A Novel PSO-DE-Based Hybrid Algorithm for Global Optimization

Ben Niu* and Li Li*

School of Management, Shenzhen University,
Shenzhen, 518060 P.R. China
drniuben@gmail.com, llii318@163.com

Abstract. This paper presents a new hybrid global optimization algorithm PSODE combining particle swarm optimization (PSO) with differential evolution (DE). PSODE is a type of parallel algorithm, in which PSO and DE are executed in parallel to enhance the population with frequent information sharing. To demonstrate the effectiveness of the proposed algorithm, four benchmark functions are performed, and the performance of the proposed algorithm is compared to PSO and DE to demonstrate its superiority.

Keywords: Particle swarm optimization; differential evolution; global optimization.

1 Introduction

The particle swarm optimization (PSO) is motivated from the simulation of simplified social behavior first developed by Kennedy and Eberhart [1, 2]. Due to its simplicity in coding and consistency in performance, it has already been widely used in many areas [3].

Differential evolution(DE) is a population-based parameter optimization technique originally proposed by Price [4].In DE new individuals are generated by mutation and DE's crossover, which cunningly uses the variance within the population to guide the choice of new search points. Although DE is very powerful, there is very limited theoretical understanding of how it works and why it performs well.

In recent years, some attempts have been made to combine the merits of PSO and DE in the context of hybrid methods. Zhang WJ and Xie X F. [5] introduced a hybrid PSO with DE, in which the bell-shaped mutations with consensus on the population diversity by DE operator. Hendtlass T. [6] proposed a hybrid model that each individual obeys the conventional swarm algorithm, but from time to time the DE is run which may move one individuals form a poorer area to a better area to continue the search.

In this paper, a novel hybrid global optimization method, termed PSODE, is introduced for application as a tool in solving challenging global optimization problems.

---

* Corresponding author.

D.-S. Huang et al. (Eds.): ICIC 2008, LNAI 5227, pp. 156–163, 2008.

PSODE is based on a two-population based scheme, in which the individuals of one population is enhanced by PSO and the individuals of the other population is evolved by DE. An information sharing mechanism is presented by the parallel simulation of PSO and DE. The interactions between the two populations influence the balance between exploration and exploitation and maintain some diversity in the whole population, even when it is approaching convergence, thus reducing the risk of convergence to local sub-optima.

The rest of the paper is organized as follows. Section 2 describes the PSO and DE. Section 3 motivates and describes the PSODE algorithm and gives the pseudocode for the algorithm. Section 4 defines the benchmark problems used for experimental comparison of the algorithms, and the experimental settings for each algorithm. Section 5 presents the results followed by conclusions in section 6.

2 Review of Standard PSO and DE

2.1 PSO

The fundament to the development of PSO is hypothesis that a potential solution to the optimization problem is treated as a bird without quality and volume, which we often call a particle, flying through the *D*-dimensional space, adjusting its position in search space according to its own experience and that of its neighbors.

The *i*th particle is represented as $x_i = (x_{i1}, x_{i2}, ... x_{iD})$ in the D-dimensional space, where $x_{id} \in [l_d, u_d], d \in [1, D]$, l_d, u_d are the lower and upper bounds for the *d*th dimension, respectively. The rate of velocity for particle i is represented as $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$, is clamped to a maximum velocity $V_{\max}$, which is specified by the user. In each time step *t*, the particles are manipulated according to the following equations:

$$v_i(t+1) = w \times v_i(t) + R_1 c_1 (P_i - x_i(t)) + R_2 c_2 (P_g - x_i(t)) \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t) \tag{2}$$

where R_1 and R_2 are random values between 0 and 1. c_1 and c_2 are acceleration constants, which control how far a particle will move in a single iteration. w is inertia weight, which often decreases linearly from about 0.9 to 0.4 during a run [7].

2.2 DE

DE technique combines simple arithmetic operators with the classical events of cross-over, mutation and selection to evolve from a randomly generated starting population to a final solution.

$DE/rand/1/exp$ scheme is recommended to be the first choice when trying to apply differential evolution to any given problem [4]. This particular version is adopted in our work, which is briefly descried as follows. For a minimization problem, i.e. $\min F(\bar{x})$, DE starts to work with a population of N candidate solutions, i.e. $\vec{x}_i^t$, $i = 1, 2, ...N$, where i indexes the population and t is the current generation.

For the mutation operation , a perturbed vector $\vec{v}_i^t$ is generated according to

$$\vec{v}_i^t = \vec{x}_{r1}^t + F(\vec{x}_{r2}^t - \vec{x}_{r3}^t) \tag{3}$$

with random indexes $r1, r2, r3 \in \{1, 2, ...N\}$ and a scaling factor $F \in [0, 2]$.

For the crossover operation, the perturbed vector $\vec{v}_i^t = [v_{i1}, v_{i2}, ...v_{iD}]$ and target vector $\vec{x}_i^t = [x_{i1}, x_{i2}, ...x_{iD}]$ both are used to generate a trial vector $\vec{x}_i^{t'} = [x'_{i1}, x'_{i2}, ...x'_{iD}]$:

$$x'_{ij} = \begin{cases} v_{ij,}\ if\ randb(j) \le CR\ or\ j = randr(i) \\ x_{ij,}\ if\ randb(j) \ge CR\ and\ j \ne randr(i) \end{cases}, \tag{4}$$

where $j \in [1, D], randb(j) \in [0,1]$ is the jth evaluation of a uniform random number generator, $CR \in [0,1]$ is the crossover constant. $randr(i) \in [1, 2, ...D]$ is a randomly chosen index which ensures that $\vec{x}_i^{t'}$ gets at least one parameter from $\vec{v}_i^t$.

For selection operation, a greedy scheme is performed:

$$\vec{x}_i^{t+1} = \begin{cases} \vec{x}_i^{t'}, if\ \Phi(\vec{x}_i^{t'}) < \Phi(\vec{x}_i^t) \\ \vec{x}_i^t, \quad otherwise \end{cases}, \tag{5}$$

where $\Phi(\vec{x})$ represents a fitness function.

3 PSODE Algorithm

In this paper, we propose a new algorithm based on PSO and DE. The original objective is to get benefits form both approaches. The major difference between Differential Evolution and Particle Swarm Optimization is how new individuals are generated. These new individuals generated on each generation are called offspring. It is caused by the selection schemes. Using DE only vectors will be admitted for the following generation that yields a smaller objective function value than the respective target vector. This is called a greedy selection scheme because no deteriorations with regards to the objective function value are possible. In contrast the PSO algorithm accepts all evolved particles, regardless of their objective function value.

In the basic PSO, all individuals are attracted by the best position found by themselves and the whole population. In this way the sharing of information among individuals is only achieved by employing the publicly available information P_g.

Table 1. Pseudocode for the PSODE algorithm

```
Algorithm PSODE
Begin
    Initialize all the populations
    Divide them into tow groups: P1 and P2
    Evaluate the fitness value of each particle
    Repeat
        Do in parallel
        Perform DE operation on P1
        Perform PSO operator on P2
        End Do in parallel
        Barrier synchronization  //wait for all processes to finish
        Select the fittest local individual  from P1
        Select the fittest local individual  from P2
        Determine the global best in the whole population
        Evaluate the fitness value of each particle
    Until a terminate-condition is met
End
```

Therefore, the population may lose diversity and is more likely to confine the search around local minima.

To solve the problem of diversity lose and premature convergence in the basic PSO model, we proposed a hybrid global optimization model. Our approach generates two population offspring individuals, one generated by the PSO mechanism and the other by DE one. There is mutual exchange of best particle information between two populations when they are executed in parallel. The idea behind the proposed algorithm is that the information can be transferred among individuals of different population that will help the individuals to avoid misjudging information and becoming trapped by poor local minima. Table 1 shows the pseudocode of the proposed hybrid algorithm PSODE.

4 Experiment Setting and Benchmark Problems

To investigate the performance of PSODE, we compared it to SPSO and DE in a set of benchmark optimization problems that are commonly used in literature [7, 8]. The benchmark problems used are a set of four non-linear functions, used as minimization problems, which present different difficulties to the optimization algorithms. They are Sphere function, Rosenbrock function, Rastrigrin function and Griewank function.

The parameters used for PSO are recommended from Shi and Eberhart [7]. The maximum velocity $v_{\max}$ and minimum velocity $v_{\min}$ for SPSO were set at half value of the upper bound and lower bound, respectively. The acceleration constants c_1 and c_2 for PSO are both set as 2.0, the inertia weights are set to be $w_{max} = 0.9$ and $w_{min} = 0.4$. The DE parameters used here are $F = 0.8$ and $CR = 0.5$. For PSODE, the parameters, $c_1, c_2, w_{max}, w_{min}, F$ and CR, are all the same with those defined in PSO and DE. The population size is set as 40, and the dimension of the functions is equal to 10. A total of 20 runs for each experimental setting are conducted.

5 Experiment Results

SPSO, DE and PSODE were used to optimize the four benchmark functions using the settings presented in the previous paragraph. The results for the benchmark problem are shown in Table 2. Moreover, Figures 1-4 show the convergence graphs for the benchmark functions. All results below were reported as '0000e+000'.

Table 2. Results for all algorithms on benchmark problems

Function	Results	DE	PSO	PSODE
Sphere	Mean	8.3857e-038	1.8051e-045	2.2864e-051
	Std	3.5160e-074	1.6596e-089	2.3016e-101
Rosenbrock	Mean	2.8662 e+000	1.6916e+000	0.8804e+000
	Std	1.6892 e+000	1.7038e+003	1.2502e+000
Rastrigrin	Mean	2.0894e+000	3.3311e+000	7.9601e-001
	Std	1.4189e+000	2.9464e+000	5.8400e-001
Griewank	Mean	3.8496e-002	3.8822 e-002	3.1011e-002
	Std	7.8891e-004	6.0670 e-004	2.8418e-004

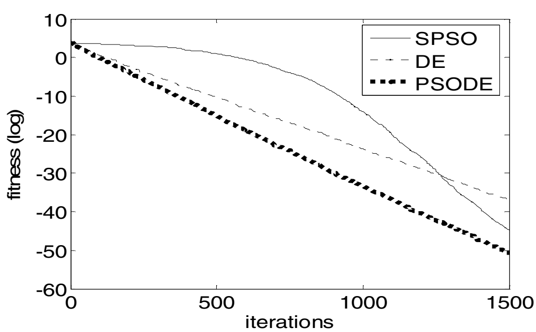

Fig. 1. Convergence graph for Sphere function

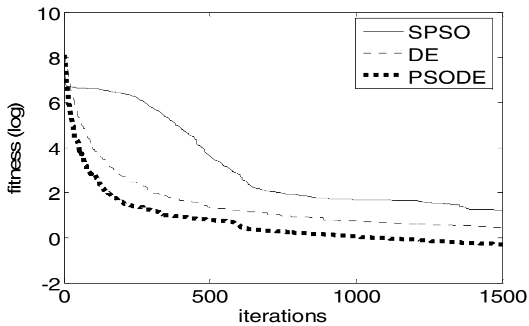

Fig. 2. Convergence graph for Rosenbrock function

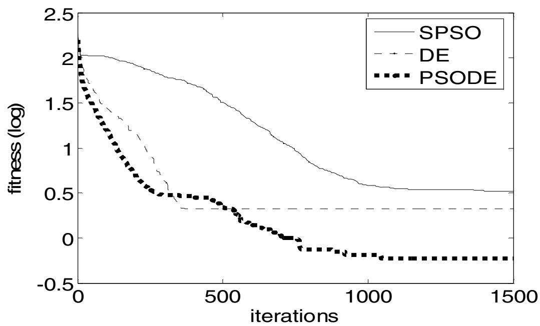

Fig. 3. Convergence graph for Rastrigrin function

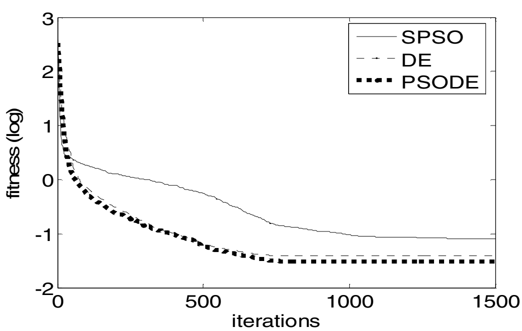

Fig. 4. Convergence graph for Griewank function

For the simplest function Sphere all the algorithms converge exponentially fast toward the fitness optimum. Since those problem is unimodal function, having only a single global minimum, fast convergence to the optimum is not a problem. However, only PSODE had particularly fast convergence, as can be seen from Figure 1. PSO converges slowly, but outperforms DE after 1300 iterations.

On function Rosenbrock, PSODE is superior to both PSO and DE. We may note that DE performs significantly better than PSO.

Function Rastrigrin is a highly multi-modal function when in 10 dimensions or less. Regarding this case DE converges very fast to good values near the optimum, while straggled with premature convergence after 400 generations. PSO performs worse than DE and PSODE. It stagnates and flats out with no further improvement after 1000 generations. On all of algorithms PSODE clearly performs best and gives consistently a near-optimum result.

On function Griewank PSO performs worse than DE and PSODE. PSODE performs nearly the same as DE at the first generations, but outperform DE after 800 generations.

It should be noted that with the PSODE method , the standard deviation of the final solution for 20 trails was found to be significantly low on all of the functions compared with DE and PSO, as show in Table 2. This illustrated that the results generated by PSODE is robust to all of the benchmark problems.

6 Conclusions and Future Work

In this paper a new hybrid optimization algorithm is presented which is based on the integration of PSO and DE. The proposed algorithm is simple in concept and no extra extra parameters are introduced. Four benchmark functions has been used to test PSODE in comparison with PSO and DE. Among them, two functions were unimodal and two were multimodal.

For the multimodal functions PSODE found better results than those generated by the other two methods PSO and DE. For the unimodal functions, of which the convergence rate is more important than the final results, our PSODE outperformed the other two algorithms in terms of accuracy and convergence rate. It can be concluded that by combing the two methods, the advantages of both methods are exploited to produce a hybrid optimization method which is both robust and fast.

Because PSO and DE are executed in parallel in our proposed algorithm, future work is focused on simulating PSODE algorithm by a parallel computer. In addition, different hybrid models of PSO and DE algorithm will be studied.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 70271014.

References

1. Eberhart, R.C., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: 6th IEEE international symposium on Micromachine and Human Science, pp. 39–43. IEEE Press, Piscataway (1995)
2. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceeding of IEEE International Conference on Neural Networks, pp. 194–1948. IEEE Press, Piscataway (1995)
3. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)

4. Kenneth, V.: Price, An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London (1999)
5. Zhang, W.J., Xie, X.F.: DEPSO: Hybrid Particle Swarm with Differential Evolution Operator. In: Proc. of the IEEE International Conference on Systems, Man and Cybernetics, pp. 3816–3821. IEEE Press, Washington (2003)
6. Hendtlass, T.: A Combined Swarm Differential Evolution Algorithm for Optimization Problems. In: Monostori, L., Váncza, J., Ali, M. (eds.) IEA/AIE 2001. LNCS (LNAI), vol. 2070, pp. 11–18. Springer, Heidelberg (2001)
7. Shi, Y., Eberhart, R.C.: A Modified Particle Swarm Optimizer. In: Proceeding of the 1998 IEEE International Conference on Evolutionary Computation, pp. 69–73. IEEE Press, Piscataway (1998)
8. Niu, B., Zhu, Y.L., He, X.X., Wu, H.: MCPSO: A Multi-Swarm Cooperative Particle Swarm Optimizer. Applied Mathematics and Computation 185, 1050–1062 (2007)