Article ID: 1000 – 8152(2002)03 – 0431 – 04

Adaptive Quasi-Monte Carlo Method for Multiple-Extrema Optimization *

LEI Guiyuan

(Department of Mathematics, Zhejiang University · Hangzhou, 310028, P.R.China)

Abstract: Quasi-Monte Carlo random search is useful in nondifferentiable optimization. By borrowing the ideas of population from genetic algorithms, we introduce an adaptive random search in quasi-Monte Carlo method(AQMC) for global optimization. The adaptive search technique enables local search to head for local extrema quickly. The low discrepancy of quasi-random sequence ensures that the function field be searched evenly and various local extrema including global extremum be found.

Key words: global optimization; nondifferentable function; quasi-Monte Carlo methods; adaptive random search; subpopulation

Document code: A

自适应拟蒙特卡罗多极值优化方法

雷桂媛

(浙江大学数学系·杭州,310028)

摘要: 拟蒙特卡罗搜索方法能用来有效地解决不可微优化问题.借用遗传算法中种群的概念,介绍了一种解全局优化的拟蒙特卡罗自适应搜索算法.由于应用了自适应搜索技术,局部搜索能够快速找到局部极值.同时,拟随机序列的低偏差性保证了函数定义域能够被均匀地搜索,为找到多个局部极值包括全局极值提供了保证.

关键词: 全局优化; 不可微函数; 拟蒙特卡罗方法; 自适应搜索技术; 子群体

1 Introduction

In nondifferentiable optimization the Monte Carlo method of random search can be used to approximate the global optimum of a function (see Chap. 7 in [1]). Taking advantage of the low discrepancy of the quasi-random sequences, Niederreiter[2] introduced the quasi-Monte Carlo random search. The method is described as follows:

Problem Let f be a bounded continuous function defined on the bounded subset E of $\mathbb{R}^s (s \geqslant 1)$. The correct value M of the supremum of f over E to be found.

Let $x_1, \cdots, x_N$ be quasi-random numbers in E, then:

$$m_N = \max_{1 \leqslant n \leqslant N} f(x_n) \tag{1}$$

is taken as an approximation for M.

Theorem 1 Denote $d_N = d_N(E) = \sup_{x \in E} \min_{1 \leqslant n \leqslant N} d(x, x_n)$ be the dispersion of the quasi-random sequence $x_1, \cdots, x_N$ in E. Define $d(y, z) = \max_{1 \leqslant j \leqslant s} | y_j - z_j |$ for $y = (y_1, \cdots, y_s), z = (z_1, \cdots, z_s) \in \mathbb{R}^s$. $\omega(t) = \sup_{\substack{x, y \in E \\ d(x,y) \leqslant t}} | f(x) - f(y) |, t \geqslant 0$ is the modulus of continuity of f. We have $M - m_N \leqslant \omega(d_N)$.

Proof f is a bounded continuous function defined on E, then there should be one point x^* in E which satisfies $f(x^*) = M$. Choose x_k such that $d(x^*, x_k) = \min_{1 \leqslant n \leqslant N} d(x^*, x_n)$. We have

$$0 \leqslant M - m_N \leqslant f(x^*) - f(x_k) \leqslant \omega(d(x^*, x_k)) \leqslant \omega(d_N).$$

Because d_N is of an order of magnitude $O(N^{-1/s})$ and f is a bounded continuous function, the method described above is convergent. However, the rate of convergence is in general very slow. In order to speed up the convergence rate, Niederreiter and Peart[3] developed

* Foundation item: supported by the Special Funds for Major State Basic Research Projects (G19990328).
Received date: 2000 – 07 – 10; Revised date: 2002 – 01 – 06.

the quasi-random search by using "localization of search" (LQMC). In 1990, Wang and Fang[4] introduced a sequential number-theoretic method for optimization (SNTO).

But the effectiveness of LQMC and SNTO methods demand that N be sufficiently large to satisfy $N > \varepsilon^{-s}$, where $0 < \varepsilon < 1/2$ is the search step size in local search. In addition, if the function has many local maxima, in particular, the local maximum is much close to M, LQMC and SNTO methods could be led into a "wrong track", that is to say, the global maximum could not be found.

In this paper, we introduce an adaptive quasi-Monte Carlo method for global optimum (AQMC). AQMC has many advantages compared with LQMC and SNTO. Firstly, N is rather small when s is large. Secondly, adaptive search technique enables local search to head for local extrema quickly. The last but not least important advantage is that AQMC method is able to find various local extrema including global extremum. The algorithm is described in Section 2. Numerical experiments will be given in Section 3.

2 AQMC algorithm

Assume f be defined on rectangular region $E = [\boldsymbol{a}, \boldsymbol{b}], \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^s$. Borrowing ideas of population from genetic algorithms[5], we take the initial segment $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N \in E$ of infinite quasi-random sequence[6~10] as the first population, each point is an individual. Calculate fitness for each individual. Select one individual (selection probability is proportion to fitness p_{ij}), i.e., the kth individual and perform adaptive local search. The local search points corresponding to the kth individual are called the sub-population of the kth individual. Let $f\max_k$ denote the maximal function value of all individuals of sub-population of kth individual. When the search size ε_{ik} converges to zero, $f\max_k$ can be taken as an approximation for one local maximum and $f\max = \max\limits_{1 \leqslant k \leqslant N} f\max_k$ as that for the global maximum. New population with N individuals will be generated after all individuals of the previous population having clambered to neighboring local maxima. The initial segments for arbitrary N of infinite quasi-random sequences distribute in E uniformly and the new points generated from the rest of the quasi-random sequence fill in the gaps in the previously generated distribution in E, which ensures that the function field can be searched evenly and the global maximum can be found.

The AQMC algorithm can be described as follows:

1) Step 0 $\quad i = 1$.

2) Step 1 a) Generate the ith segment $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$ of sequence as the N individuals of the population, set $\boldsymbol{x}_{ij} = \boldsymbol{x}_j, \varepsilon_{ij} = \varepsilon_0 (0 < \varepsilon_0 < 1/2), j = 1, \cdots, N$.

b) Calculate $f\max_j, p_{ij}$.

3) Step 2

If (stop criterion satisfied): Program ends

Else: Select one individual $\boldsymbol{x}_{ik}$ according to p_{ij} and perform adaptive local search (LAQMC).

4) Step 3

If ($\varepsilon_{ij} \to 0$ for $1 \leqslant j \leqslant N$): Go to step 4

Else: Calculate p_{ij} and go to Step 2

5) Step 4 $\quad i = i + 1$, go to Step 1.

The stop criterion may be set according to various situations. For example, if $f\max$ has not been improved after several generations, we stop running the program. We can also set the total generation number in advance. Moreover, there are many applications that are to find optimal parameters, that is to say, the global extremum is known, and then we can control the error between $f\max$ and the global extremum M.

Now we focus on the adaptive local search method (LAQMC). For the selected individual $\boldsymbol{x}_{ik}$, we map the first N_i points of the segment $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$ to the neighborhood of $\boldsymbol{x}_{ik}$ by $g_C: E \to C$.

$$\begin{cases} 1 \leqslant N_i = [c_2 \times N \times \max\{\varepsilon_{ik}, c_1\}] \leqslant N, \\ 0 < c_1 \leqslant 1,\ 0 < c_2 \leqslant 1, \end{cases} \tag{2}$$

$$g_C(\boldsymbol{x}) = \boldsymbol{c} + \varepsilon_{ik}(2\boldsymbol{x} - (\boldsymbol{a} + \boldsymbol{b})),\ \boldsymbol{x} \in E, \tag{3}$$

where $[x]$ denotes the greatest integer $< x$. $\boldsymbol{c}$ is initialized to be $\boldsymbol{x}_{ik}$, if $f(g_C(x_j)) > f(\boldsymbol{c})$, then $\boldsymbol{c}$ is set to be $g_C(x_j), j = 1, \cdots, N_i$. As showed in flow chart (Fig. 1), $\varepsilon_{i+1,k}$, the next search step size of kth individual, will be adjusted according to this search result. If function value bigger than $f(\boldsymbol{x}_{ik})$ is found, then $\varepsilon_{i+1,k} = d(\boldsymbol{c}, \boldsymbol{x}_{ik})$, and $\boldsymbol{x}_{ik}$ will be replaced by $\boldsymbol{c}$. Otherwise, we have $\varepsilon_{i+1,k} = c_3 \times \varepsilon_{ik}$, where $0 < c_3 < \varepsilon_0$, we suggest that $c_3 = \varepsilon_0^3$.

It is evident that the adaptive local search is heading for local maximum points quickly because the search di-

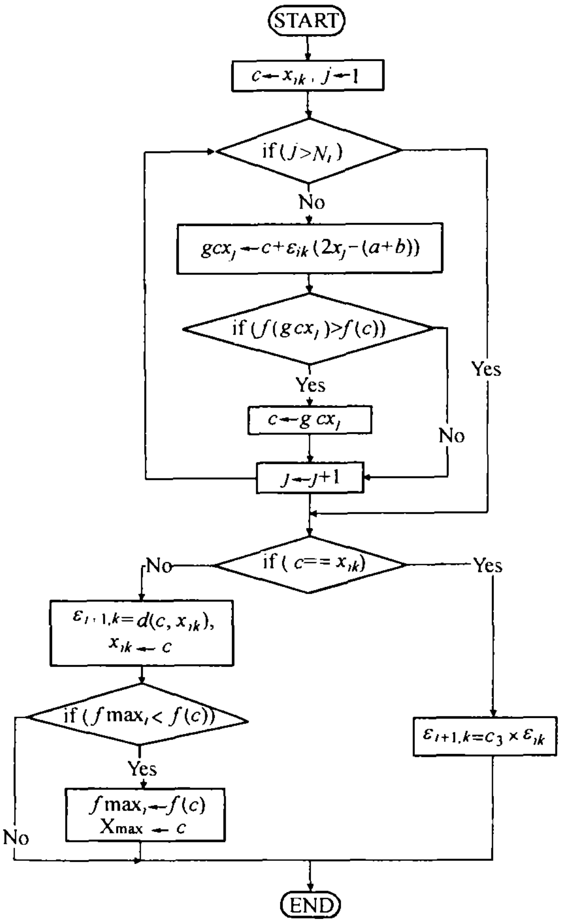

Fig. 1 Flow chart of local search of AQMC(LAQMC)

rection and search step size ε_{ik} are adjusted according to the previous search result. In addition, N_i is proportion to ε_{ik}. So the adaptive local search of AQMC algorithm does not cost much time.

3 Numerical experiments

We have carried out numerical experiments on some classical functions to compare the search result of AQMC with that of LQMC and SNTO. Sobol' sequences[11,12] were used in the experiments. Some examples are given as follows, in which f_1 is taken from [13] and f_2 from [14].

Example 1

$$f_1(x,y,z,u) = -\left(x-\frac{3}{11}\right)^2-\left(y-\frac{6}{13}\right)^2-\left(z-\frac{12}{23}\right)^2-\left(u-\frac{8}{37}\right)^2,$$

$(x,y,z,u)\in I^4$.

We have known that the global maximum of f_1 is 0. Table 3.5 in Wang and Fang[13] shows that the SNTO method will find the global maximum after calculating more than 2000 function values (the error of size $O(10^{-7})$). However, the same precision is attained only after calculating less than 400 function values for LAQMC method. Table 1 shows the result of LAQMC method. All symbols in the tables have been explained in Section 2.

Table 1 The results of LAQMC methods for $f_1(c_1 = 0.5,\ c_2 = 1.0, c_3 = 0.015625)$

N_i	$f$max	x	y	z	u
64	-0.0634333120	0.3281250000	0.6718750000	0.4531250000	0.1093750000
32	-0.0110488345	0.3125000000	0.4687500000	0.5937500000	0.2812500000
32	-0.0110488345	0.3125000000	0.4687500000	0.5937500000	0.2812500000
32	-0.0068469013	0.3077392578	0.4687500000	0.5778808594	0.2653808594
32	-0.0000484404	0.2779846191	0.4627990723	0.5243225098	0.2197570801
32	-0.0000484404	0.2779846191	0.4627990723	0.5243225098	0.2197570801
32	-0.0000050746	0.2747418284	0.4620668292	0.5225442052	0.2165142894
32	-0.0000021081	0.2729177587	0.4610534571	0.5227468796	0.2171223126
32	-0.0000001323	0.2728607565	0.4615664767	0.5215498338	0.2164952887
320					

The AQMC method is superior to LQMC method not only in the local search ability but also in the global search ability. For some multiple-extrema functions, LQMC method may be led into "wrong track", while AQMC method is able to find the various local extrema including the global extremum.

Example 2

$$f_2(\boldsymbol{x}) = -\sum_{i=1}^{n} x_i\sin(\sqrt{|x_i|}),\ -500 \leqslant x_i \leqslant 500.$$

The global minimum of f_2 is reached at $x_i = 420.9687, i = 1,\cdots,n$. The local minima are reached at $x_k \approx (\Pi(0.5+k))^2$, $k = 0,2,4,6$ and $x_k \approx -(\Pi(0.5+k))^2$, $k = 1,3,5$. Points $x_i = 420.9687$,

$i = 1, \cdots, n, i \neq j$, $x_j = -302.5232$ reach the second minimum, they are far away from the global minimum point. It has likely been led into "wrong track" through general search methods. Let $n = 2$, LQMC method stayed at the second minimum, while the AQMC method found the global minimum point $x_1 = 420.9690988064, x_2 = 420.96909888064$ after calculating 1328 function values (the error of size $O(10^{-7})$). The second minima and some other local minima close to the global minimum have also been found by the AQMC method.

The computational results show the following:

1) AQMC is a global optimization method while LQMC and SNTO may lead into "wrong track".

2) The local search of AQMC is about 5 times faster than LQMC and SNTO.

3) The population size of AQMC is rather less than the sample size of LQMC and SNTO.

4) AQMC method is able to find various local extrema.

4 Conclusion

As showed in Theorem 1 and the numerical experiments, adaptive quasi-Monte Carlo search method is a global search method. In particular, the adaptive technique in local search speeds up the search hugely. It is worth pointing out that the AQMC is able to find various local extrema, it is very useful in many applications of optimization. We suggest that the adaptive local search of AQMC be used for localization of search combined with GA(genetic algorithms). It can be safely expected that the hybrid algorithms will be promising.

References

[1] Rubinstein R Y. Simulation and the Monte Carlo Method [M]. New York: Wiley, 1981

[2] Niederreiter H. A quasi-Monte Carlo method for the approximate computation of the extreme values of a function [A]. In Studies in Pure Mathematics (To the Memory of Paul Turán) [M]. Basel: Birkhäuser Verlag, 1983, 523 – 529

[3] Niederreiter H and Peart P. Localization of search in quasi-Monte Carlo Methods for global optimization [J]. SIAM J. of Scientific and Statatistical Computing, 1986,7(2):660 – 664

[4] Wang Y and Fang Y T. Number theoretic methods in applied statistics [J]. Chinese Annals of Mathematics, Series B, 1990, 11(3):41 – 55, 859 – 914

[5] Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs [M]. Berlin: Springer-Verlag, 1994

[6] Halton J H. On the efficiency of certain quasi-random sequences of points in evaluation multi-dimensional integrals [J]. Numerische Mathematik, 1960,2(1):84 – 90

[7] Sobol' I M. The distribution of points in a cube and the approximate evaluation of integrals [J]. USSR Computational Mathematics and Mathematical Physics, 1967,7(4):86 – 112

[8] Faure H. Discrépance de suites associées à un système de numèration (en dimension s) [J]. Acta Arithmetica, 1982,41(4):337 – 351

[9] Niederreiter H. Point sets and sequences with small discrepancy [J]. Monatshefte fur Mathematik, 1987, 104(4):273 – 337

[10] Niederreiter H. Quasi-Monte Carlo methods and pseudo-random numbers [J]. Bulletin of the American Mathematical Society, 1978, 84(6):957 – 1041

[11] Press W H, Teukolsky S A, Vetterling W T, et al. Numerical Recipes in C: the Art of Scientific Computing [M]. 2nd ed. New York: Cambridge University Press, 1992, 309 – 313

[12] Bratley P and Fox B L. Algorithms 659 implementing sobol's quasirandom sequence generator [J]. Association for Computing Machinery Transactions on Mathematical Software, 1988, 14(1): 88 – 100

[13] Wang Y and Fang K T. Number-Theoretic Methods in Statistics [M]. London: Chapman and Hall, 1994,101 – 108

[14] Liu Y, Kang L S and Chen Y P. Non-Numerical Parallel Algorithm-Genetic Algorithm [M]. Beijing: Science Press, 1997, 115 – 116 (in Chinese)

本文作者简介

雷桂媛 现在浙江大学数学系计算数学专业攻读博士学位.主要研究方向为计算复杂性.前期工作为蒙特卡罗和拟蒙特卡罗方法在积分、优化和模拟中的应用研究. Email: guiyuanlei@elong.com