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

Bin Wu
Machine Learning Center
Faculty of Mathematics and Computer Science
Hebei University
Baoding 071002
CHINA

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

Abstract: - We propose a Detecting Peak's Number (DPN) technique for multimodal optimization. In DPN, we use the idea of orthogonal intersection 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 mainly are variable population and variable radius, and give a series of experiments 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 [1].

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

1 Introduction

Some practical optimization problems, for which one searches for a set of minima instead of only one, are often dealt with. So how to obtain all optima/suboptima of multimodal function have been widely studied. Up to now, there are lots of techniques to solving the problem above, such as crowding population, niche, clustering, species saving and so on.

Simple GA is not adapted to this problem, because elite strategy makes the population tend to converge to a single solution. That is, to multimodal function problem population diversity should be placed importance on. When we use select operator to pick out new individual, we should give bad individuals a change to life sometimes, for they are probably important to search.

Paper [2] propose species conserving algorithm to solve the problems described above by partitioning the population $P(t)$ into a set of dominated species X_s and copying the dominating individual of each of these species into the next generation. The algorithm builds the set X_s by successively considering each of the individuals in $P(t)$, in decreasing order of fitness. When an individual is considered, it is checked against the species seeds found so far. If X_s does not contain any seed that is closer than half the species distance $r/2$ to the individual considered, then the individual will be added to X_s.

But this algorithm is based on the invariable radius, when encounter a local domain problem, show as Fig.1 right, we must adjust the radius to very small to satisfy the problem, but in fact we don't know the size of radius in advance. Lots of algorithm [3] encounter this difficult.

So we should know this problem "Which individuals should be conserved". The answer is conserving the individuals belonging to different peak. Paper [1] gives a solving method; here we give another feasible method to solve this problem.

We propose a Detecting Peak's Number (DPN) technique. The technique is based on the technique of niching, that is every individual i have a domain region whose radius is r_i, and based on the concept of variable population and variable radius, that is when finding more than one peak in individual domain region, we will generate new individual and its radius. We make detecting about every individual in his domain region, and conserve the individual there existing a peak in them domain region, meanwhile we introduce the limited life time strategy, that is only when the peak of a individual is not detected in consecutive some times, we will delete it.

This paper is organized as follows: section 2, propose the detecting peak's number technique and analyse the DPN technique; section 3, gives a evolutionary algorithm based on DPN technique, called DPNA; section 4, provides some experimental solutions; section 5, gives a summery and conclusion; section6, describe further work.

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 not. Firstly generate some orthogonal directions; secondly make the 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 child population; if there exist one peak at least in the region of individual A, the individual A come into the child 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 child 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

Presume there are five individuals and the individual is 1, 2, 3, 4, and 5 respectively from left to right and the individual 5 is better than individual 1 in term of the function value. According to function value, we can sort the individual in accordance with ascend sequence, from Fig.1 we know the ascend sequence is 12345, but we can not sure there exist one peak in terms of the sequence; form Fig.2 we know the ascend sequence is 15243, but we can not sure there exist just one peak in terms of the sequence; form Fig.3 we know the ascend sequence is 31452, and we can sure there are must exist more than one peak in term of the sequence (the reason is below).

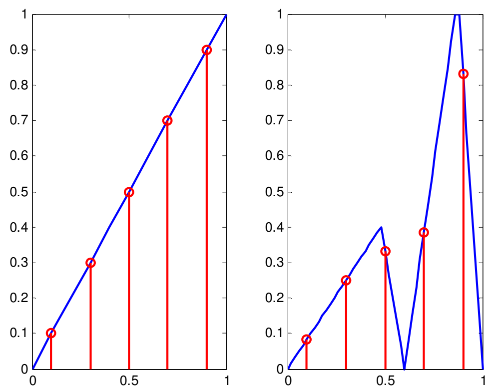

Fig.1 12345

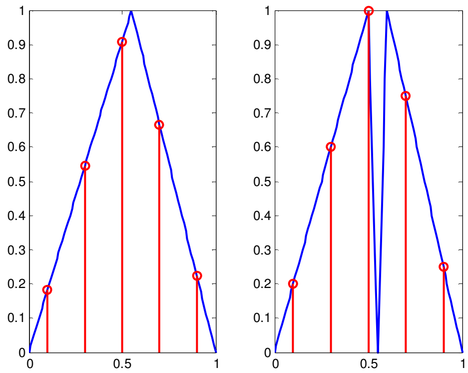

Fig.2 15243

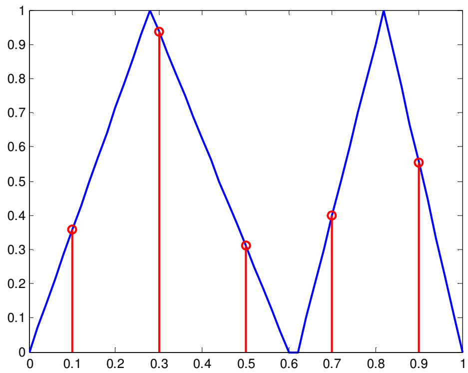

Fig.3 31452

In any case we can get a rule to judge the peak's number in term of 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 child 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 child 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 child 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.

2.5 Analysis of DNP technique

The main difficulty for multimodal function optimization is that the peak's number and the peak's location are not known in advance easily, So DNP technique is proposed aimed at the multimodal function optimization, which haves four characters as follows:

DNP characters

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

The reason being in possession of the four characters above are: firstly we try to find all suboptima and the suboptima always locate in some region, so that any individual have itself domain region is a good idea, which is niching; secondly the peak's number can not be known in advance and the every individual control itself domain region, so it is necessary to let 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 can not 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.

3 Algorithm design based on DPN

Algorithm (DPNA) design force on how to change population and how to change the radius, and the latter is more important.

We give a description about algorithm (DPNA) as follows:

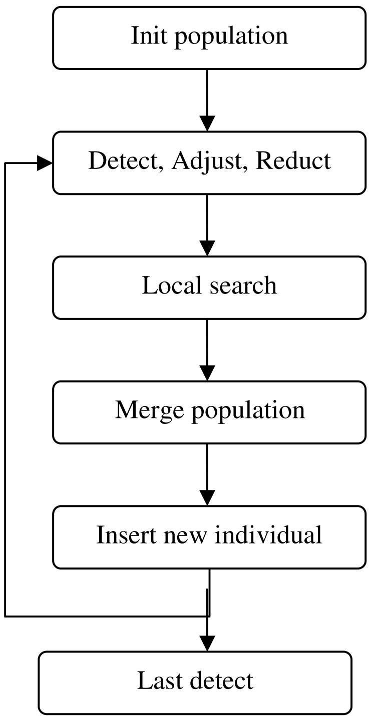

Fig. 4 Algorithm procedure

(**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.

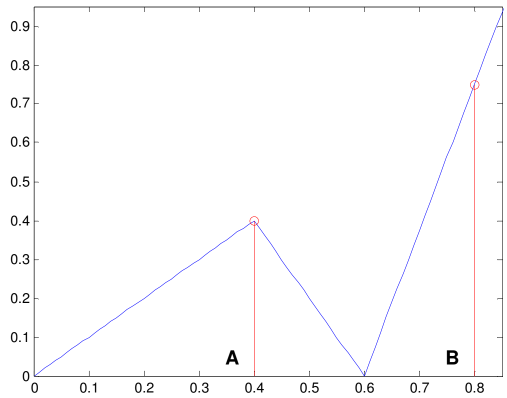

Fig.5

The reason that we adopt this method is: Fig.5 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).

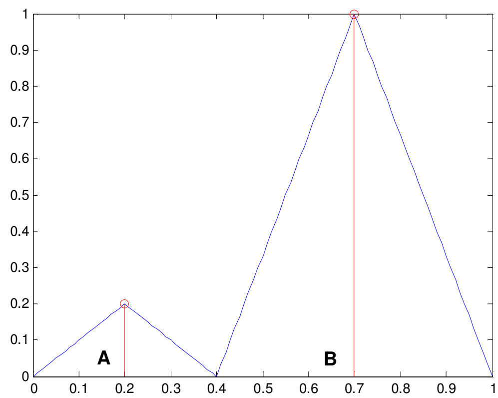

Fig.6

The reason that we adopt this method is: Fig.6 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 solutions

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

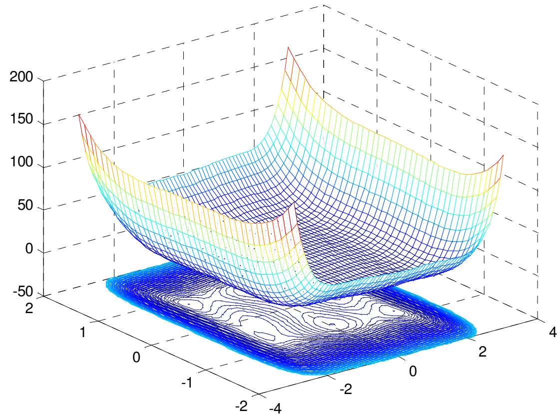

Fig.7 six-peaks function

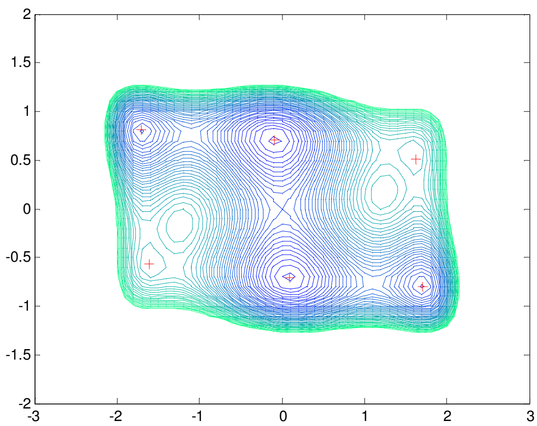

Fig.8 a solution of six-peaks function

(2)Shubert function

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

Shubert Function is often used as locally-multi modal benchmark. Within the range $\left(-4\le x_i\le 8\right)$, 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.

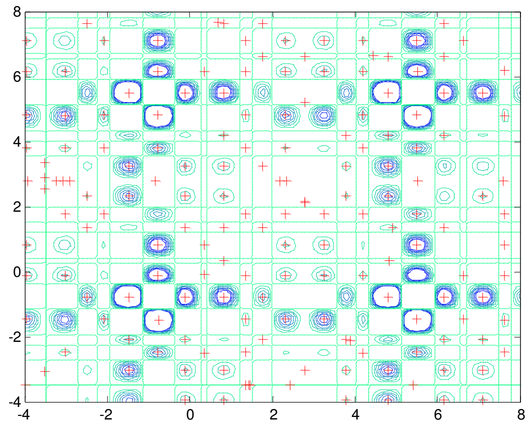

Fig.9 Experimental solution of Shubert function

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

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

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.

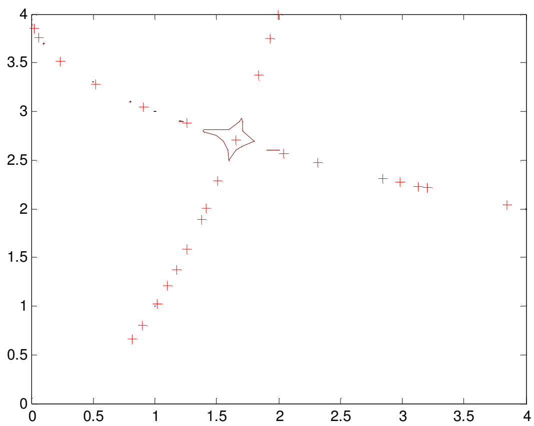

Fig.10 Experimental solution of Rosenbrock2 function

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

5 Conclusion

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 Further work

We try to further analysis and perfect our algorithm (DPNA) based on Detecting Peak's Number (DPN) technique.

7 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]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.

[2]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.

[3]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.

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