **do**
  $x(i,i)=\sqrt{1-\sum_{k=1}^{i-1}[x(k,i)]^2}$
  for $j=i+1:1:d+1$ **do**
    $x(i,j)=-\dfrac{1}{x(i,i)}\left[\dfrac{1}{d}+x(1:i-1,i)^T\cdot x(1:i-1,j)\right]$
  end for
end for
Output the column vectors, and let $a_j=x(:,j)$, $j=1,2,\ldots,d+1$

If the HiCS does not find a better state at the initial given regular simplex, more points can be added to discretize $O(x_k,\rho)$ by rotating regular simplex. For a given rotation angle

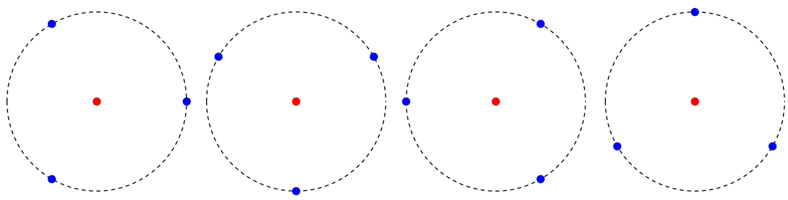
$\theta=(\theta_1,\theta_2,\ldots,\theta_d)$, the rotation matrix $\boldsymbol{Q}$ is given as

$$
\begin{aligned}
\boldsymbol{Q}=\prod_{i=2}^{d-1} &\begin{array}{c}
\begin{array}{ccccccccc} & & & & i & & & & \end{array} \\
\begin{pmatrix}
1 & & & & \vdots & & & & \\
 & \ddots & & & \vdots & & & & \\
 & & 1 & & \vdots & & & & \\
 & & & \cos\theta_i & 0 & -\sin\theta_i & & & \\
 & & & 0 & 1 & 0 & & & \\
 & & & \sin\theta_i & 0 & \cos\theta_i & & & \\
 & & & & & & 1 & & \\
 & & & & & & & \ddots & \\
 & & & & & & & & 1
\end{pmatrix}
\end{array} \\
&\begin{pmatrix}
\cos\theta_1 & -\sin\theta_1 & 0 & & \\
\sin\theta_1 & \cos\theta_1 & 0 & & \\
0 & 0 & 1 & & \\
 & & & \ddots & \\
 & & & & 1
\end{pmatrix}
\begin{pmatrix}
1 & & & & \\
 & \ddots & & & \\
 & & 1 & 0 & 0 \\
 & & 0 & \cos\theta_d & -\sin\theta_d \\
 & & 0 & \sin\theta_d & \cos\theta_d
\end{pmatrix}.
\end{aligned}
\tag{3.1}
$$

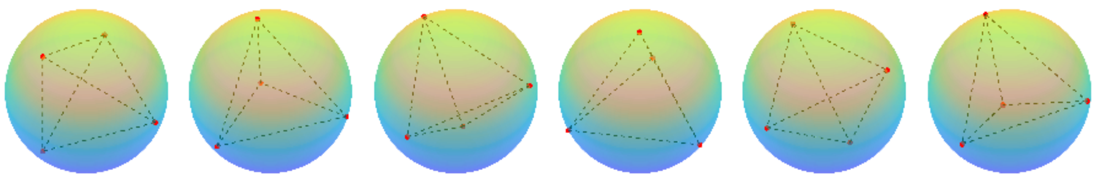
Then vertices of new simplex are

$$
a_j=\boldsymbol{Q}a_j+x_k, \quad j=1,\ldots,d+1. \tag{3.2}
$$

Standard two- and three-dimensional rotational simplexes which obey the uniform dis-



(a) Two-dimensional regular simplexes with rotation



(b) Three-dimensional regular simplexes with rotation

Figure 1: The regular simplexes of discretizing the search set $O(x_k,\rho)$ which obeys uniform distribution principle.

tribution principle are shown in Fig. 1. It should be noted that there could be other strategies to discretize $O(x_k,\rho)$ if more information is considered. For example, the new adding discretized points can be concentrated on these areas that the function values have a sufficient decrease.

To further save computational amount, we choose a dynamic refinement strategy to discretize the search surface and minimize the subproblem in practical implementation. These simplexes satisfies the uniform distribution principle. Based on the dynamic refinement strategy, we propose the practical HiCS, as the Algorithm 3 shows. The computational amount is not larger than $(d+1)m_{\max}$ in each iteration, $m_{\max}$ is the maximum number of rotation, which is linearly dependent on the dimension of optimization problems. Whence, it allows us to treat high-dimensional optimization problems.

Algorithm 3 Practical HiCS

Input x_0, $\rho>0$, and $m_{\max}\in\mathbb{Z}^+$
for $k=0,1,2,\cdots$ **do**
  Set $m=0$
  if $m\le m_{\max}$ **then**
    Discretize $O(x_k,\rho)$ to obtain $O_h^m(x_k,\rho)$. Find $x_j=\operatorname{argmin}_{x\in O_h^m(x_k,\rho)} f(x)$.
    if $f(x_j)<f(x_k)$ **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

It is evident that once the HiCS converges, the search space shrinks 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. Note that the convergent result provides a good initial value for other optimization approaches, including directional search and model-based algorithms.

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

Algorithm 4 Adaptive HiCS

Input x_0, ρ, ε and $\eta>0$ and $m_{\max}\in\mathbb{Z}^+$
if $\rho>\varepsilon$ **then**
  for $k=0,1,2,\cdots$ **do**
    Set $m=0$
    if $m\le m_{\max}$ **then**
      Discretize $O(x_k,\rho)$ 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
      Set $\rho=\eta\rho$
    end if
    Set $k=k+1$
  end for
end if

4 Numerical results

In this section, we choose two kinds of high-dimensional optimization functions, including the unimodal Gaussian function, multimodal problems, to demonstrate our proposed algorithm's performance. These objective functions are all differentiable. However, it is emphasized that the HiCS can be applied to non-differentiable problems. In Algorithm 3, the discretized 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 rotation $m=32$.

4.1 The unimodal problem: Gaussian function

The first objective function is the unimodal Gaussian problem

$$f(x)=-20\exp\left(-\sum_{j=1}^{d}x_j^2\right), \tag{4.1}$$

which has one minimum 0 with $f(0)=-20$. The objective function is differentiable in $\mathbb{R}^d$, however, it quickly diffuses out towards zero out of the upside-down "bell".

We first investigate the convergent property of HiCS for 10 dimensional Gaussian function using 30 experiments. In the set of experiments, the search radius ρ is fixed as 0.3, start points are all randomly generated in the space $[-1,1]^{10}$. For each experiment, the HiCS indeed converges and captures a neighborhood of the peak 0 in finite iterations as Theorem 2.1 predicts. Fig. 2 gives the required iterations for convergence in the 30 numerical experiments. In these 30 runs, the average iterations of convergence is 20.5,

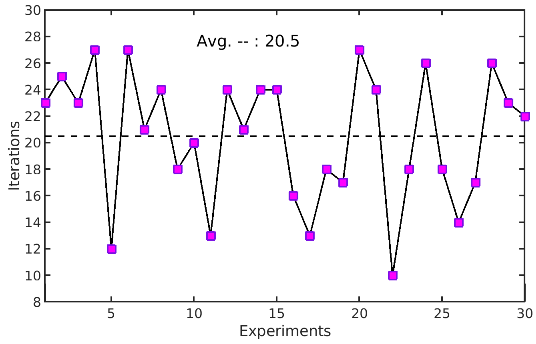


Figure 2: The required iteration steps of the HiCS for the Gaussian function (4.1) in 30 runs with randomly generated start points in the space $[-1,1]^{10}$, and $\rho=0.3$. The flat dashed line shows the average.

while the maximum is 27, and the minimum is 9.

Then we decrease the search radius ρ to 0.1 to observe the behavior of HiCS in 30 numerical tests. The initial values are also randomly generated in the same region. The required iterations for convergence are given in Fig. 3. In these 30 runs, the average iterations of convergence are 77.2, while the maximum is 121, and the minimum is 54. From these results, the HiCS converges in finite iterations. Meanwhile, it is obvious that the value of ρ affects the number of iterations.

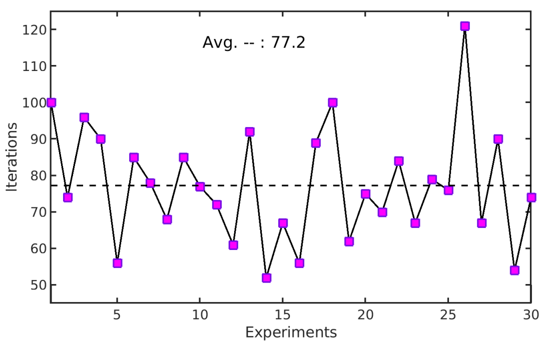


Figure 3: The required iteration steps of the HiCS for the Gaussian function (4.1) in 30 runs with randomly generated start points in the space $[-1,1]^{10}$, and $\rho=0.1$. The flat dashed line shows the average.

In the following, we apply the adaptive HiCS to 1000 dimensional Gaussian function. The initial value is randomly generated in domain $[-1000,1000]^{1000}$, the initial search radius $\rho_0=2.0$, and control factor $\eta=(\sqrt{5}-1)/2$. Fig. 4 presents the iteration process. The left image in Fig. 4 gives the difference between $f(x_k)$ and $f(0)=-20$. The right one in Fig. 4 plots the changes of search radius ρ and ℓ^2-distance between the iterator and the global minimizer $x^*=0$, where $\|x\|_{\ell^2}=\left((\sum_{i=1}^d x_i^2)/d\right)^{1/2}$. From these results, it can be found that the HiCS is convergent for each ρ. Based on the iteration, the adaptive HiCS can approximate the global minimum by decreasing the search radius ρ. Meanwhile, during the iteration, the global minimizer is always in the search neighbourhood.

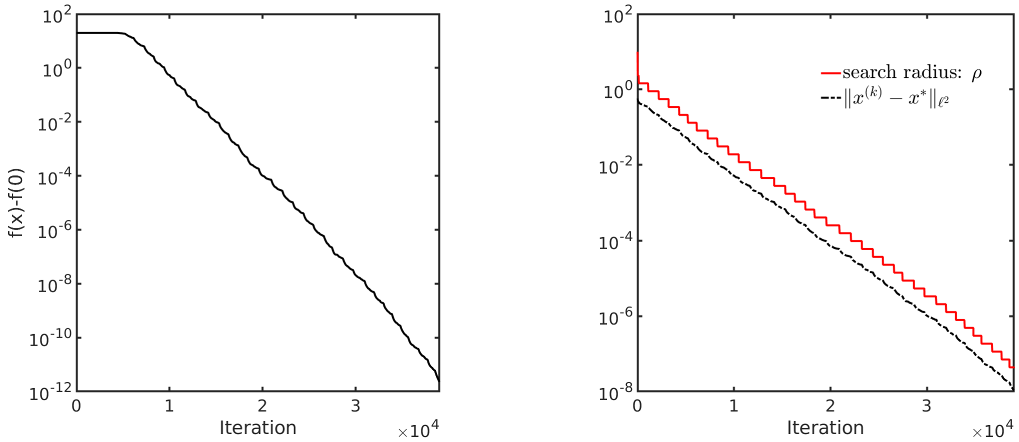


Figure 4: The iteration process of the adaptive HiCS to 1000 dimensional Gaussian function. Start point is randomly generated in the space $[-1000,1000]^{1000}$, $\rho=2.0$ and control factor $\eta=(\sqrt{5}-1)/2$. The left plot is the energy difference, and the right one is the search radius and the ℓ^2 distance between the current iterator and the global minimizer $x^*=0$.

4.2 The multimodal problems: Ackley and Arwhead functions

The second test objective function is the Ackley function [5] which is a widely used benchmark function for testing optimization algorithms. The expression of the Ackley function can be written as

$$f(x)=-20\cdot\exp\left(-\frac{1}{5}\cdot\sqrt{\frac{1}{d}\sum_{i=1}^d x_i^2}\right)-\exp\left(\frac{1}{d}\sum_{i=1}^d\cos(2\pi x_i)\right)+20+e, \tag{4.2}$$

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 [6]. Our previous result has shown that the HiCS can capture different local minimizer and the global minimizer for 2 dimensional problem through the choice of different ρ [4]. In this subsection, we will apply the improved HiCS to higher dimensional Ackley function. In the following simulation, the control factor $\eta=(\sqrt{5}-1)/2$.

We first take 100 dimensional Ackley function as an example to test the performance of our proposed algorithm for finding minimizers. We run adaptive HiCS 100 times for

Table 1: The successful number N_s of capturing the global minimizer for each different initial search radius ρ_0 when applying the adaptive HiCS to 100 dimensional Ackley function from 100 time numerical experiments. The initial values are randomly generated in $[-10,10]^{100}$.

ρ_0	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2
N_s	98	99	97	73	93	100	99	84	76	57
ρ_0	0.1	0.09	0.08	0.07	0.06	0.05	0.04	0.03	0.02	0.01
N_s	75	79	72	69	84	86	52	0	0	0

each different initial search radius ρ_0 from 0.01 to 2.0. The start points are all randomly generated in $[-10,10]^{100}$. The convergent criterion is the search radius smaller than 10^{-10}. Tab. 1 gives the successful number N_s of capturing the global minimizer. When the algorithm is successful, the distance between the convergent iterator and the global minimizer is smaller than the search radius $\rho < 10^{-10}$. From these results, it is easy to find that our method can approximate the global minimizer. The value of ρ_0 heavily affects the probability of obtaining the global minimizer. When $\rho_0 > 0.04$, the adaptive HiCS can find the global minimizer with high probability. When ρ_0 is about 0.04, the successful probability is falling quickly to about 50%. As ρ_0 decreases to smaller than 0.03, the HiCS could not find the global minimizer. Besides, it should be pointed out that these so-called unsuccessful experiments have obtained other local minimizers.

We continue to apply the adaptive HiCS to 2500 dimensional Ackley function. The initial search radius is ρ_0=3.5, and the initial position is generated randomly in $[-10,10]^{2500}$. The iteration process is presented in Fig. 5. For such a high dimensional optimization problem, the iteration behavior is similar to previous numerical experiments. When $\rho = 3.5$, the HiCS costs 90 steps to achieve convergence. By further shrinking search radius, the adaptive HiCS can capture global minimizer. As one can see from Fig 5, the global minimizer always locates in the search neighbourhood in this case. It demonstrates that the HiCS has the capacity of hupping the local basin even for such a high dimensional problem.

The last benchmark example is the Arwhead function, which has been also used by Powell to test the NEWUOA derivative-free method [7]. The expression of the Arwhead function is

$$f(x) = \sum_{i=1}^{d-1} [(x_i^2 + x_n^2)^2 - 4x_i + 3]. \tag{4.3}$$

The least value of f is zero, which occurs when the minimizer x^* take the values x_j=1, j= 1,2,...,$d-1$ and $x_d = 0$. We directly apply the adaptive HiCS ($\eta = 0.5$) to 1000 dimensional Arwhead function. The starting vector is given by $x_j^{(0)} = 1$, $j = 1,2,\ldots,d$, as Powell done in Ref. [7] The initial search radius $\rho_0 = 3$ and $\eta = (\sqrt{5}-1)/2$.

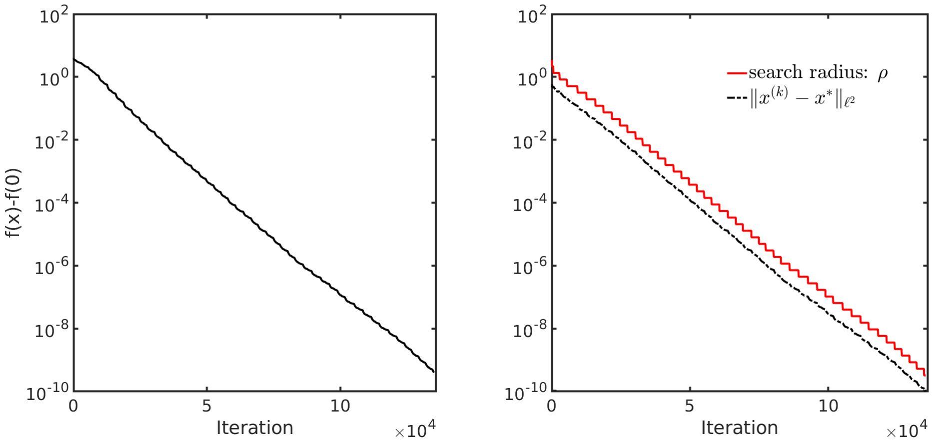


Figure 5: The iteration process of the adaptive HiCS to 2500 dimensional Ackley function with initial search radius $\rho_0 = 3.5$. Start point is randomly generated in the space $[-10,10]^{2500}$.

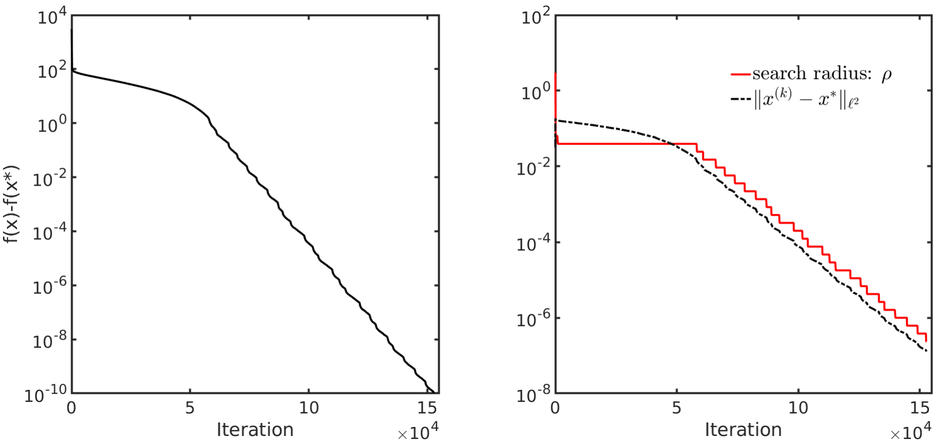


Figure 6: The iteration process of the adaptive HiCS ($\eta=(\sqrt{5}-1)/2$) to the 1000 dimensional Arwhead function.

Fig. 6 gives the iteration process of applying the adaptive HiCS to 1000 dimensional Arwhead function. The sequences of function values and iterators approximate the global minimum and the global minimizer. The function value always decreases as the proposed algorithm indicates. While the distance $\|x^{(k)} - x^*\|_{\ell^2}$ demonstrates more interesting phenomena. In the beginning, the search radius ρ is larger than the distance, which means the global minimizer x^* is in the search neighborhood. Then when the distance is about 1.67×10^{-1}, the ρ is smaller than the distance, which indicates x^* is not in the search neighborhood. It means that the iterator locates in the valley of a local minimizer. However, as iteration evolves, the HiCS can jump out of the local energy trap well and again contains the global minimizer in the search region.

5 Conclusion

Inspired by the hill-climbing behavior of the blind, we have proposed a new derivative-free method to unconstrained optimization problems in our previous work [4]. This paper establishes a rigorous mathematical theory of the HiCS, which theoretically guarantees the finite-step convergence under mild conditions. Numerical results also have demonstrated the satisfactory property. In practice, the computational complexity of the HiCS mainly depends on the discretized strategy on search boundaries. We proposed a new simplex discretization method to save computational amount to ddress high-dimensional problems in this work. Using the simplex method, the number of function evaluations is linearly dependent on the dimension of problems, which allows us to solve high-dimensional optimization problems. Finally, we demonstrate the efficiency of our proposed algorithm by solving several higher-dimensional benchmark problems.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (11971410, 11771368) and Project for Hunan National Applied Mathematics Center of Hunan Provincial Science and Technology Department (2020ZYT003). KJ is partially supported by the Key Project of the Education Department of Hunan Province of China (19A500).

References

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

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

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

[4] Y. Huang and K. Jiang, Hill-climbing algorithm with a stick for unconstrained optimization problems, Adv. Appl. Math. Mech., 2017, 9: 307–323.

[5] J. M. Dieterich and B. Hartke, Empirical review of standard benchmark functions using evolutionary global optimization, Appl. Math. 2012, 3: 1552–1564.

[6] T. Bäck, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Oxford University Press, 1996.

[7] 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.