A Detecting Peak's Number Technique for Multimodal Function Optimization

Qiang Hua
Machine Learning Center
Faculty of Mathematics and Computer Science
Hebei University
Baoding 071002
CHINA
huaq@mail.hbu.edu.cn

Bin Wu
Machine Learning Center
Faculty of Mathematics and Computer Science
Hebei University
Baoding 071002
CHINA
wubinbb@gmail.com

Hao Tian
Department of Planning and Development
Handan Iron&Steel Co.,LTD.
Handan, 056015
CHINA

Abstract: - A Detecting Peak's Number (DPN) technique is proposed for multimodal optimization. In DPN technique, we want to know the peak's number of locally multimodal domain of every individual, firstly we use the idea of orthogonal intersection for getting the exploration direction in every locally multimodal domain, and then we attempt to detect peak's number in every one-dimension direction as the result of detecting of locally multimodal domain.

At last we design an evolution algorithm (DPNA) based on the characters of DNP technique, which contain four characters: niching, variable population, variable radius and life time, and then give a series of experiment results which show the effectiveness of algorithm, as the DPNA is not only adapting to obtaining multiple optima or suboptima, but also effective for problem of ill-scaled and locally multimodal domain described in [11].

Key-Words: - Evolution Algorithm; Multimodal Function Optimization; Niching; Detecting Peak's Number

1 Introduction

The general purpose of multimodal function optimization is to gain all optima/suboptima or obtain them as most as possible. When coping with this problem, we must face a serious of difficult, as follows, first multimodal function is continuous but maybe not smooth; second we do not how many modals there are in definition field in advance; third we do not where the modals is in definition field exactly; the last the heights of modals are not bounded to be equal, that is some is high, and some is low.

For the first difficult, some techniques based on Evolution Algorithm (EA) have been proposed for multimodal function optimization, such as fitness sharing [1-3], crowding[4-6], Sequential Niche[7], restricted competition selection[8-9], species conservation[10], neighbor selection[11]. Meanwhile, these techniques try to solve others problem as well, further, there are two problems dealt with in these techniques, that is how to preserve diversity in EA, and which individuals should be conserved in evolution process. Because in one hand it is a multimodal function and we want to get peaks as many as possible, how to maintain the diversity of population is a key problem, in which all kinds of niching method is used widely [1-10]; in another hand the height of peak is different, it become a key whether a individual with a low fitness is substituted by a individual with a high fitness, some methods were proposed [10-11].

Besides, there is another key problem about niching; generally we only get a best individual in a domain region, but we do not know how much the fittest radius is, if overlarge, the value of low peak is ignore; if undersize, the cost of space search becomes large. So it is difficult to deal with multimodal function optimization problems in ill-scale situation. Shin Ando and Shigenobu Kobayashi [11] gives a method to deal with ill-scale problem, here we will give another method.

A Detecting Peak's Number (DNP) technique is proposed aimed at kinds of problems of multimodal function optimization, which has four characters as follows:

DNP characters

(1) niching; (2) variable population; (3) variable radius; (4) life time.

The reasons why DNP contains four characters above are: firstly we try to find all suboptima and the suboptima which always locate in some region, so that any individual have itself domain region is a good idea, that is niching; secondly the peak's number can not be known in advance and every individual control itself domain region, so it is necessary to get the mount of individuals change according to the possible number of the peak; thirdly there are some relation between variable population and variable radius, and the another reason is the peak's location cannot be known in advance, so it is necessary to change the individual radius; lastly we introduce life time to try to avoid the effect of randomness of detecting procedure.

This paper is organized as follows: section 2, propose the detecting peak's number technique; section 3, gives a evolutionary algorithm based on DPN technique, called DPNA; section 4, provides some experimental results; section 5, gives a summery and conclusion;

2 DPN technique

Presume there is an individual A with the dominating radius R_a, now we want to judge whether there is a peak in A domain region or more than one peak or not. Firstly generate some orthogonal directions in domain region of individual A; secondly make detection to every direction, when detecting there exist more than one peak, we will make a adjustment to generate two new individuals with a less radius than individual A; thirdly when all direction are detected, if there exist more than one peak in the region of individual A, individual A and the two new individuals generated by individual A come into the next population; if there exist one peak at least in the region of individual A, the individual A come into the next population; if there is probably no peak in the region of individual A and the individual A is still in life time, it will come into the next population. The detail procedure is as follows:

2.1 Generate n orthogonal direction

The method of structuring the random orthogonal direction is described as follows: (presume we have known the initial direction e_1)

(1) If we have known one direction e_1, because of the relation

$$e_{11} \cdot e_{21} + e_{12} \cdot e_{22} + \ldots + e_{1n} \cdot e_{2n} = 0$$

There is $n-1$ degree of freedom, we need generate $n-1$ random number as the coordinate of e_2.

(2) If we have known k<n directions $e_1, e_2, \ldots, e_k$, because of the relation

$$e_{i1} \cdot e_{(k+1)1} + e_{i2} \cdot e_{(k+1)2} + \ldots + e_{in} \cdot e_{(k+1)n} = 0 \quad i = 1, \ldots, k$$

There is $n-k$ degree of freedom, we need generate $n-k$ random number as the coordinate of $e_{(k+1)}$.

(3) Since the principle of inductive method, we can structure n direction using the method above.

The reason of using orthogonal direction is that the new directions could differentiate the former direction at most, and we have to point out that it is not necessary to use every direction, when the dimension of solution spaces is large, we can only use some directions to perform the algorithm.

2.2 Detecting on one-dimension direction

We get some points in order on one-dimension direction, compare the first point and the last point and label the smaller point as 1, other points as 2,3,…,n-1 and the bigger point as n, then we sort these points in ascend order, and get a sequence at last by which we judge how many the peak's number is. When there are peaks, we only consider some situation as Fig 2, Fig 3 and Fig 4.

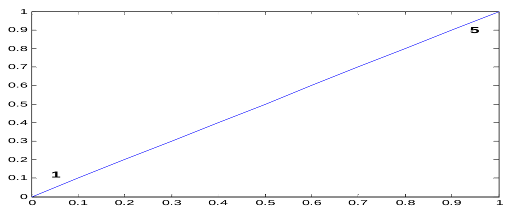

Fig. 1

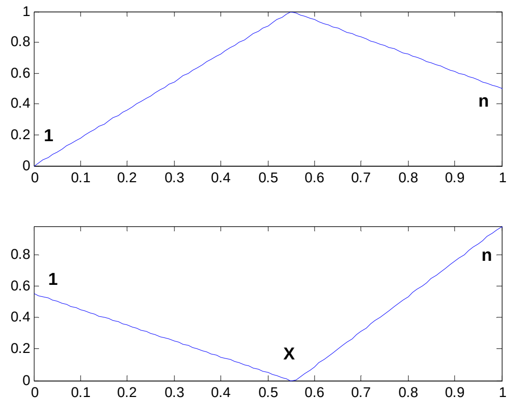

Fig. 2

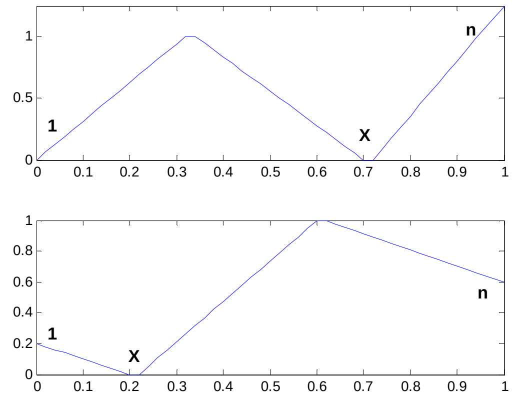

Fig. 3

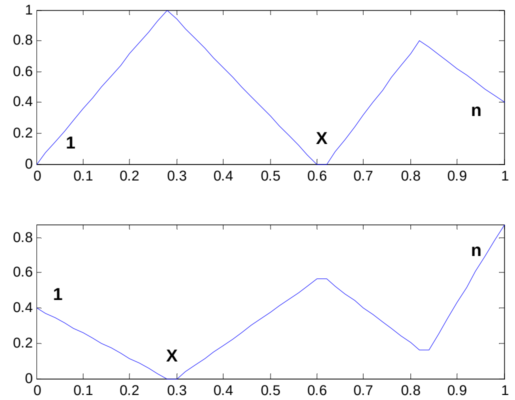

Fig. 4

We consider these pictures as situation of more than one peak except Fig 1 and the first picture in Fig 2, because there should be a peak on the left and right of point x at least.

So the all situations we will consider are no peak, at least one peak, and more than one peak. We take n=5 for example to give the algorithm. According to all kinds of different cases we can get a rule to judge the peak's number by the sequence, as follows:

If the first individual is not 1, there are exist more than one peak. Not lose generally, presume the individual is t (t is not 1 and 5). Because individual t is in the first and individual 5 is better than individual 1, there is a peak at least between individual 1 and t, the same between individual t and 5.

If the first individual is 1 and the second individual is not 2, there are two situations. If the second individual is 5, there is one peak at least. If the second is not 5, there is more than one peak and the reason is similar with (1).

If the first two individuals is 1, 2 and the third individual is not 3, there are two situations. If the third individual is 5, there is one peak at least. If the third is not 5, there is more than one peak and the reason is similar with (1).

If the first three individual is 1, 2, 3 and the fourth individual is not 4, the fourth individual must be 5, there is one peak at least.

If the sequence is 12345, there is probably no peak.

According the rule above we get a simple rule just judge whether there is peak or not, that is only if the sequence is not 12345, there is one peak at least in the region of individual A. These rules above still are used in programmer, since these rules with adjustment will change the radius of individual and change the size of population.

2.3 Adjustment

When there exist more than one peak, we will generate two new individual from individual A, according to the condition there must be an individual t worse than individuals adjacent it, so the center of new individuals are B (center(1)+center(t))/2 and C (center(t)+center(5))/2, and the radiuses are distance(1,t)/2 and distance(t,5)/2, and we will reserve the individual A with a new radius min(distance(A, B), distance(A, C)), because individual A maybe is better individual, so this method we could void lose excellent individual.

2.4 Reduction of population

After all direction are detected, we make reduction below

(1) If there exist more than one peak in the region of individual A, individual A and all new individual generated in 'section 2.3 adjustment' come into the next population, peak(A)=1, peak(each_ new)=1;

(2) If there is one peak at least in the region of individual A, individual come into the next population, peak(A)=1;

(3) If there is probably no peak in the region of individual A and peak(A)>-5, individual A come into the next population, peak(A)=peak(A)-1;

Here 'peak(individual)' is an individual parameter, and it is used to control life time, if the peak(A)=-5, means we did not find peak in consecutive five times, so we will delete the individual, and the -5 is adjustable as you will.

3 Algorithm design based on DPN

Evolution algorithm need be designed for applying DPN technique. When talk about evolution, generally speaking new excellent individuals displace old bad individuals, however, it is not all, that is new individuals should be tend to explore unknown field in above procedure. We give the evolution algorithm combined DPN based on such evolution thinking. What should be paid attention on is when a individual will displace a other individual, or when we want to delete a individual, that is, on one hand, how should we avoid jumping local peak region, when a better individual displace a bad individual; on the other hand, because of the introduction of DPN technique, how should we solve such problems as more and more individuals being generated and overlapping individuals.

Aim at all problems above, we give a description about evolution algorithm combined DNP technique (DPNA) as follows:

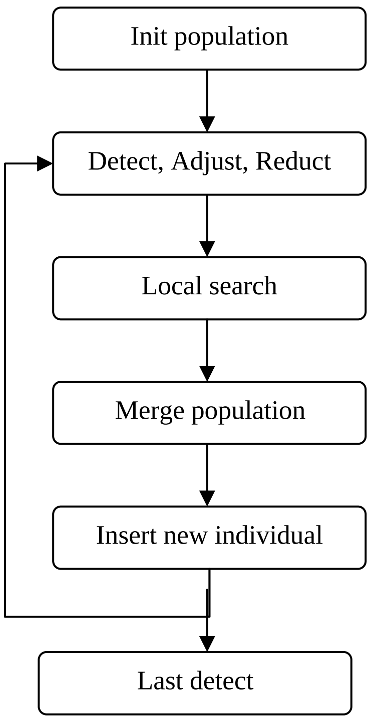

Fig. 5 Algorithm procedure

(**DPNA-1**)Init population

We make every individual domain region contain only one individual according the sequence of individual fitness, that is, the better individual get the domain region earlier.

This method could make individual with better fitness not affected by large domain region of individual with bad fitness.

(**DPNA-2**)Detect and adjust

We have described it in section 2 of passage.

(**DPNA-3**)Local search

We search some new individuals in the domain region of old individual, if there is a peak at least in the domain region of old individual, we will instead old individual with the best new individual if and only if there is a modal in the domain region of the best new individual and the best new individual is better then old individual; else if there is not a peak in the domain region of old individual, we will instead old individual with new individual if the best new individual is better then old individual.

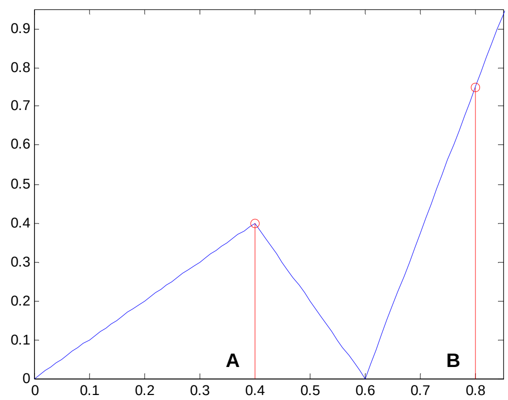

Fig.6

The reason that we adopt this method is: Fig.6 is a possible situation and we want to avoid it, when individual A is searching locally in its domain region, individual B could be generated randomly, the fitness of individual B is better than A, but there is no peak in domain region of individual B, so we give the method above to avoiding this situation. The method could avoid individual's jumping out little peak.

(**DPNA-4**)Merge population

It is similar with (DPNA-1) but not all. That is firstly we reduce the radius of individual overlapping the new generated individual, which are generated from (DPNA-2), because we want the new individual and domain region not to overlap by another individual; secondly we execute (DPNA-1).

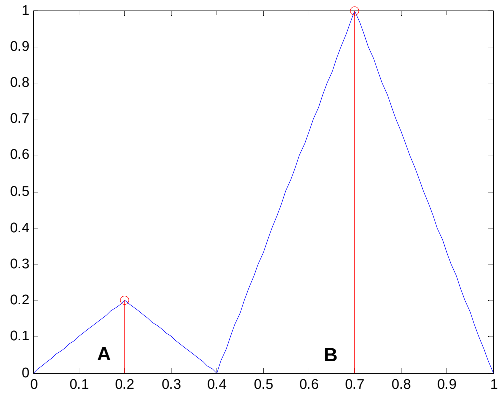

Fig.7

The reason that we adopt this method is: Fig.7 is a possible situation and we want to avoid it, after step (DPNA-3) individual A with a peak in its domain region may come into the domain region of individual B, but the fitness of individual A is lower than individual B, so we give the method above to avoiding this situation. This method could keep new individual with bad fitness alive, which could in a region with peak.

(**DPNA-5**)Insert new individual

We try to insert some new points, which are not in the domain region of old individual and let the radiuses of these new points are the mean of all old individual radiuses. Meanwhile we use a threshold to assure the radius of new added individual not too small.

(**DPNA-6**)Last detect

The 'last detect' is similar with '(2)detect and adjust' but some differences, here is more strict, that is, we use radius/10 for detecting peak and assure no better individual around the detected individual. After this step, we get the individuals which are on the peak of domain region and discard the other individuals.

4 Experimental results

The proposed algorithm was applied to some multimodal mathematical functions.

(1)Six-peaks function (Fig.7)

$$y=\left(4-2.1x_1^2+\frac{1}{3}x_1^4\right)x_1^2+x_1x_2+\left(-4+4x_2^2\right)x_2^2 \quad (1)$$

Six-peak function has six peaks with the range $x_1 \in [-3\ 3]$, $x_2 \in [-2\ 2]$. We setup the initialize individual radius with 3, the limited life time with 5, the inserted new individual with 10 and the max iterative with 100. We want to emphasize that according to the setting, we know the initialize individual is less about 2 individuals only, but we can find the peak in search range still, which show the effectiveness of DPNA. Fig.8 is an experimental solution and we make the experiment 100 times, the experimental solution show the times we could find the number of peak, show in table 4.1.

Table 4.1

6 peaks	5 peaks	4 peaks	3 peaks	2 peaks	1 peak
53	27	16	4	0	0

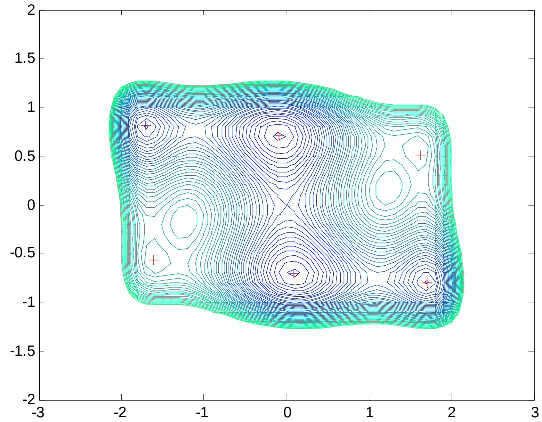

Fig.8 a solution of six-peaks function

(2)Shubert function

$$y=\left(\sum_{i=1}^{5} i\cos\left((i+1)x_1+i\right)\right) * \left(\sum_{i=1}^{5} i\cos\left((i+1)x_2+i\right)\right) \quad (2)$$

Shubert Function is often used as locally-multi modal benchmark. Within the range $(-4 \le x_i \le 8)$, it has over 100 suboptima distributed symmetrically. There are 2^n pairs of optima, each pair within a cluster of 4^n suboptima. Fig.9 shows its contour by light lines [1].

We setup the initialize individual radius with 2, the limited life time with 5, the inserted new individual with 10 and the max iterative with 50. Fig.9 is an experimental solution and we have found great majority peak.

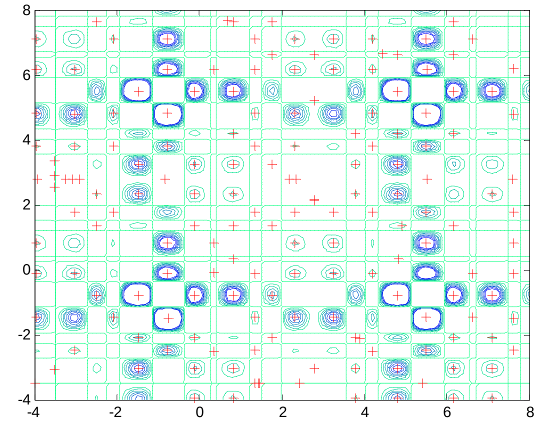

Fig.9 Experimental solution of Shubert function

(3) Function derived from Rosenbrock's function [1]

$$F_1(x_1,x_2)=100\left(x_1-x_2^2\right)^2+\left(x_2-1\right)^2$$
$$F_2(x_1,x_2)=F(x_1,x_2)_1\times100\left(x_1-\left(x_2-4\right)^2\right)^2 \quad (3)$$

Formula (3) has a parabolic dominant attractor and a weaker attractor placed parallel to each other, and it is an ill-scale function.

We setup the initialize individual radius with 2, the limited life time with 5, the inserted new individual with 10 and the max iterative with 50. Fig.10 is an experimental solution and we have found great majority peak.

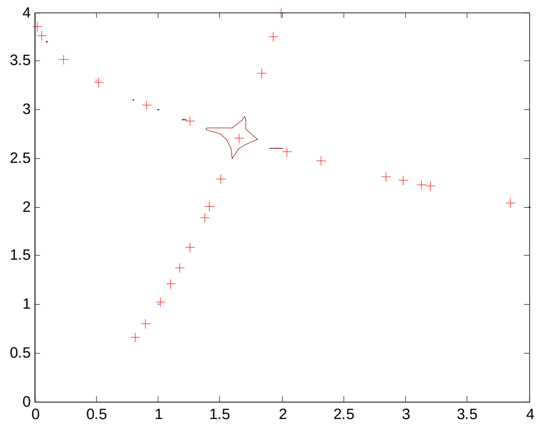

Fig.10 Experimental solution of Rosenbrock2 function

All experimental solutions above show the effectiveness of the new proposed algorithm.

5 Conclusions

The individual number and individual radius of new proposed algorithm (DPNA) is variable, so the DPNA is not only adapting to obtaining multiple optima or suboptima, but also effective for problem of ill-scaled and locally multimodal domain described in [1].

6 Acknowledgements

This paper is supported by the Machine Learning Centre of the Hebei University, sponsored by young foundation of Hebei University and the natural science foundation of Hebei province in China (F2007000221).

References:

[1] D. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In Proceedings of the 2nd International Conference on Genetic Algorithms, pages 41-49, 1987.

[2] B. L. Miller and M. J. Shaw, "Genetic algorithms with dynamic niche sharing for multimodal function optimization," in Proc. IEEE Int. Conf. Evolutionary Computation, 1996, pp. 786–791.

[3] B. Sareni and L. Krähenbühl, "Fitness sharing and niching methods revisited," IEEE Trans. Evol. Comput., vol. 2, pp. 97–106, Sept. 1998

[4] K. De Jong. An analysis of the behavior of a class of genetic algorithms. (Doctoral dissertation, University of Michigan) Dissertation Abstracts International, 36(10), 5140B. (University Micro_lms No. 76-9381), 1975.

[5] S. W. Mahfoud. Crowding and preselection revisited. In R. M anner and B. Manderick, editors, Parallel Problem Solving from Nature II, pages 27-36, Amsterdam, 1992. North-Holland.

[6] Mengshoel, O. J. and Goldberg, D. E. (1999). Probability crowding: deterministic crowding with probabilistic replacement. In Banzhaf, W., Daida, J., and Eiben, A. E., editors, Proceedings of the Genetic and Evolutionary Computation Conference 1999, pages 409–416, Morgan Kaufmann, San Francisco, California.

[7] David Beasley, David R.Bull, Ralph R.Martin, A Sequential Niche Technique for Multimo dal Function Optimization, Evolutionary Computation 1(2):101--125, 1993.

[8] C.-G. Lee, D.-H. Cho, and H.-K. Jung, "Niching genetic algorithm with restricted competition selection for multimodal function optimization," IEEE Trans. Magn., vol. 34, pp. 1722–1725, May 1999.

[9] J.-K. Kim, D.-H. Cho, H.-K. Jung, and C.-G. Lee, "Niching genetic algorithm adopting restricted competition selection combined with

pattern search method," IEEE Trans. Magn., vol. 38, pp. 1001–1004, Mar. 2002.

[10] Jian-Ping Li, Marton E. Balazs, Geoffrey T. Parks and P. John Clarkson. A Species Conserving Genetic Algorithm for Multimodal Function Optimization. Evolutionary Computation 10(3): 207-234, 2002.

[11] Shin Ando and Shigenobu Kobayashi. Fitness-based Neighbor Selection for Multimodal Function Optimization. Proceedings of Genetic and Evolutionary Computation Conference 2005 (GECCO'05), page1573–1574, 2005.

[12] Chang-Hwan Im, Hong-Kyu Kim and Hyun-Kyo Jung. A Novel Algorithm for Multimodal Function Optimization Based on Evolution Strategy. IEEE TRANSACTIONS ON MAGNETICS, VOL. 40, NO. 2, MARCH 2004.