International Conference on Communication, Circuits and Systems (ICCCAS), Chengdu, China, 2002: 1170-1173

Hybrid Particle Swarm Optimizer with Mass Extinction

Xiao-Feng Xie, Wen-Jun Zhang, Zhi-Lian Yang
Institute of Microelectronics, Tsinghua University, Beijing 100084, P.R.China
Email: xiexiaofeng@tsinghua.org.cn

Abstract - A hybrid particle swarm optimizer with mass extinction, which has been suggested to be an important mechanism for evolutionary progress in biological world, is presented to enhance the capacity in reaching optimal solution. The testing results of three benchmark functions that typically used in evolutionary optimization research indicate this method improves the performance effectively.

1. Introduction

The particle swarm optimization (PSO) algorithm is an evolutionary computation technique that originally introduced by Kennedy and Eberhart in 1995 [1, 2]. The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm theory [3]. Work presented in [4, 5] describes the complex task of parameter selection in the PSO model. Several researchers have analyzed the performance of the PSO with different settings, e.g., neighborhood settings [6], cluster analysis [7]. It has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement [8].

Comparisons between PSO and the standard GA were done analytically [9] and also with regards to performance [10]. Angeline [10] points out that the PSO performs well in the early iterations, but has problems reaching a near optimal solution in several real-valued function optimization problems. Both Eberhart [9] and Angeline [10] conclude that hybrid models of the standard GA and the PSO, could lead to further advances.

Paleontological findings have revealed that mass extinction has been a common phenomenon in evolution [11]. It has been suggested to be an important mechanism for evolutionary progress in biological world [12], since extinction allows the repopulation of niches and gives space for new adaptations. In the field of evolutionary algorithms, this idea has been the motivation for so-called (mass-) extinction models, which has been introduced recently [13-15]. It is therefore natural to ask if mass extinction can be exploited to increase the efficiency of PSO. Here we review the role of mass extinction in the fossil record and simulate this process in a hybrid particle swarm optimizer with mass extinction. Both standard and hybrid versions are compared on three numerical optimization problems typically used in evolutionary optimization research. The preliminary results suggest that mass extinction can enhance the performance.

2. Standard particle swarm optimization (SPSO)

The fundament to the development of PSO is a hypothesis [16] that social sharing of information among conspeciates offers an evolutionary advantage. PSO is similar to the other evolutionary algorithms in that the system is initialized with a population of random solutions. However, each potential solution is also assigned a randomized velocity, and the potential solutions, call *particles*, corresponding to individuals. Each particle in PSO flies in the D-dimensional problem space with a velocity which is dynamically adjusted according to the flying experiences of its own and its colleagues. The location of the ith particle is represented as $X_i = (x_{i1},\ldots, x_{id}, \ldots, x_{iD})$, where $x_{id}\in[l_d, u_d]$, $d\in[1, D]$, l_d, u_d are the lower and upper bounds for the dth dimension, respectively. The best previous position (which giving the best fitness value) of the ith particle is recorded and represented as $P_i = (p_{i1},\ldots, p_{id}, \ldots, p_{iD})$, which is also called *pbest*. The index of the best particle among all the particles in the population is represented by the symbol g. The location P_g is also called *gbest*. The velocity for the ith particle is represented as $V_i = (v_{i1},\dots, v_{id}, \dots, v_{iD})$, which is clamped to a maximum velocity $V_{max} = (v_{max,1},\dots, v_{max,d}, \dots, v_{max,D})$, which is specified by the user.

The particle swarm optimization concept consists of, at each time step, changing the velocity and location of each particle toward its *pbest* and *gbest* locations according to the equations (1a) and (1b), respectively:

$$v_{id} = w * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * rand() * (p_{gd} - x_{id}) \quad (1a)$$

$$x_{id} = x_{id} + v_{id} \quad (1b)$$

Where w is inertia weight [17], c_1 and c_2 are acceleration constants [8], and *rand*() is a random function in the range [0, 1]. For equation (1a), the first part represents the inertia of pervious velocity; the second part is the "cognition" part, which represents the private thinking by itself; the third part is the "social" part, which represents the cooperation among the particles [18]. If the sum of accelerations would cause the velocity v_{id} on that dimension to exceed $v_{max,d}$, then v_{id} is limited to $v_{max,d}$. V_{max} determines the resolution with which regions between the present position and the target position are searched [4, 8].

The process for implementing PSO is as follows:

a). Set current iteration generation G_c=1. Initialize a population which including m particles, For the ith particle, it has random location X_i in specified space and for the dth dimension of velocity V_i, $v_{id} = Rand_2() * v_{max,d}$, where $Rand_2()$ is a random function in the range [-1, 1];

b). Evaluate the fitness for each particle;

c). Compare the evaluated fitness value of each particle with its *pbest*. If current value is better than *pbest*, then set the current location as the *pbest* location. Furthermore, if current value is better than *gbest*, then reset *gbest* to the current index in particle array;

d). Change the velocity and location of the particle according to the equations (1a) and (1b), respectively;

e). $G_c=G_c+1$, Loop to step b) until a stop criterion is met, usually a sufficiently good fitness value or G_c is achieve a predefined maximum generation G_{max}.

The parameters of PSO includes: number of particles m, inertia weight w, acceleration constants c_1 and c_2, maximum velocity V_{max}.

3. Hybrid PSO with mass extinction (HPSO)

During Earth's long history, environmental stresses have been prolonged and/or severe enough to invoke widespread ecological instability to induce the simultaneous extinction of many species. These events are recorded in the fossil record are known as mass extinctions [11], which has played a key mechanism in shaping the history of life on Earth. Although mass extinctions account for only a small percentage of all extinctions, they are singularly important because the remove stagnant groups from niches, creating the opportunity for new species to flourish and establish an ecological niche [19].

For standard PSO, the concept of inertia weight w was introduced by Shi [17] to satisfy the requirements for different balances between the local search ability and global search ability for different problems. The small or time decreasing w is usually adopted [4] to decrease the average number of iterations required. However, as iteration generations increases, the diversity of particles will be diminished and lead to equilibrium in swarm. Since there have little difference among the information of particles, the foundation of PSO, i.e. social sharing of information, will lost the effectivity. The particles become inactively, which is flying with very small velocities.

The mass extinction from natural system will also make for the successive evolution of the social model in PSO, since the social system is also following the natural laws. For the simple hybrid PSO model in this work, the mass extinction is performed by reinitializing the velocities of all particles at a predefined extinction interval I_e after the step d) of PSO process. The pseudocode is shown in Fig. 1.

```
// after step d) in standard PSO process
IF (Gc%Ie==0) {   // reinitialize swarm velocities
   FOR (i=0; i<m; i++) {
      FOR (d=1; d≤D; d++) {
         vid = Rand2() * vmax,d;
      }
   }
}
```

FIG. 1 Pseudocode of mass extinction model

4. Results and discussion

For comparison, three benchmark functions that are commonly used in the evolutionary computation literature [5, 10, 20] are used. All functions have same minimum value, which are equal to zero. The f_2 and f_3 are multimodal functions.

The function f_1 is the Rosenbrock function:

$$f_1(x) = \sum_{d=1}^{D-1} (100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2) \quad (2a)$$

The function f_2 is the generalized Rastrigrin function:

$$f_2(x) = \sum_{d=1}^{D} (x_d^2 - 10\cos(2\pi x_d) + 10) \quad (2b)$$

The function f_3 is the generalized Griewank function:

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

For the purpose of comparison, the asymmetric initialization method used in [5, 10, 20] is adopted here for population initialization. Table 1lists the initialization ranges, and table 2 lists the V_{max} and X_{max} values for all the functions, respectively. The acceleration constants are set as: c_1=c_2=2. The fitness value is set as function value. We had 500 trial runs for every instance.

Table 1: Asymmetric initialization ranges

Function	Range
f_1	(15,30)
f_2	(2.56,5.12)
f_3	(300,600)

Table 2: V_{max} and X_{max} values for each function

Function	X_{max}	V_{max}
f_1	100	100
f_2	10	10
f_3	600	600

In order to investigate whether the hybrid PSO scales well or not, different numbers of particles m are used for each function which different dimensions. The numbers of particles m are 20, 40, 80 and 160. G_{max} is set as 1000, 1500 and 2000 generations corresponding to the dimensions 10, 20 and 30, respectively.

TABLE 3: The mean fitness values for the Rosenbrock function

m	D.	G_{max}	*FPSO*[20]	$HPSO_1$	$HPSO_2$
20	10	1000	66.01409	70.41591	45.11909
	20	1500	108.2865	136.01683	92.49036
	30	2000	183.8037	143.82718	127.53624
40	10	1000	48.76523	40.93693	31.12621
	20	1500	63.88408	95.87104	58.97711
	30	2000	175.0093	113.38927	82.8878
80	10	1000	15.81645	28.89738	19.61489
	20	1500	45.99998	64.93305	38.71401
	30	2000	124.4184	92.5544	64.3218

TABLE 4: The mean fitness values for the Rastrigrin function

m	D	G_{max}	*FPSO*[20]	$HPSO_1$	$HPSO_2$
20	10	1000	4.955165	2.92628	6.09713
	20	1500	23.27334	14.99364	30.13731
	30	2000	48.47555	35.5962	71.67209
40	10	1000	3.283368	1.45853	2.97695
	20	1500	15.04448	9.00546	17.59697
	30	2000	35.20146	21.1315	43.53199
80	10	1000	2.328207	0.6198	1.48249
	20	1500	10.86099	5.11605	10.99831
	30	2000	22.52393	13.02084	28.87577

TABLE 5: The mean fitness values for the Griewank function

m	D.	G_{max}	*FPSO*[20]	$HPSO_1$	$HPSO_2$
20	10	1000	0.091623	0.09100	0.08626
	20	1500	0.027275	0.02483	0.03203
	30	2000	0.02156	0.01569	0.04035
40	10	1000	0.075674	0.07873	0.07300
	20	1500	0.031232	0.02202	0.02345
	30	2000	0.012198	0.01299	0.01415
80	10	1000	0.068323	0.06815	0.06171
	20	1500	0.025956	0.02262	0.02258
	30	2000	0.014945	0.01084	0.01143

Table 3 to 5 lists the mean fitness values for the three benchmark functions. Where ***FPSO*** is the results of fuzzy adaptive PSO in [20] with fuzzy adaptive w, and $\boldsymbol{HPSO_1}$ and $\boldsymbol{HPSO_2}$ is the results of hybrid PSO with mass extinction, where w=0.4, and I_e is set as $G_{max}/20$ and $G_{max}/5$, respectively.

By compare the results, it is easy to see that $HPSO_2$ in Table 3, and $HPSO_1$ in Table 4 and 5 have better results than *FPSO* for almost all cases. However, $HPSO_1$ in Table

3, and $HPSO_2$ in Table 4 and 5 have better results in some cases only. It means the performance may be affected by different extinction interval I_e. The best performance should be achieved in the cases that the mass extinctions are occurred when the swarm is going to convergence.

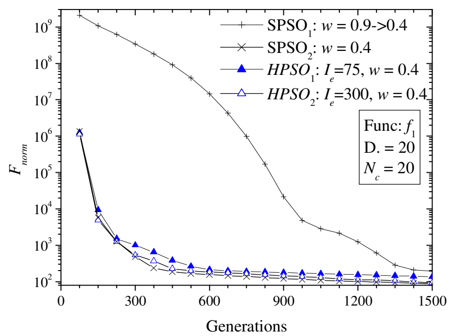

FIG. 2 F_{norm} of Different PSO settings for f_1

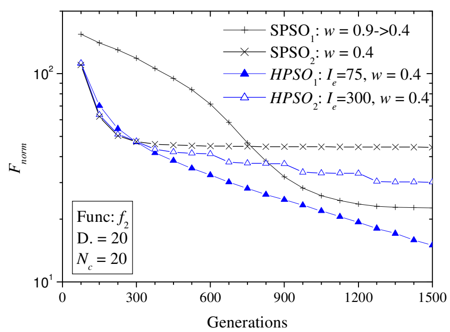

FIG. 3 F_{norm} of Different PSO settings for f_2

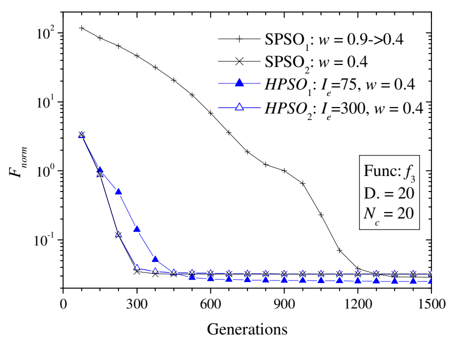

FIG. 4 F_{norm} of Different PSO settings for f_3

Fig 2 to 4 show the mean fitness value of the best particle found (F_{norm}) during 1500 generations for the three benchmark functions with 20 dimensions, respectively. Where $\boldsymbol{SPSO_1}$ and $\boldsymbol{SPSO_2}$ are the results of standard PSO with a linearly decreasing w which from 0.9 to 0.4 or be fixed as 0.4, respectively.

For the two multimodal functions f_2 and f_3, $SPSO_2$ performs worse than $SPSO_1$ since the evolution will be stagnated in some generations. However, with suitable mass extinction interval, the hybrid PSO versions have the capability with sustainable evolution and get better results.

5. Conclusion

In this paper, a hybrid particle swarm optimizer with mass extinction was introduced to improve the performance. The hybrid method provides the natural mechanism to make particles actively, which encourages employing the small inertia weight with fast convergence. Three benchmark functions have been used for testing. The simulation results illustrate the performance of hybrid PSO with mass extinction can improve the performance to some extent, at least for the three benchmark functions.

However, the algorithm performance is depended on the selection of extinction interval I_e, This problem may be overcome by employing a self-adaptive method for adjusting the extinction interval I_e.

References

[1] J. Kennedy, R. Eberhart. Particle swarm optimization. Proc. IEEE Int. Conf. on Neural Networks, 1995: 1942-1948

[2] R. Eberhart, J. Kennedy. A new optimizer using particle swarm theory. Proc. 6th Int. Symposium on Micro Machine and Human Science, 1995: 39-43

[3] M. M. Millonas. Swarms, phase transition, and collective intelligence. In C. G. Langton, (Eds.), Artificial life III. Addison - Wesley, MA, 1994

[4] Y. Shi, R. Eberhart. Parameter selection in particle swarm optimization. Proc. of the 7th Annual Conf. on Evolutionary Programming, 1998: 591-600

[5] Y. Shi, R. Eberhart. Empirical study of particle swarm optimization. Proc. of Congress on Evolutionary Computation, 1999: 1945-1950

[6] P. N. Suganthan. Particle swarm optimiser with neighbourhood operator. Proc. of Congress on Evolutionary Computation, 1999:

1958-1962

[7] J. Kennedy. Stereotyping: improving particle swarm performance with cluster analysis. Proc. IEEE Int. Conf. on Evolutionary Computation, 2000: 1507-1512

[8] R. Eberhart, Y. Shi. Particle swarm optimization: developments, applications and resources. Proc. IEEE Int. Conf. on Evolutionary Computation, 2001: 81-86

[9] R. C. Eberhart and Y. Shi. Comparison between genetic algorithms and particle swarm optimization. Proc. of the 7^{th} Annual Conf. on Evolutionary Programming, 1998: 611–616

[10] P. J. Angeline. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. Proc. of the 7^{th} Annual Conf. on Evolutionary Programming, 1998: 601-610

[11] C. Newton, L. Laporte. Ancient environments. Prentice Hall, New Jersey, 1989

[12] P. Bak. How nature works. Springer-Verlag, NY, 1986

[13] T. Krink, R. Thomsen. Self-organized criticality and mass extinction in evolutionary algorithms. Proc. IEEE Int. Conf. on Evolutionary Computation, 2001: 1155-1161

[14] G. B. Fogel, G. W. Greenwood, K. Chellapilla. Evolutionary computation with extinction: experiments and analysis. Proc. IEEE Int. Conf. on Evolutionary Computation, 2000: 1415-1420

[15] J. Marin, R.V. Solé. Evolutionary optimization through extinction dynamics. Proc. Genetic and Evolutionary Computation Conference, 1999: 1344-1349

[16] E. O. Wilson. Sociobiology: the new synthesis, Belknap Press, Cambridge, MA, 1975

[17] Y. Shi, R. Eberhart. A modified particle swarm optimizer. Proc. IEEE Int. Conf. on Evolutionary Computation, 1998: 69-73

[18] J. Kennedy. The particle swarm: social adaptation of knowledge. Proc. IEEE Int. Conf. on Evolutionary Computation, 1997: 303-308

[19] C. Marshall. Mass extinction probed. Nature, 1998, 392: 17-20

[20] Y. Shi, R. Eberhart. Fuzzy adaptive particle swarm optimization. Proc. IEEE Int. Conf. on Evolutionary Computation, 2001: 101-106