A Niching Cumulative Genetic Algorithm with Evaluated Probability for Multimodal Optimization

Hongjun Yang, Yixu Song, Lihua Wang, Peifa Jia State Key Laboratory on Intelligent Technology and Systems Tsinghua National Laboratory for Information Science and Technology Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R. China yanghj_72@126.com, songyixu@163.com jinziye@sina.com.cn, dcsjpf@mail.tsinghua.edu.cn

ABSTRACT

Multi-modal problems with multiple local/global optima are ubiquitous in real-world application. Many multi-modal optimization algorithms have been developed to search as many local/global optima as possible. However, to locate and maintain many optima simultaneously, both the search quality and efficiency of these algorithms may be influenced. Here, we propose a new niching genetic algorithm that attempts to improve both the search quality and efficiency. To the end, we incorporate two mechanisms into the algorithm: cumulative population technique and an evaluated probability of new individuals. The first mechanism is designed to keep the found solutions by storing all known information, whilst the second is responsible for exploiting unexplored space effectively by guiding the exploration process. The proposed approach is compared with five different niche genetic algorithms on six well known multimodal functions of different characteristics. Empirical results indicate that the proposed approach outperforms other algorithms. It not only increases the probability of finding both global and local optima, but also reduces the average number of function evaluations.

Categories and Subject Descriptors

G.1.6 [Numerical Analysis]: Optimization; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods and Search

General Terms

Algorithms

Keywords

Niching Genetic Algorithms; Fitness Evaluation; Multimodal Optimization; Cumulative population; Evaluated Probability

1. INTRODUCTION

The problems of interest in scientific and engineering field are or can be advantageously formulated as optimization problems. Due to physical (and/or cost) constraints, these problems may contain several high quality global or local solutions which have to be identified and the most appropriate solution should be chosen [1]. These problems are known as multimodal optimization problems.

GECCO'14, July 12-16, 2014, Vancouver, BC, Canada.

Copyright © 2014 ACM 978-1-4503-2881-4/14/07...\$15.00. http://dx.doi.org/10.1145/2598394.2605677. Many Evolutionary Algorithms(EAs) introducing niching technique [2-5] have recently been proposed and attempt to solve the multimodal optimization problem. Niching technique can maintain diversity of population and converge in parallel to multiple solutions by restrictive reproduction. Various niching techniques were proposed in the past few decades, such as crowding [6, 7], fitness sharing [8, 9], clearing [10], clustering [11], Multi-Niche Crowding [12], etc.

However, these niching algorithms need take more computing efforts to locate and maintain many optima. Therefore, many algorithm [4, 13] incorporating local search technique, known as memetic algorithms, greatly improve the ability of evolutionary algorithms to accurately locate optimal solutions. But, these approaches usually need a large number of fitness evaluations to find high quality solutions, which might not always be an effective approach especially for problem with very expensive function evaluation. To reduce the number of fitness evaluation, M. Hall [14] proposed that new individual should be evaluated according to its cumulative local information. But, it needs to set a distance threshold which can depends on prior knowledge about optimization problem. Improper setting may make the algorithm get into the repeated check.

Motivated by these observations, a new niching genetic algorithm is developed to improve the search quality and efficiency. It introduces two mechanisms: cumulative population technique and an evaluated probability of new individual. The first mechanism can keep the found solutions by storing all known information, whilst the second is responsible for exploiting unexplored space effectively.

The remainder of the paper is organized as follows. Some related works are briefly described in Section 2. Implementing details of the proposed algorithm is presented in Section 3. In section 4, we evaluate the proposed algorithm and discuss its outcomes based on experiment results. The paper ends in section 5 with a brief discussion.

2. RELATED WORK

2.1 Crowding

Crowding was proposed by De Jong to preserve population diversity and prevent premature convergence. Crowding is applied in the replacement step of GAs in order to decide which individuals and their offspring will remain in the population according to a similarity metric which is based on the number of matching alleles. Depending on the replacement strategies, crowding approach is classified as Deterministic crowing [6] and Probabilistic crowding [7]. Deterministic Crowding selects the fittest individual in the replacement step. Probabilistic Crowding

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

selects the surviving individual based on a probabilistic formula that takes fitness into account.

The Multi-Niche Crowding, propose by Cede [12], differs from the above crowding approaches by implementing the crowding concept in the selection rules and replacement rules. In crowding selection, each individual in the population has the same chance for mating in every generation. First, an individual is selected for mating sequentially or randomly. Secondly, other individual is selected from a group of individuals of size , picked at random from the population. The other individual chosen must be the one who is the most "similar" to the first individual. In the replacement rules, MNC uses a replacement policy called worst among the most similar. Its goal is to pick replacement individual which is the worst in similar individuals to offspring. Its disadvantage need also set some parameters by prior knowledge.

2.2 Fitness Sharing

Fitness sharing, proposed by Goldberg and Richardson, alter the fitness of each individual according to its neighboring individuals. It is based on the penalization of the areas of the searching space with more solutions within the population. The detail can be found in [8]. Fitness sharing maintains the diversity of population by adjusting the fitness of the individuals according to sharing function. The shortcoming of this approach is that sharing function and the niche radius, which are related to optimization problem, need be set firstly. Clearing proposed by Petro ski [10] is a special fitness sharing method. By comparing all individuals in the niche radius and setting the capacity of each niche, several best individuals survive and the others are all cleared. Dynamic niche sharing also belongs to fitness sharing method in [9]. It identifies niche by niche radius and the number of peaks to identify. At the same time, the disadvantage of clearing and dynamic niche sharing is similar to that of fitness sharing.

2.3 Dynamic Niche Clustering

Dynamic niche clustering [11] proposed by Gan and Warwick combines clustering and fitness sharing. For each generation, a clustering operation is executed with a dynamically updated cluster number. Next, fitness sharing is implemented with the niche radius provided by clustering. But it is not an easy task to calculate the cluster number, which will affect the quantity and quality of identified optimal solutions greatly.

3. Niching Cumulative Genetic Algorithm with Evaluated Probability (EPNCGA)

Based on niching genetic algorithms(NGAs), the EPNCGA use cumulative population and evaluated probability to implement multimodal function optimization. We know that all information may help us solve the problem, the cumulative population can prevent the loss of useful information or redundant fitness evaluation by storing all evaluated information; on the other hand, we can also determine if new individual is evaluated by evaluated probability to reduce the number of fitness function evaluation. The algorithmic flow of the proposed algorithm can be depicted in Algorithm 1. After the initial population is created, every individual is evaluated by special problem. Next, the proposed niche identification method (NIM) is implemented to find all interesting area of peak. Selection, crossover, and mutation are then performed. Finally, evaluated probability of new individual is calculated to determine if new individual from crossover and mutation is evaluated and add to population. The evolutionary process is repeated until the stopping criterion is satisfied. The detail steps are described in the algorithm 1.

Algorithm 1. Algorithmic Flow of EPNCGA

1.	<pre>pop=GenerateInitialPopulation(pop_size);</pre>
2.	Evaluate(pop);
3.	while the stopping criterion not met do
3.1	niches=NIM(pop);
3.2	<pre>crossoveroffspring = Select(niches,pop);</pre>
3.3	Crossover(crossoveroffspring);
3.4	Evaluate_and_Add(crossoveroffspring);
3.5	repeat 3.2-3.4 until add Nc individuals
3.6	<pre>mutationoffspring=randselect(pop);</pre>
3.7	Mutate(mutationoffspring);
3.8	Evaluate_and_Add(mutationoffspring);
3.9	repeat 3.6-3.8 until add Nm individuals
3.10	end while

3.1 Niche Identification Method

Niching technique is an effective approach which can locate and maintain many solutions. A niche can be defined as a subspace in the search space which includes a global or local optima. Therefore, accurate identification of niches is the key to improving the accuracy and efficiency of these solutions. Standard fitness sharing requires a unified sharing radius, but it doesn't suit for the multimodal problems where niches have different shape and range. Dynamic niche cluster methods [11] can determine clusters of different size, but two further parameters need be set firstly. Lin et al. [15] proposed a niche identification method which can determine the center location and radius of each of existing niches based on fitness topographical information of individuals. It also set one parameter by prior knowledge. Hillclimbing function [13] was used to identify niche, but it need to evaluate the positions between two individuals. By modifying the method proposed by [15], a new niche identification method is proposed.

Empirically, if two closer niches A and B satisfy the following condition, we should merge these two niches. This condition expresses that all individuals of niche B far away from x_A than x_B .

Condition 1: $f(x_A) \ge f(x_B)$ and $\forall x \in B$, $d(x_A, x_B) \le d(x_A, x)$

Here, for convenience, x_N denotes the optimum individual in a niche N, which also is referred as niche center; d(x, y) denote the Euler distance between two individuals x and y; f(x) denotes the fitness of the individual x.

Using condition 1, the complete niche identification procedure can then be defined as follows.

1. The individual x with the highest fitness in the population is selected as the center x_C of the first niche C (C=1), and this individual is then marked.

2. Calculate the distances between x_c and all other individuals, and construct an individual's sequence $(x^1, x^2, x^3...)$ such that $d(x^1, x_c) \le d(x^2, x_c) \le d(x^3, x_c) \le ...)$.

3. Sequentially check the fitness $f(x^i)$ and $f(x^{i+1})$ of the adjacent two individuals x^i and x^{i+1} until $f(x^{i+1}) > f(x^i)$. Mark all individuals $x^1, x^2, x^3, \dots, x^i$ in niche *C*.

4. The individual with the highest fitness in all unmarked individuals is selected as the center of new niche C(C=C+1) and repeat process 2) and 3) until all individuals are marked.

5) Check all niches the algorithm will merge two closer niches which satisfy condition 1.

3.2 Evaluated probability of new individual

Each new individual should help improve the accuracy of solutions or maintain diversity of population. Empirically, if the density of evaluated individuals in one niche meets the normal distribution rather than uniform distribution, or, the density nearby niching center is higher, search quality and efficiency of the algorithm may be greatly improved. Inspired by these, we design an evaluated probability function of new individual.

Suppose optimization problem meets continuity, new individual's feature is similar to the nearest. At *n* generation, we can calculate evaluated probability p(x, y, n, N) of new individual *x* by the nearest *y* in niche *N*. p(x, y, n, N) relates the distance d(x, y) and the fitness difference *f* between *y* with x_N .

$$p(x, y, n, N) = \begin{cases} \frac{1}{2} + \left[\frac{\alpha(n)d(x, y) - f}{1 - f}\right]^{-7} / 2, & \alpha(n)d(x, x_n) \ge f \\ \frac{1}{2} - \left[\frac{f - \alpha(n)d(x, y)}{f}\right]^{-7} / 2, & otherwise \end{cases}$$
(1)

Here, the difference f can be denoted as $f = \frac{f(x_N) - f(x_n)}{f(x_N) - f_{\min}}$, f_{\min}

is the minimum fitness in niche N, $\alpha(n) = \varphi(1/\rho)$ is increasing function, ρ denotes the average density of individuals in the search space.

The complete evaluating procedure can be defined as follows.

- 1. Finding the nearest individual \mathcal{Y} and niche N by x;
- 2. Calculating f and p(x, y, n, N) by formula(1);
- 3. Generating a random number r between 0 and 1;
- 4. If $r \le p(x, y, n, N)$, then x is evaluated and inserted to population.

3.3 Selection, Crossover and Mutation

To make sure that the algorithm can locate and maintain many optima simultaneously, we use fitness-proportionate selection to select a niche according to the average space of each individual in niche. Then, Tournament selection operator is applied to the current niche to generate two offspring. Next, random single point crossover operator is used. The mutation operator may maintain the diversity of population and prevent the genetic algorithm from premature convergence. The algorithm adopts random selection operator to select a offspring and random mutation operator to generate a new individual.

4. Performance Evaluation

4.1 Experiment Setting and Measures

4.1.1 Benchmark Function

To assess the performance of the proposed approach, we compare its result with those obtained by five other NGAs, Clearing (CPGA)[10], Dynamic Niche Sharing(DNSGA) [9], Deterministic Crowding(DCGA) [6], Multi-niche Crowding (MNCGA) [12] and CMNGA [14]. We adopted 6 multimodal benchmark functions, namely F4, F5, F6, F7, F8 and F9 from [16]. They have different characteristics, such as multiple even and uneven niche, multiple equal and unequal maxima. Their detail can be found in [16].

4.1.2 Evaluation Measures

We use peak ratio (PR) and peak accuracy (PA) measure [17] to judge convergence speed and the ability to locate and maintain all fitness function peaks. The details can be found in [17].

$$PR = \frac{Number of the found peaks}{Total number of peaks}$$
(2)

$$PA = \sum_{i=1}^{N_{peaks}} |f(x_i) - f(peak_i)|$$
(3)

4.1.3 Parameter Setting

 N_p denotes the population size; p_c denotes the probability of crossover; p_m denotes the probability of mutation; g_{max} denotes the max number of generations. In DNSGA and CPGA, σ_{sh} denotes a niche radius. N_{peak} denotes the number of peak in DNSGA; N_{kappa} denotes the capacity of niche in CPGA; C_s denotes the crowding selection size, C_f denotes Crowding factor, and S denotes the crowding factor group size in MNCGA; N_i denotes initial the population size, N_e denotes the max number of evaluation, N_c denotes the number of crossover, and N_m denotes the number of mutation in CMNGA and EPNCGA; N_{crowd} denotes the crowding selection size, and N_{min} denotes the number of the nearest neighbors in CMNGA. The following is Parameter Setting.

Table 1. Parameter Setting

Method	Parameter	F4	F5	F6	F7	F8	F9
CPGA	N_p	50	50	50	50	100	100
DNSGA	p_c	0.9	0.9	0.9	0.9	0.9	0.9
DCGA	p_m	0.01	0.01	0.01	0.01	0.01	0.01
MNCGA	g_{max}	500	500	500	500	1000	1200
CPGA	σ_{sh}	0.2	0.2	0.2	0.2	2	0.5
	N _{peak}	6	6	6	6	8	10
DNSGA	σ_{sh}	0.2	0.2	0.2	0.2	2	0.5
Dittoort	N _{kappa}	9	9	9	9	10	12
	C_s	15	15	15	15	25	30
MNCGA	C_{f}	3	3	3	3	3	4
	Ś	15	15	15	15	25	30
	Ni	30	30	30	30	50	50
CMNGA	Ne	600	600	600	600	1500	1500
EPNCGA	N_c	5	5	5	5	8	8
	N_m	10	10	10	10	20	20
CMNGA	N _{crowd}	10	10	10	10	10	10
2	N _{min}	5	5	5	5	8	8

4.2 Experiment result

Convergence speed of the algorithm is determined by the number of fitness evaluations that the algorithm requires to locate all optima, within a fixed accuracy level ε . We evaluate their performance in two accuracy level, $\varepsilon \in \{1E-2, 1E-3\}$. Table 2 shows the average performance of the algorithm (mean) and its standard deviation(std). we can clearly observe that the proposed approach produces faster convergence speed in the majority of benchmark functions. Though all approach almost can locate all peaks within 1500 fitness evaluations in the accuracy level $\varepsilon = 1E - 2$, the EPNCGA and CMNGA show remarkable performance. They can locate and maintain all peaks within 550, but EPNCGA performs better in the majority of the functions. These data illustrate the EPNCGA improve the search efficiency by evaluated probability. Figure 1 presents the value of PR and PA during the run of each approaches(mean of 100 independent runs) for 2D function F8. Again, the EPNCGA exhibits better performance with respect to locating the peaks and fine-tuning the solutions found. Within 200 fitness evaluations, the EPNCGA can locate all peaks and has the best peak accuracy. This figure also illustrates that the EPNCGA can improve both the search quality and efficiency. Figure 2 shows contour plots of 2D function F9 and the positions of all individuals evaluated by the EPNCGA. The number of individuals is 600. The red symbol \triangle denotes the position of peak and the black point denotes evaluated individual. Figure 2 presents the distribution of individuals. The more individuals concentrate on the area nearby peaks, the fewer are on the area far away from peaks. Also, the distribution of individuals is similar to the normal distribution. Overall, the EPNCGA can identify and maintain most peaks in the multimodal function.

	Accuracy	y F4		F5		F6		F7		F8		F9	
Method	level	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
EPNCGA	1E-2	62	16	72	40	64	18	66	20	64	18	547	276
21.10011	1E-3	142	36	204	102	139	28	208	85	139	28	1240	8
CUDICA	1E-2	70	30	70	21	80	61	80	76	80	61	485	190
CMINGA	1E-3	195	98	221	92	180	88	254	150	180	88	1199	287
DNSGA	1E-2	167	85	189	120	180	119	299	284	181	120	*	*
	1E-3	772	290	13352	7099	754	351	15860	5160	754	351	*	*
CPGA	1E-2	348	456	1002	1533	296	336	1047	1664	296	336	*	*
	1E-3	1754	1407	16284	4708	1864	1222	17173	3353	1864	1222	*	*
DCGA	1E-2	1923	2358	703	2775	1152	3877	3142	6832	1152	3877	*	*
	1E-3	8955	4371	17611	6311	8893	8192	18232	5456	8893	8192	*	*
MNCGA	1E-2	691	1824	927	2153	1021	2533	523	845	1021	2533	*	*
WINCGA	1E-3	3357	2534	14263	8582	7393	8467	15673	7553	7393	8467	*	*

Table 2. Convergence speed in two accuracy level

*: the algorithm only finds part of all optima



Figure 1. Peak Accuracy and Peak Radio of F8 function



Figure 2. Contour Plot of F9 by EPNCGA

5. CONCLUSION

Multi-modal optimization algorithms need locate and maintain many optima simultaneously, so both the search quality and efficiency of these algorithms may be influenced. Here, we attempt to overcome these influence by introducing two mechanisms into the algorithm: cumulative population technique and an evaluated probability of new individuals. The first mechanism is designed to keep the found solutions by storing all known information, whilst the second is responsible for exploiting unexplored space effectively by guiding the exploration process. Comparing experimental results with 5 different NGAs on 6 well known multimodal functions of different characteristics, show that the proposed approach outperforms other algorithms. Among most benchmark functions, the proposed approach not only increases the probability of finding both global and local optima, but also reduces the average number of function evaluations.

In the future, we will apply the proposed algorithm to study the higher dimensions and the expensive multimodal problem.

6. ACKNOWLEDGMENTS

This work is supported by National Key Technology R&D Program of China under Grant 2013BAI01B01.

7. REFERENCES

- [1] M.G. Epitropakis, et al., Multimodal optimization using niching differential evolution with index-based neighborhoods, in 2012 IEEE Congress on Evolutionary Computation. CEC 2012, pages 1-8, 2012.
- [2] M.G. Epitropakis, et al., A dynamic archive niching differential evolution algorithm for multimodal optimization, in 2013 IEEE Congress on Evolutionary Computation (CEC), pages 79-86, 2013.
- [3] E. Cuevas and M. González, An optimization algorithm for multimodal functions inspired by collective animal behavior, *Soft Computing*, 17(3), pages 489-502, 2013.
- [4] B.Y. Qu, et al., Niching particle swarm optimization with local search for multi-modal optimization, *Inform Sciences*, 197, pages 131-143, 2012.
- [5] M. Li, et al., A hybrid niching PSO enhanced with recombination-replacement crowding strategy for multimodal function optimization, *Applied Soft Computing*, 12(3), pages 975-987, 2012.
- [6] R. Manner, et al., Crowding and Preselection Revisited, *Parallel Problem Solving from Nature*, pages 27-36: North-Holland, 1992.
- [7] O.J. Mengshoel and D.E. Goldberg, The Crowding Approach to Niching in Genetic Algorithms, *Evolutionary Computation*, 16(3), pages 315-354, 2008.
- [8] B. Sareni and L. Krahenbuhl, Fitness Sharing and Niching Methods Revisited, *IEEE transactions on Evolutionary Computation*, 2(3), pages 97-106, 1998.
- [9] B.L. Miller and M.J. Shaw, Genetic algorithms with dynamic niche sharing for multimodal function optimization, in *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 786-791, 1996.
- [10] A. Petrowski, A clearing procedure as a niching method for genetic algorithms, in *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 798-803, 1996.
- [11] J. Gan and K. Warwick, A Genetic Algorithm with Dynamic Niche Clustering for Multimodal Function Optimisation, *Artificial Neural Nets and Genetic Algorithms*, Springer Vienna, pages 248-255, 1999.
- [12] W. Cedeño, et al., Multi-niche crowding in genetic algorithms and its application to the assembly of DNA restriction-fragments, *Evolutionary Computation*, 2, pages 321-345, 1995.
- [13] J.E. Vitela and O. Castaños, A sequential niching memetic algorithm for continuous multimodal function optimization, *Appl Math Comput*, 218(17), pages 8242-8259, 2012.
- [14] M. Hall, A Cumulative Multi-Niching Genetic Algorithm for Multimodal Function Optimization, *International Journal of Advanced Research in Artificial Intelligence*, 1(9), 2012.
- [15] C. Lin and W. Wu, Niche identification techniques in multimodal genetic search with sharing scheme. *Advances in Engineering Software*, 33, pages 779-791, 2002.
- [16] S. Das, et al., Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art, *Swarm and Evolutionary Computation*, 1(2), pages 71-88, 2011.
- [17] R. Thomsen, Multimodal optimization using crowding-based differential evolution, in *Congress on Evolutionary Computation CEC2004*, pages 1382-1389, 2004.