2012 IEEE International Conference on Information Science and Technology
Wuhan, Hubei, China; March 23-25, 2012

An Adaptive Hybrid Combination of PSO and Extremal Optimization

Zhenyi Chen, Gaofeng Wang, and Chen Dong

Abstract—**Particle Swarm Optimization (PSO) has proved to be an effective global optimization in recent years. However, PSO still suffers from the prematurity to local optima. In order to solve this disadvantage, researches have carried out by combination with other optimizers. In recent years, a local optimization called Extremal Optimization (EO) has been introduced into PSO and gain improvements. Although, the combination of PSO with EO would bring severe computation overhead result in the simple way they combined. In this paper, an adaptive hybrid combined PSO (AHPSO-EO) is proposed. It can improve the computation effectively by an adaptive way. The experimental results on benchmark functions reveal that the adaptive PSO combined with EO accelerates the convergence and improves the performance of proposed algorithm.**

I. Introduction

THE particle swarm optimization (PSO) was originally proposed by Eberhart and Kennedy [1]. PSO was inspired by observation of social behavior of bird flocking. It employs an iteration based search to locate global optimal solution. The PSO has been widely applied to many optimization problems due to its powerful capability to search for the global optimal solution and easy implementation. However, PSO has the disadvantages in premature convergence. While in dealing with multimodal functions, PSO is easy to convergence early to a local optima.

Extremal Optimization (EO) is a powerful general-purpose heuristic local search algorithm which was developed by Boettcher and Percus [2]. In EO, the weak particles are used to generate the next generation and mutate the closest neighbors. The algorithm had explored many obstacles to find effectively the local optima after large fluctuations. EO is initially designed to solve combinatorial optimization problems like traveling salesman problem. It is also used successfully to solve variant hard NP problems.

There are several combinations of PSO with EO are proposed recently in order to overcome the disadvantage of premature convergence. In these algorithms, PSO is used to guarantee the fast convergence, and EO to jump out from the local optima. But to the best of the authors' knowledge, in all these papers, mutation operators were randomly applied on the particles in PSO algorithm during all period. As we all know, the particles would go slower and slower and at last keep stillness at last. In order to prevent the swarm from going still too fast, EO must act more often as the algorithm goes.

Manuscript received December 5, 2001.

This work was supported by the National Natural Science Foundation of China under the National Outstanding Young Scientist Award No. 60788402.

Z. Chen is with the Computer School, Wuhan University, Wuhan, Hubei 430072, China. He is also with the Fuzhou Electric Power Bureau, Fuzhou, Fujian 350009, China.

G. Wang is with the Institute of Microelectronics and Information Technology, Wuhan University, Wuhan, Hubei 430072, China (e-mail: gaofeng@ whu.edu.cn).

C. Dong is with College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian 350108, China.

In this work, an adaptive PSO framework with a hybrid EO was proposed for benchmark function optimization. The paper is organized as follows. In Section II, a brief overview of the standard PSO and extremal optimization is contained. In Section III, a hybrid PSO is elaborated with novel operations for tackling the problem. In Section IV, on solving 3 unimodal and multimodal benchmark functions, the experimental results of the comparison between the results of the proposed and other methods are analyzed. Conclusions are drawn in Section V.

I. Problem Formulation

A. Particle Swarm Optimization

PSO is one of the evolutionary computation techniques. The standard PSO [3] considers a swarm S containing n particles (S =1, 2, ⋯, n) in a D-dimensional continuous solution space. Each i-th particle is represented by its position denoted as a D-dimensional vector: $X_i = (X_{i1}, X_{i2},\ldots, X_{iD})$, where X_{ij} is the j-th coordinate component of the position of the i-th particle. Moreover, the velocity of the i-th particle can also be defined by another D-dimensional vector: $V_{\mathrm{i}} = (V_{\mathrm{i}1}, V_{\mathrm{i}2}, \ldots, V_{iD})$. The position of a particle X_{i} represents a possible solution to the optimization problem under study, while the velocity vi gives the change rate for the position of i-th particle in the next iteration. A fitness value for a particle position can be obtained by evaluating the objective function at the interested particle position. The fitness value is an indictor of the quality of the particle position as a solution candidate to the optimization problem under study. The best position of a particle at a given iteration stage corresponds to the position that gives the best fitness value among all its historical positions so far.

Denote the best position of the i-th particle as $p_{\mathrm{i}} = (p_{i1}, p_{i2}, \ldots, p_{iD})$ and the index of the best particle position among $\{p_i\}$ in the population (i.e., the best position so far of the population) as g. Then, the particle positions and velocities can be updated according to the following equations.

$$V_{id} \leftarrow w\cdot V_{id} + c_1\cdot rand()\cdot(p_{id}-X_{id}) + c_2\cdot rand()\cdot(p_{gd}-X_{id}) \quad (1)$$

$$X_{id} \leftarrow X_{id} + V_{id} \quad (2)$$

Where w is the inertia weight, rand() is random function whose value is between 0 and 1, and c1 and c2 are both positive constants, which are called as acceleration coefficients. A maximum allowable velocity, Vmax, is a prescribed constant that is often used for limiting the velocity of the particles.

B. Extremal Optimization (EO)

Extremal Optimization (EO) was inspired by the Bak-Sneppen (BS) model, which shows the emergence of self-organized criticality (SOC) [4] in ecosystems. In EO algorithm, set a solution $X = (x_1, x_2, x_3)$ is randomly generated. The variable x1, x2, and x3 at the current solution X is called components of X. The fitness value λ_i of each component xi is being ranked.

The worst component is mutated and gets a new random value that results in a new solution. It would be accepted to replace the current solution if the new solution has a better fitness. Otherwise, mutation is continued as long as desired. The change in the fitness of a variable leads to its neighbors' fitness changes. Through performing mutation on the worst variable and its components, through keeping mutation on the worst components and its neighbors' successively, the EO can avoid getting stuck in local minima and keep improving the results.

For a minimization problem with n variable, EO can be described as follows:

1. Generate an individual S. Set the optimal solution Sbest = S
2. For the "current" individual S

(a) evaluate the fitness λ_i for each decision variable, A, ie(,. n)

(b) find j satisfying $\lambda_j \leq \lambda_i$, for all *i*, i.e., x_j creates the "worst fitness",

(c) choose $S' \in N(S)$ such that x_j must be change its state, N(S)is the neighborhood of S,

(d) accept S = S unconditionally,

(e) if the current cost function value is less than the minimum cost function value, i.e. c(S) < C(Sbest), then set Sbest = S

3. Repeat at Step 2 as long as desired.
4. Return S*best* and C (S*best*).

From an optimization point of view, EO may replace the extremely undesirable variables of a single sub-optimal solution with new ones randomly or according to power-law distributions. To avoid getting stuck in local minima and to improve the results, the specific algorithmic flow of τ-EO is referred to [5].

II. The Proposed adaptive PSO combinated with EO

In the presented combination of PSO, the EO was combined to prevent PSO from early convergence. To the best of the authors' knowledge, in all the proposed combination of PSO, EO is just run in fixed interval iterations of PSO (PSO-EO) and the optimal interval values were set based on the experimental results [6]. However, the EO in the proposed combination PSO will result severely in extra computations, an ill-suited interval can deteriorate the PSO.

In the paper, an adaptive hybrid combination of PSO and EO (AHPSO-EO) is proposed. In order to save the computational overhead and take full advantage of the PSO's fast convergence to the global minimum in earlier stage of algorithm, there is no much need for the EO to join the searching of the algorithm. On the contrary, the PSO is easily to be trapped into local optima in later stage and the EO would be employed more often to get the PSO to escape from local optima. Hence, a linear decrease strategy is developed for the interval to provide a balance between global and local exploration abilities in the proposed PSO to reduce the expense of computing and the average number of iterations to locate the optimum. The interval *Inv* can be counted as follows.

$$Inv = \left\lfloor Inv_{start} - iter * \frac{(Inv_{start} - Inv_{end})}{iterations} \right\rfloor \quad (3)$$

Where Inv_{start} and Inv_{end} are the range of interval iterations.

The AHPSO-EO can be summarized as follows.

1) Initialize the particle swarm with random positions and velocities within the search space.

2) Calculate the fitness value of each particle.

3) Update the personal best position p_i for each particle and the global best position p_g for the population.

4) Update the velocity and position of each particle by using equation (1) (2).

5) Calculate the *Inv* according to equation (3).

6) For simplicity, if $\mathrm{mod}(iter, Inv) = 0$, then EO using Gauss mutation operator was executed. The mutation that results in a better fitness value will be accepted. Otherwise, if none of them results in a better particle, this mutation will be repeated for *Tg* times, until getting an acceptable solution.

Check the termination criterion is satisfied: If the criterion is met, then stop. Otherwise, go to step (2).

III. Experiments And Discussions

The proposed PSO discussed in the preceding sections will be tested on some unimodal and multimodal benchmark functions under the same conditions in this section. The performance of this AHPSO-EO algorithm will be evaluated in comparison with the standard PSO (S-PSO) algorithm, the linearly decreasing inertia weight PSO (LD-PSO) algorithm and the PSO-EO.

A. Benchmark Functions

Four benchmark functions popularly used will be used here for the experiments on the new adaptive PSO algorithm.

1) Rosenbrock function

$$f_1(x) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (4)$$

2) Rastrigrin function

$$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) \quad (5)$$

3) Griewank function

$$f_3(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (6)$$

4) Ackley function

$$f_4(x) = 20 + e - 20 e^{\left[-0.2\sqrt{(1/n)\sum_{i=1}^{n} X^2_i}\right]} - e^{(1/n)\sum_{i=1}^{n}\cos(2\pi X_i)} \quad (7)$$

B. Experiments Settings and Results

The parameters c_1 and c_2 are set at 2; and the value of w decrease from 0.9 to 0.4 in a linear way through the iterations. *Tg* is set at 3 in PSO-EO and AHPSO-EO. The value of *Inv* in PSO-EO is set at 100. While in AHPSO-EO, the *Inv* decreases linearly from 150 to 100. The other common parameter settings are listed in Table 1.

TABLE I. PARAMETERS SETTING FOR INITIALIZATION

Function	Dimension	Max generation	Swarm size	Search space	Global min
f_1	30	10000	30	(-30,30)	0
f_2	30	10000	30	(-5.12,5.12)	0
f_3	30	10000	30	(-600,600)	0
f_4	30	10000	30	(-32.768,32.768)	0

The experiments were executed in computer AMD II X2 2.61G and 2G memory with system Windows XP SP3. The programs are written in MATLAB 7.0.4 and perform 20 times for every function. The worst mean, and best results found by each algorithms are listed in Tables II-V. The results are also shown in these tables.

TABLE II. COMPARED RESULTS ON ROSENBROCK FUNCTION

Algorithm	worst	mean	Best
AHPSO-EO	98.99078	32.24527	0.010611
MPSO-EO	99.19224	40.02358	0.339398
LD-PSO	71.39845	18.59915	3.514403
S-PSO	551.9085	216.5492	88.68766

C. Discussions

The Rosenbrock function is a classic optimization problem that the global optimum is inside a long, narrow, parabolic sh-ed valley. The AHPSO-EO converged to a closer position to the optimum and gets a better value than other algorithms. The Rastirgrin is a multimodal function, which has a lot of local optimums. It's very difficult for general algorithms to locate the global optima. AHPSO-EO is far more efficient that it found the global optima five times in experiments and shows great accuracy in the tests. The Griewank function is a highly multimodal benchmark function that it is easy for the optimization algorithm to be trapped into local optimum. The AHPSO-EO performs best in the four algorithms and it found the global optima 8 times. The Ackely function is multimodal. The AHPSO-EO also shows best performance among four algorithms.

TABLE III. COMPARED RESULTS ON RASTRIGRIN FUNCTION

Algorithm	worst	mean	Best
AHPSO-EO	0.002384	0.000136	0
MPSO-EO	0.015502	0.001029	6.25E-13
LD-PSO	50.7429	29.55025	16.9143
S-PSO	135.284	100.4596	58.5686

TABLE IV. COMPARED RESULTS ON GRIEWANK FUNCTION

Algorithm	worst	mean	Best
AHPSO-EO	0.049208	0.013662	0
MPSO-EO	0.039346	0.013053	0
LD-PSO	0.041856	0.018464	0
S-PSO	0.039731	0.010772	0.000247

TABLE V. COMPARED RESULTS ON ACKLEY FUNCTION

Algorithm	worst	mean	Best
AHPSO-EO	1.33E-14	6.57E-15	6.22E-15
MPSO-EO	1.33E-14	7.64E-15	6.22E-15
LD-PSO	1.33E-14	6.57E-15	2.66E-15
S-PSO	0.367094	0.143622	0.022796

IV. CONCLUSIONS

It is desirable to further apply it to solving more complex engineering optimization problems in the real world. We believe this novel AHPSO-EO may be attractive for hard combinatorial optimization problems. The future work includes the studies on how to extend AHPSO-EO to solve more problems and explore the efficiency of it on discrete NP problems.

It's a new way to improve the PSO by hybrid with other optimizer, and more tests should be conducted on this way. In addition, the optimal values of the parameters in the proposed methods will be investigated in the future.

REFERENCES

[1] R. C. Eberhart and J. Kennedy, "A new optimizer using particles swarm theory," In *Proceedings of the sixth International Symposium on Micro Machine and Human Science*, Nagoya, 1995, pp. 39-43.

[2] S. Boettcher and A G. Percus, "Extremal Optimization: Methods derived from Co-Evolution," in *Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, 1999, pp. 101-106.

[3] Y. H. Shi and R. C. Eberhart, "A Modified Particle Swarm Optimizer," In *IEEE International conference of Evolutionary Computation*, Piscataway, 1998 , pp. 69-73.

[4] P. Bak and K. Sneppen, "Punctuated equilibrium and criticality in a simple model of evolution," *Physical Review Letters*, vol. 71, no.24, pp. 4083- 4086, 1993.

[5] S. Bocttcher and A.G. Percus, "Optimization with extremal dynamics," *Physical Review Letters*, vol. 86, no. 23, pp. 5211-5214, 2001.

[6] M.R. Chen, X. Li, X. Zhang, and Y. Z. Lu, "A Novel particle swarm optimizer hybridized with extremal optimization,"*Applied Soft Computing*, vol.10, no. 2, pp. 367-373, 2010.