

# A finite-step convergent derivative-free method for unconstrained optimization

Yunqing Huang

Xiangtan University

joint work with Kai Jiang

June 3, 2019

# Outline

- 1 Motivation
- 2 Algorithm Description
- 3 Numerical Results
- 4 Summary

# Outline

1 Motivation

2 Algorithm Description

3 Numerical Results

4 Summary

# Motivation: the blind climbs mountains

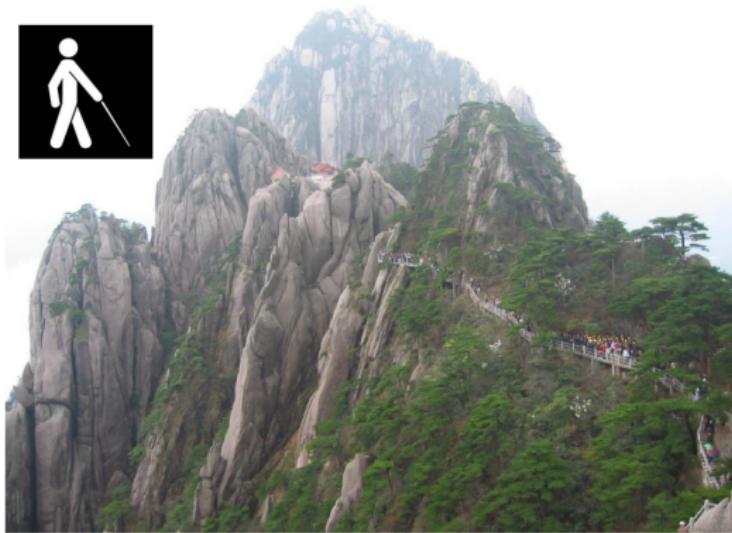
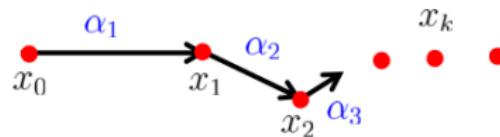


Figure 1 The picture is come from Yuan's talk.

- Find a higher place with a stick in a circular motion
- No predetermined search directions

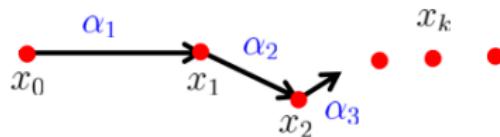
# How to search ?

- directional search method = direction + step length

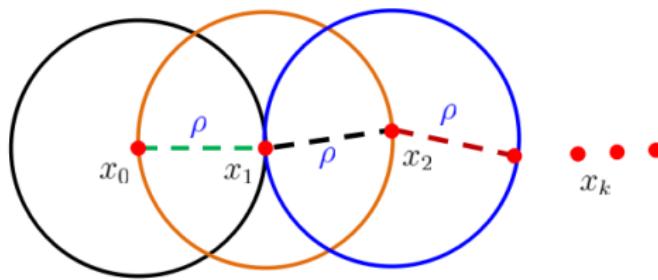


# How to search ?

- directional search method = direction + step length



- boundary search: no direction, no step length



# Outline

1 Motivation

2 Algorithm Description

3 Numerical Results

4 Summary

# Suspected minimum point

## Definition (Suspected minimum point)

For a given function  $f(x)$ ,  $\tilde{x}$  is a suspected minimum point (SMP) if  $f(\tilde{x}) \leq f(x)$  for  $\forall x \in \partial U(\tilde{x}, \rho)$ , where  $\partial U(\tilde{x}, \rho) = \{y : \|y - \tilde{x}\| = \rho\}$  is the boundary of a neighborhood  $U(\tilde{x}, \rho)$  of  $\tilde{x}$ .

- When  $\tilde{x}$  is a minimizer in the neighborhood of  $U(\tilde{x}, \rho)$ ,  $\tilde{x}$  is a SMP.
- The opposite is not always true.
- The distance of the SMP and a minimizer is smaller than  $\rho$ .

# Hill-Climbing method with a Stick (HiCS)

- Consider the unconstrained optimization problem

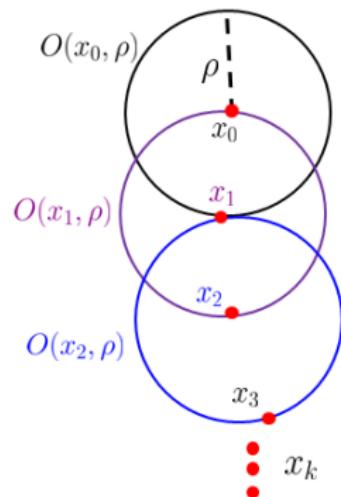
$$\min_x f(x, \gamma), \text{ where } f : \mathbb{R}^n \mapsto \mathbb{R}.$$

---

## Algorithm 1 HiCS

---

- Initialization:** given  $x_0$  and  $\rho$ .
- for**  $k = 0, 1, 2, \dots$ 
  - Find  $\bar{x} = \operatorname{argmin}_{y \in O(x_k, \rho)} f(y)$
  - If  $f(\bar{x}) < f(x_k)$ , then set  $x_{k+1} = \bar{x}$ .
  - Otherwise, a SMP is found, and declare the iteration successful.



# Finite-Step Convergence

## Theorem (Finite-step convergence)

*Suppose that objective function  $f(x)$  is continuous and the search domain  $\Omega$  is a compact set. If there are not two SMPs  $x_*$  and  $x^*$  satisfying  $f(x^*) = f(x_*) = A$  when  $|x_* - x^*| = \rho$ . Then HiCS algorithm converges in finite steps.*

# Finite-Step Convergence

## Theorem (Finite-step convergence)

Suppose that objective function  $f(x)$  is continuous and the search domain  $\Omega$  is a compact set. If there are not two SMPs  $x_*$  and  $x^*$  satisfying  $f(x^*) = f(x_*) = A$  when  $|x_* - x^*| = \rho$ . Then HiCS algorithm converges in finite steps.

## Proof.

- The stick hill-climbing method produces  $\{x_n, f(x_n)\}_{n=0}^\infty$ .
- From the assumption,  $f(x)$  is bounded, then the decreasing sequence  $\{f(x_n)\}$  is convergent. Assume that  $f(x_n) \rightarrow A$ .
- Exists a convergent subsequence  $\{x_{n_k}\} \subset \{x_n\}$ , s.t.  $x_{n_k} \rightarrow x^*$ .  $x^*$  is a SMP.
- We have a bounded subsequence  $\{x_{n_k-1}\} \subset \{x_n\}$  which satisfies  $|x_{n_k} - x_{n_k-1}| = \rho$ . Obviously,  $f(x_{n_k-1}) \rightarrow A$ .
- $\{x_{n_k-1}\}$  has a convergent subsequence  $\{x_{n_m}\}$ , s.t.  $x_{n_m} \rightarrow x_*$ .  $x_*$  is also a SMP.
- From  $\{x_{n_m}\}$ , we have a subsequence  $\{x_{n_m+1}\} \subset \{x_{n_k}\}$  satisfying  $|x_{n_m} - x_{n_m+1}| = \rho$ , and  $x_{n_m+1} \rightarrow x^*$ .
- We have  $|x^* - x_*| = \rho$ , but  $f(x^*) = f(x_*) = A$ .

# Finite-Step Convergence

## Theorem (Finite-step convergence)

Suppose that objective function  $f(x)$  is continuous and the search domain  $\Omega$  is a compact set. If there are not two SMPs  $x_*$  and  $x^*$  satisfying  $f(x^*) = f(x_*) = A$  when  $|x_* - x^*| = \rho$ . Then HiCS algorithm converges in finite steps.

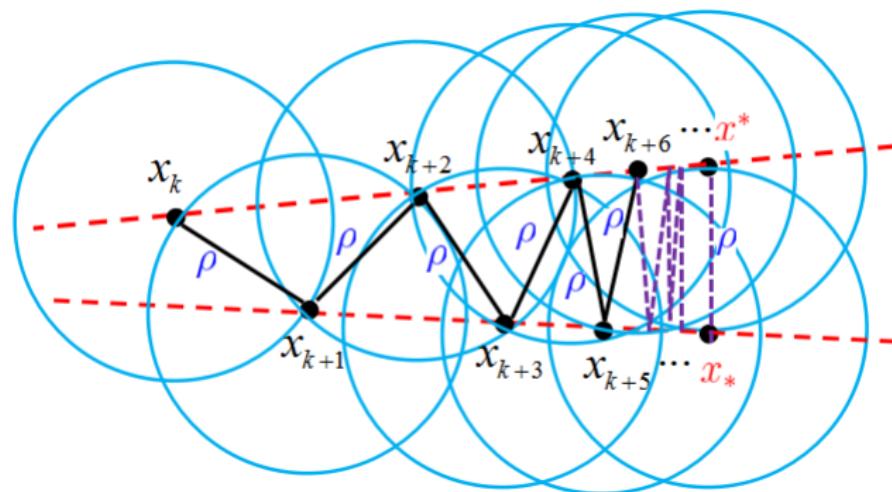
## Proof.

- The stick hill-climbing method produces  $\{x_n, f(x_n)\}_{n=0}^\infty$ .
- From the assumption,  $f(x)$  is bounded, then the decreasing sequence  $\{f(x_n)\}$  is convergent. Assume that  $f(x_n) \rightarrow A$ .
- Exists a convergent subsequence  $\{x_{n_k}\} \subset \{x_n\}$ , s.t.  $x_{n_k} \rightarrow x^*$ .  $x^*$  is a SMP.
- We have a bounded subsequence  $\{x_{n_k-1}\} \subset \{x_n\}$  which satisfies  $|x_{n_k} - x_{n_k-1}| = \rho$ . Obviously,  $f(x_{n_k-1}) \rightarrow A$ .
- $\{x_{n_k-1}\}$  has a convergent subsequence  $\{x_{n_m}\}$ , s.t.  $x_{n_m} \rightarrow x_*$ .  $x_*$  is also a SMP.
- From  $\{x_{n_m}\}$ , we have a subsequence  $\{x_{n_m+1}\} \subset \{x_{n_k}\}$  satisfying  $|x_{n_m} - x_{n_m+1}| = \rho$ , and  $x_{n_m+1} \rightarrow x^*$ .
- We have  $|x^* - x_*| = \rho$ , but  $f(x^*) = f(x_*) = A$ . **Contradiction**

# Finite-Step Convergence

## Theorem (Finite-step convergence)

Suppose that objective function  $f(x)$  is continuous and the search domain  $\Omega$  is a compact set. If there are not two SMPs  $x_*$  and  $x^*$  satisfying  $f(x^*) = f(x_*) = A$  when  $|x_* - x^*| = \rho$ . Then HiCS algorithm converges in finite steps.



Subproblem:  $x = \operatorname{argmin}_{y \in O(x_k, \rho)} f(y)$

### Definition (Sublevel set)

$$\forall \alpha \in \mathbb{R}, S_\alpha = \{x | f(x) \leq \alpha\}$$

### Theorem (Weierstrass)

Suppose that the set  $\mathcal{D} \in \mathcal{E}$  (a finite dimensional vector space over  $\mathbb{R}^n$ ) is the nonempty and closed, and that all sublevel sets of a continuous function  $f : \mathcal{D} \mapsto \mathbb{R}$  are bounded. Then  $f$  has a global minimizer.

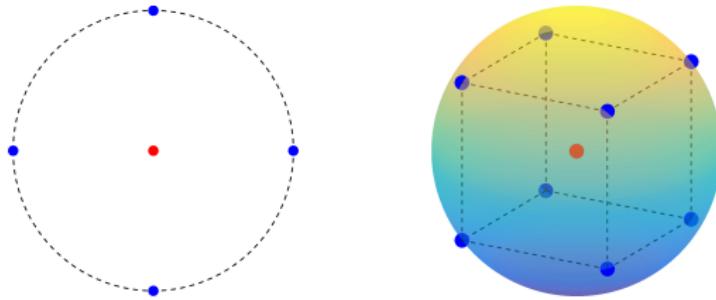
$$O(x_k, \rho) \rightarrow O_h(x_k, \rho)$$

The principles of sampling  $O(x_k, \rho)$  *without a priori information of objective function*

- Symmetry
- Uniform distribution
- As few points as possible

$$O(x_k, \rho) \rightarrow O_h(x_k, \rho)$$

- Spherical coordinate frame
  - Not uniform distribution
  - Sampling points:  $2m^{n-1}$ ,  $n$  is dimension,  $m$  is the number of equal division points in each direction
- Bisection method
  - Symmetry, uniform distribution
  - Sampling points:  $2^n$



# $O(x_k, \rho) \rightarrow O_h(x_k, \rho)$ : Regular Simplex

- Generate  $n$ -D regular simplex coordinates ( $n + 1$  vertices):
  - 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  $\theta = \arccos(-1/n)$ .

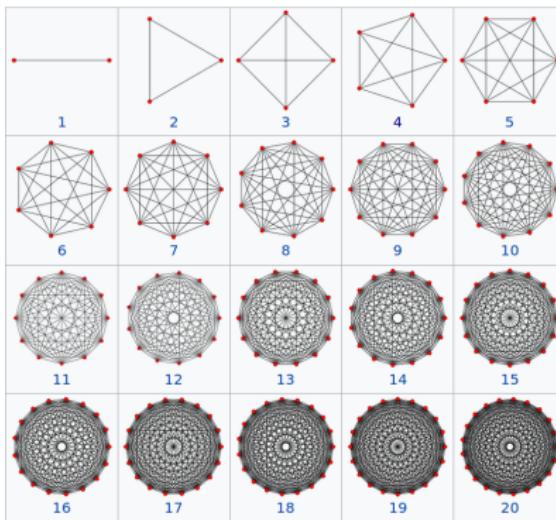


Figure 2: cited from wiki

# Refinement Strategy

- Principle: No-repeat, Uniform distribution

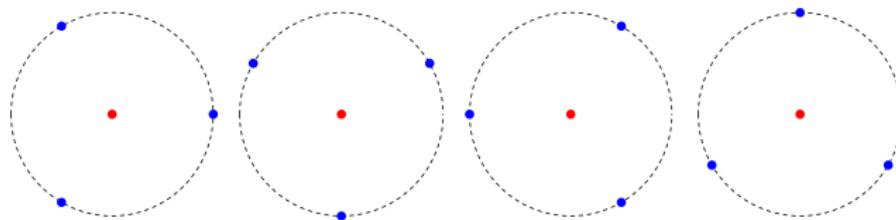
# Refinement Strategy

- Principle: No-repeat, Uniform distribution
- Strategy: Rotation

# Refinement Strategy

- Principle: No-repeat, Uniform distribution
- Strategy: Rotation
- Simplex: rotation matrices can be obtained through Euler angles

① 2D case:



② 3D case:

Movie

③ Computational complexity:  $m(n + 1)$

- In practice, the HiCS algorithm can be given as follows

---

## Algorithm 2 HiCS

---

```
1: Input  $x_0$ ,  $\rho$ , and  $m_{\max}$ 
2: for  $k = 0, 1, 2, \dots$  do
3:   Set  $m = 0$ 
4:   if  $m \leq m_{\max}$  then
5:     Discrete  $O(x_k, \rho)$  to obtain  $O_h^m(x_k, \rho)$ 
6:     if  $\exists x_j \in O_h^m(x_k, \rho)$ , s.t.  $f(x_j) < f(x_k)$  then
7:       Set  $x_{k+1} = x_j$ , and  $m = m_{\max} + 1$ 
8:     else
9:       Set  $m = m + 1$ 
10:    end if
11:   else
12:     Declare that find a SMP, end program
13:   end if
14: end for
```

---

# Adaptive HiCS (AHiCS): adjust $\rho$

- We can repeat HiCS algorithm through changing  $\rho$

---

## Algorithm 3 AHiCS

---

```
1: Input  $x_0$ ,  $\rho$ ,  $m_{\max}$ ,  $\varepsilon$  and  $\eta < 1$ 
2: if  $\rho > \varepsilon$  then
3:   for  $k = 0, 1, 2, \dots$  do
4:     Set  $m = 0$ 
5:     if  $m \leq m_{\max}$  then
6:       Discrete  $O(x_k, \rho)$  to obtain  $O_h^m(x_k, \rho)$ 
7:       if  $\exists x_j \in O_h^m(x_k, \rho)$ , s.t.  $f(x_j) < f(x_k)$  then
8:         Set  $x_{k+1} = x_j$ , and  $m = m_{\max} + 1$ 
9:       else
10:        Set  $m = m + 1$ 
11:      end if
12:    else
13:      Set  $\rho = \eta\rho$ 
14:    end if
15:    Set  $k = k + 1$ 
16:  end for
17: end if
```

---

# Adaptive HiCS (AHiCS): adjust $\rho$

- We can repeat HiCS algorithm through changing  $\rho$

---

## Algorithm 4 AHiCS

---

```
1: Input  $x_0$ ,  $\rho$ ,  $m_{\max}$ ,  $\varepsilon$  and  $\eta < 1$ 
2: if  $\rho > \varepsilon$  then
3:   for  $k = 0, 1, 2, \dots$  do
4:     Set  $m = 0$ 
5:     if  $m \leq m_{\max}$  then
6:       Discrete  $O(x_k, \rho)$  to obtain  $O_h^m(x_k, \rho)$ 
7:       if  $\exists x_j \in O_h^m(x_k, \rho)$ , s.t.  $f(x_j) < f(x_k)$  then
8:         Set  $x_{k+1} = x_j$ , and  $m = m_{\max} + 1$ 
9:       else
10:        Set  $m = m + 1$ 
11:      end if
12:    else
13:      Set  $\rho = \eta\rho$ 
14:    end if
15:    Set  $k = k + 1$ 
16:  end for
17: end if
```

---

- The restart mechanism can be obtained by set  $\eta > 1$

# Outline

1 Motivation

2 Algorithm Description

3 Numerical Results

4 Summary

# Gauss Function

$$f(x) = -20 \exp\left(-\sum_{i=1}^n x_i^2\right), \quad i = 1, \dots, n.$$

- A unique global minimum 0 with  $f(0) = -20$ .

# Gauss Function

$$f(x) = -20 \exp\left(-\sum_{i=1}^n x_i^2\right), \quad i = 1, \dots, n.$$

- A unique global minimum 0 with  $f(0) = -20$ .
- Apply HiCS to 2D Gauss function with  $x_0 = (0.57, -0.6)$ ,  $\rho = 0.3$ ,  $m_{max} = 4$ .

Iter.	$\ell^2$ -distance	Fun. Val.
1	8.27587e-01	-1.00828e+01
1	5.40491e-01	-1.49334e+01
1	2.81712e-01	-1.84741e+01
1	2.15853e-01	-1.90895e+01
4	8.58004e-02	-1.98533e+01

# Gauss Function

$$f(x) = -20 \exp\left(-\sum_{i=1}^n x_i^2\right), \quad i = 1, \dots, n.$$

- A unique global minimum 0 with  $f(0) = -20$ .
- Apply HiCS to 2D Gauss function with  $x_0 = (0.57, -0.6)$ ,  $\rho = 0.3$ ,  $m_{max} = 4$ .

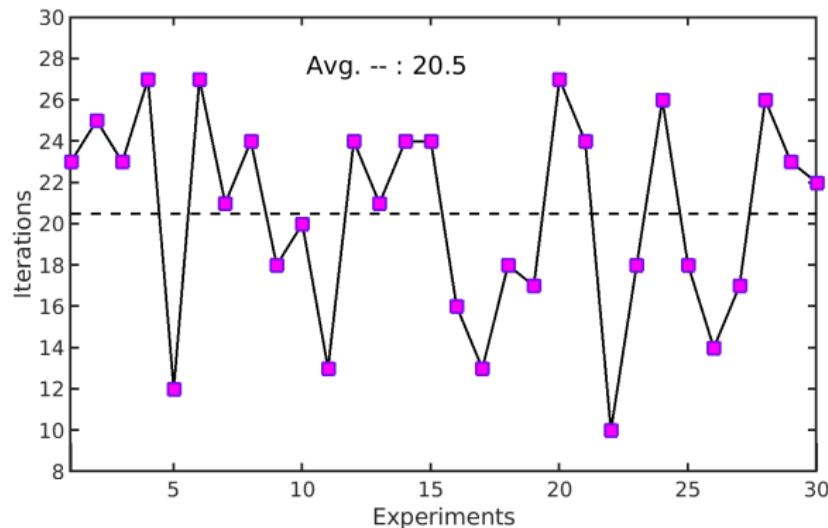
Iter.	$\ell^2$ -distance	Fun. Val.
1	8.27587e-01	-1.00828e+01
1	5.40491e-01	-1.49334e+01
1	2.81712e-01	-1.84741e+01
1	2.15853e-01	-1.90895e+01
4	8.58004e-02	-1.98533e+01

- We can further apply AHiCS approach ( $\eta = 0.5$ ) to this case.

Movie: Iteration procedure

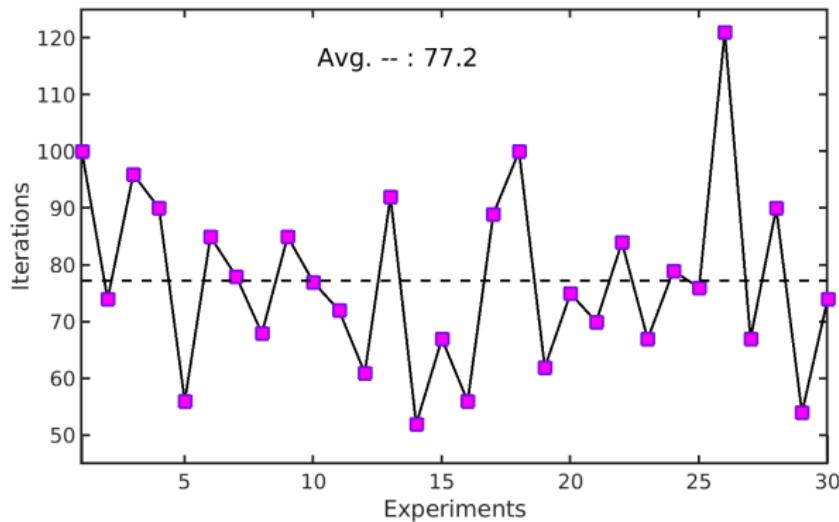
# 10D Gauss Function: finite-step convergence

- HiCS can capture the global minimizer within finite-step iterations
- The iterations of convergence of the HiCS with constant  $\rho = 0.3$  for 10D Gauss function. The initial values were randomly generated in  $[-1, 1]^2$ .



# 10D Gauss Function: finite-step convergence

- Decreasing  $\rho$  to 0.1 for 10D Gauss function. The initial values were randomly generated in  $[-1, 1]^2$ .



# 1000D Gauss Function: AHiCS

- AHiCS can approximate the minimizer through shrinking  $\rho$

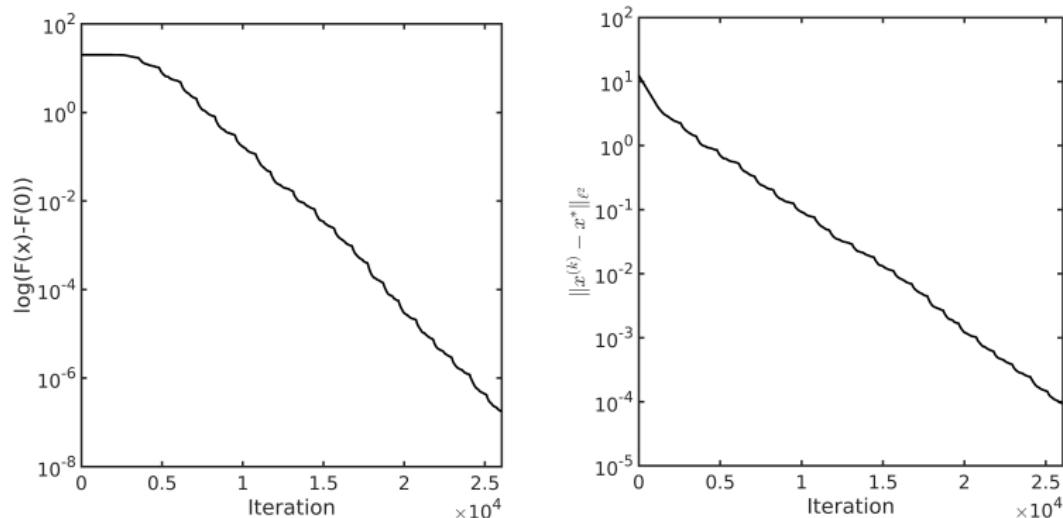


Figure 3: The initial value was randomly generated in  $[-1, 1]^{1000}$ .  $\rho_0 = 0.3$ ,  $\eta = (\sqrt{5} - 1)/2$ , and  $m_{\max} = 32$ .

# Dennis-Woods Function

$$f(z) = \frac{1}{2} \max\{\|z - c_1\|^2, \|z - c_2\|^2\}, \quad z = (x, y),$$

where  $c_1 = (1, -1)^T$ ,  $c_2 = -c_1$ ,  $\|\cdot\|$  denotes  $\ell^2$ -norm.

- continuous, strictly convex, but its gradient is **discontinuous** along the line  $x = y$

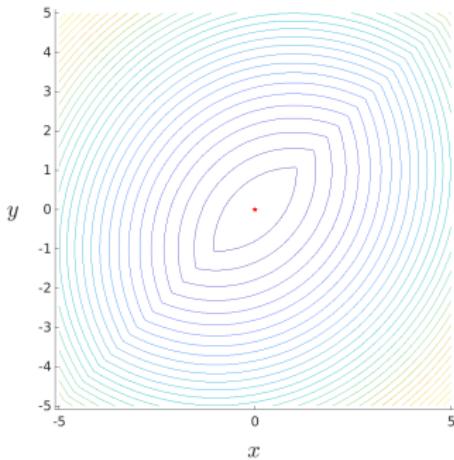


Figure 4: Contours of the variant of the Dennis-Woods function

# Dennis-Woods Function

- HiCS can capture the minimizer within finite-step iterations

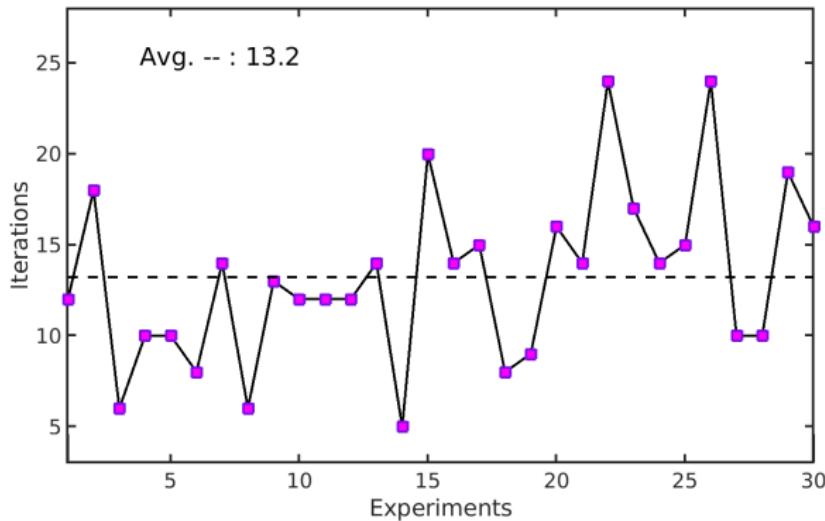


Figure 5: The iterations of convergence of the HiCS with constant  $\rho = 0.5$  for 2D Gauss function. The initial values were randomly generated in  $[-5, 5]^2$ .

# Dennis-Woods Function

- HiCS: initial value  $x_0 = (3.2, 1.5)$ ,  $\rho = 0.5$ ,  $m_{max} = 32$ .

Iter.	$\ell^2$ -distance	Fun. Val.
1 (1-15)	3.53412 ↓ 3.79272e-01	8.94500 ↓ 1.13987
3	1.34218e-01	1.12407
15	<b>3.89309e-01</b>	1.09854
32	1.12185e-01	1.02906

# Dennis-Woods Function

- HiCS: initial value  $x_0 = (3.2, 1.5)$ ,  $\rho = 0.5$ ,  $m_{max} = 32$ .

Iter.	$\ell^2$ -distance	Fun. Val.
1 (1-15)	3.53412 ↓ 3.79272e-01	8.94500 ↓ 1.13987
3	1.34218e-01	1.12407
15	<b>3.89309e-01</b>	1.09854
32	1.12185e-01	1.02906

Movie: Iteration procedure

# Dennis-Woods Function: Comparison

- The popular Nelder-Mead simplex algorithm **fails**
- The coordinate-search (CS) method (with predetermined search directions  $\mathcal{D}_{\oplus}$ ) **stalls**

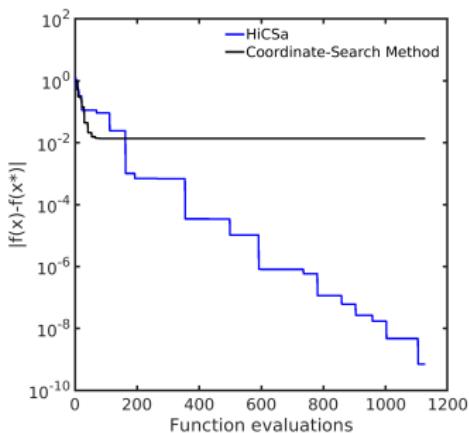


Figure 6 Application of the adaptive stick hill-climbing method ( $m_{\max} = 8$ ) and the CS method with  $\mathcal{D}_{\oplus} = \{(1, 0), (0, 1), (-1, 0), (0, -1)\}$ , starting from  $x_0 = (1.1, 0.9)$ . The initial search radii are both  $\rho_0 = 1.0$ , the control factor  $\eta = 0.5$ .

# Ackley Function

- Many local minima and a unique global minimum 0 with  $f(0) = 0$ .
- Non-convex

$$f(x) = -20 \cdot \exp \left( -0.2 \cdot \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$$

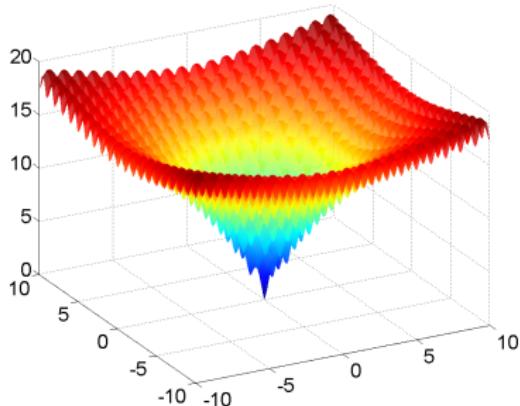


Figure 7: 2D Ackley function

# Ackley Function: 2D

- AHiCS: initial value  $x_0 = (6.12, 5.426)$ ,  $\rho = 2.5$

Movie: Iteration procedure

# Ackley Function: 2D

- Has potential to distinguish different minima

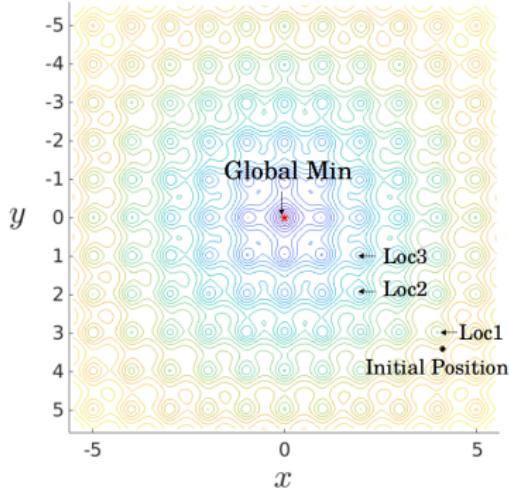


Table 1: The convergent results and required iterations with differently constant  $\rho$  starting from the initial state  $(x_0, y_0) = (4.1, 3.4)$ .

$\rho$	0.1	0.3	0.5	0.55	0.58	0.6	1.0
Iter.	7	3	2	6	7	15	7
Min.	Loc <sub>1</sub>	Loc <sub>1</sub>	Loc <sub>1</sub>	Loc <sub>2</sub>	Loc <sub>3</sub>	Global	Global

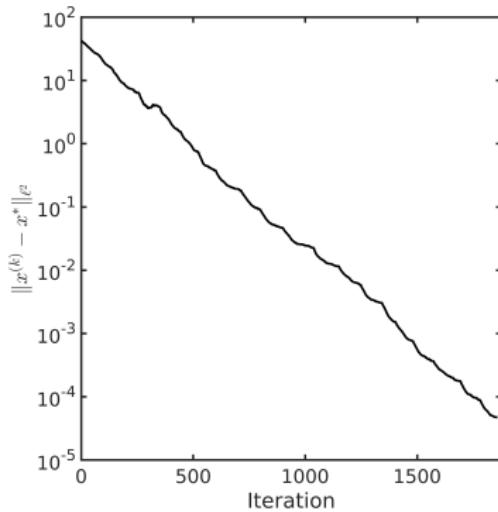
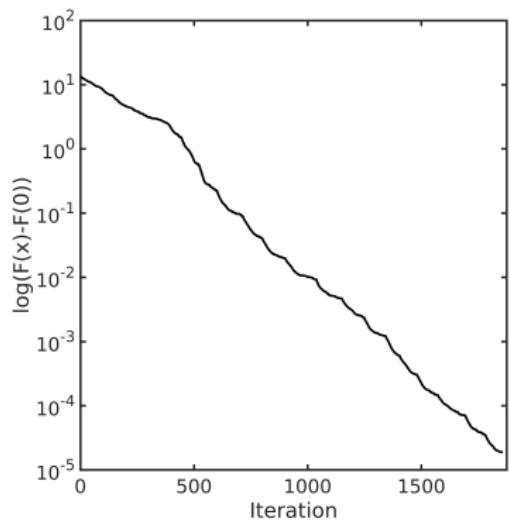
# Ackley Function: 100D

- We apply HiCS to 100D Ackley with constant  $\rho_0 = 2$ . The initial value is randomly generated in  $[-10, 10]^{100}$ .  $m_{\max} = 32$ .

Iter.	$\ell^2$ -distance	Fun. Val.
1 (1-353)	43.76984 ↓ 5.67057	13.40276 ↓ 3.65003
3	<b>5.76768</b>	3.64961
5	<b>5.84103</b>	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	<b>5.84496</b>	3.59674
1	5.76528	3.59377
1	5.71754	3.59277
2	<b>5.90908</b>	3.59054
32 ( $m_{\max}$ )	<b>5.93767</b>	3.58252

# Ackley Function: 100D

- We can continue to apply AHiCS ( $\eta = 0.5$ ) to 100D Ackley.



# Ackley Function: 100D

- AHiCS can capture different local minima with smaller  $\rho_0$ .
- Setup: the start point is randomly generated in  $[-5, 5]^{100}$ ,  $m_{\max} = 32$ ,  $\eta = 0.5$  and  $r < 1.0 \times 10^{-14}$ .

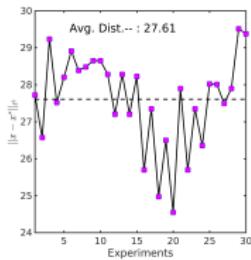


Figure 8:  $\rho_0 = 0.05$ .

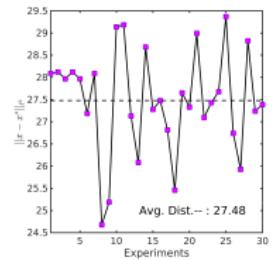
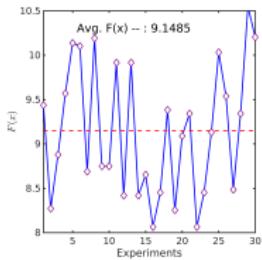


Figure 9:  $\rho_0 = 0.1$ .

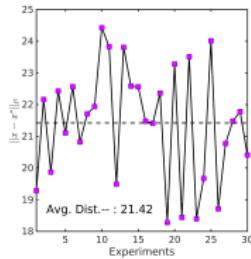


Figure 10:  $\rho_0 = 0.5$ .

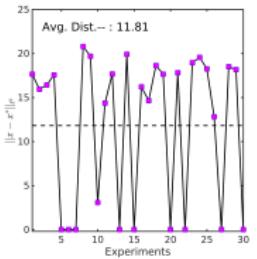
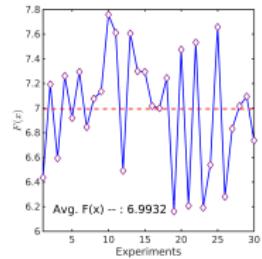
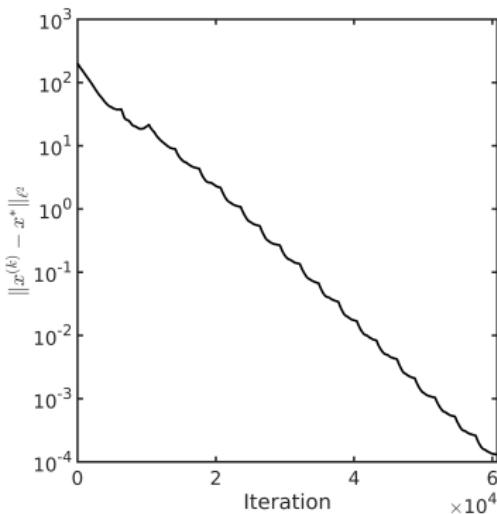
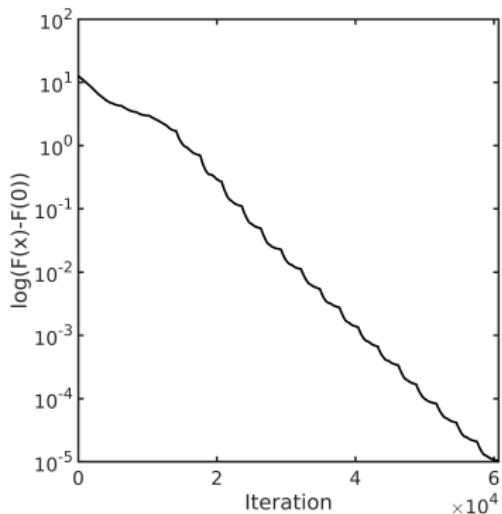


Figure 11:  $\rho_0 = 0.8$ .

# Ackley Function: 2500D

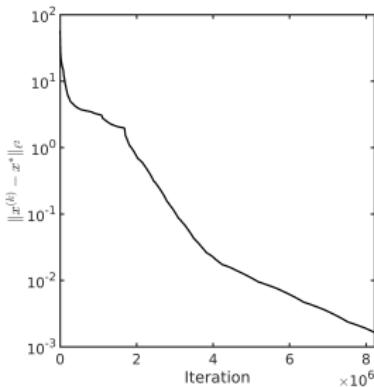
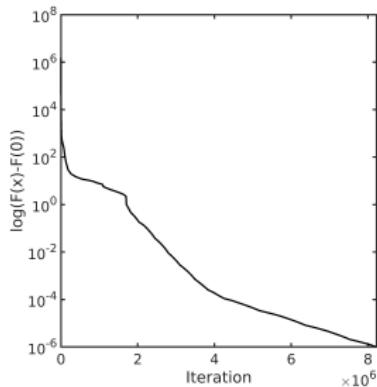
- Apply AHiCS algorithm ( $\eta = 0.5$ ) to 2500D Ackley function with random initial value in  $[-10, 10]^{2500}$  and  $\rho_0 = 3.5$ ,  $m_{max} = 16$ .



# Woods Function (CUTE)

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

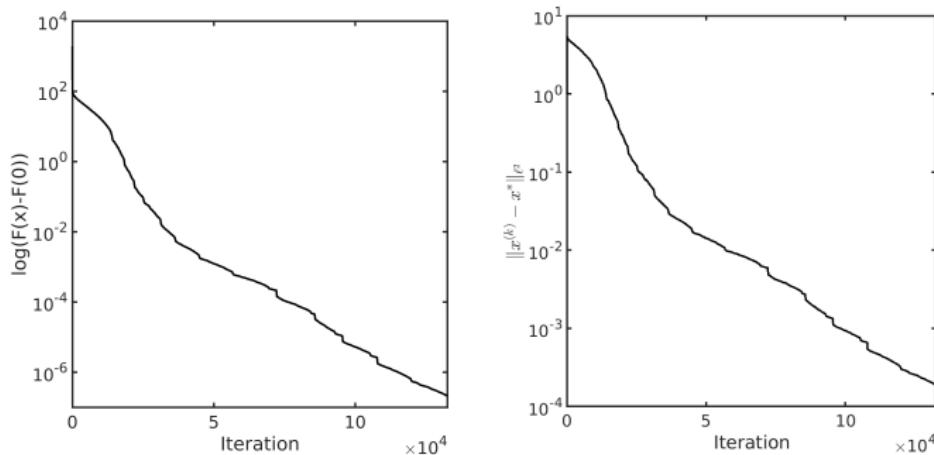
The global minimizer is  $(1, 1, \dots, 1)$  with  $f = 0$ . The initial value  $x_j^{(0)} = -3.0$  if  $j$  is even, and  $x_j^{(0)} = -1.0$  if  $j$  odd. We apply AHiCS algorithm to 320D Woods functions with  $\rho_0 = 5$  and  $m_{\max} = 8$ .



# ARWHEAD Function

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

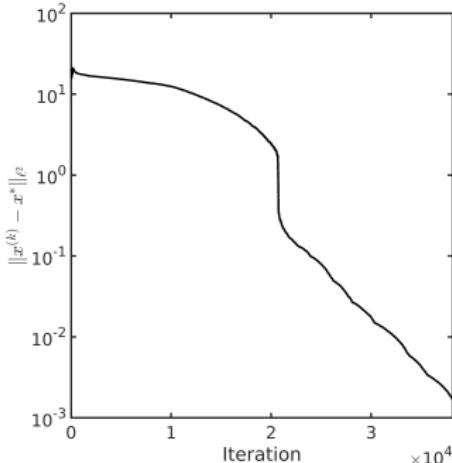
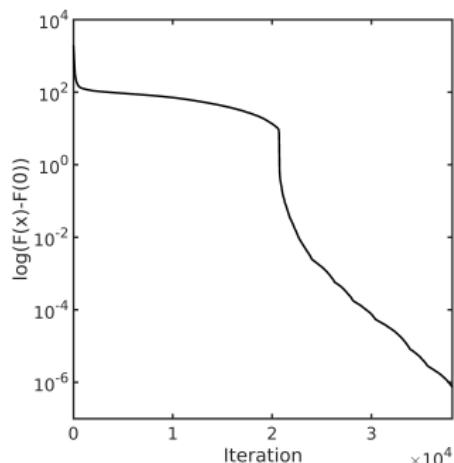
The least value of  $F$  is zero, which occurs when the variables take the values  $x_j = 1$ ,  $j = 1, 2, \dots, n-1$  and  $x_n = 0$ . We apply AHiCS method ( $\eta = 0.5$ ) to 640D ARWHEAD function. The starting vector is given by  $x_j^{(0)} = 1$ ,  $j = 1, 2, \dots, n$ , as Powell (2006) done. The search radius  $\rho_0 = 3$  and  $m_{\max} = 32$ .



# CHROSEN Function

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

The least value of  $F$  is zero, which occurs when the variables take the values  $x_j = 1$ ,  $j = 1, 2, \dots, n$ . We apply AHiCS method ( $\eta = 0.5$ ) to 100D CHROSEN function. The starting vector is given by  $x_j^{(0)} = -1$ ,  $j = 1, 2, \dots, n$ , as Powell (2006) done. The search radius  $\rho_0 = 5$  and  $m_{\max} = 32$ .



# Outline

1 Motivation

2 Algorithm Description

3 Numerical Results

4 Summary

# Summary

- Propose a Hill-Climbing method with a Stick (**HiCS**)
- Theory: Finite-step convergence
- An efficient method of obtaining  $O_h(x_k, \rho)$ : Simplex
- Be able to implement to high dimension optimization
- Other properties:
  - Only one parameter  $\rho$  which can be adjusted
  - Capture the neighborhood of a minimizer
  - Has potential to approximate the global minimizer

# Thanks for Your Attention !