Applied Mathematics & Information Sciences 5 (2) (2011), 33S-38S
– An International Journal

Particle Swarm Optimizer with Time-Varying Parameters based on a Novel Operator

R. Cheng[1] and M. Yao[2]
[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
Email Address: myao@zju.edu.cn

Received June 22, 2010; Revised December 21, 2010

This paper proposes a time-varying particle swarm optimizer based on our earlier work which introduces a novel operator (leap operator). Two new parameters are recommended in leap operator to prevent premature convergence. With these two parameters, a new modification named LPSO is constructed. Since the values of the 2 parameters are not easy to determine, in this paper, they are modified as time-varying ones. With the time-varying parameters, the modified particle swarm optimizer (TVLPSO) has good potential in finding better solutions. Compared with standard PSO and LPSO, benchmark tests are implemented.

Keywords: particle swarm optimizer, time-varying.

1 Introduction

Particle swarm optimization is a nature inspired optimization technique which was originally introduced in [1]. It was initially found working well in the field of linear function optimization. However, in later research, the standard PSO is shown to posses no ability to perform a fine grain search to improve the quality of solutions as the number of iterations is increased, although it may find the near optimal solution much faster than other evolutionary algorithms [3]. This is considered to be caused by premature convergence which only provides solutions of poor quality. So in our earlier work, a modified particle swarm optimizer (LPSO) with a novel operator (leap operator) is introduced to prevent premature convergence.

In LPSO, 2 new parameters ρ and δ are introduced [2]. Although with empirically recommended values for them, LPSO successfully prevents premature convergence and improves the quality of solutions to some degree, however, further study on ρ and δ are still required. In this paper, a time-varying δ is constructed alone with ρ. With this modification, the quality of solutions is further improved when it comes to the optimization of some benchmark test functions. The new modification is referred to as TVLPSO (Time-Varying Leap-PSO).

2 PSO and LPSO

In the standard PSO method [12], the information of each particle i in the swarm is recorded by the following variables: (i) the current position X_i, (ii) the current velocity V_i, (iii) the individual best position $pbest_i$, and (iv) the swarm best position $gbest$. In each iteration, the positions and velocities are adjusted by the following equations:

$$\begin{aligned} v_{ij}(t+1) = wv_{ij}(t) + c_1 rand()[p_{ij}(t) - x_{ij}(t)] \\ + c_2 Rand()[p_{gj}(t) - x_{ij}(t)] \end{aligned} \tag{2.1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{2.2}$$

for $j \epsilon 1,2,\ldots d$ where d is the dimension number of the search space, for $i \epsilon 1,2,\ldots n$ where n is the number of particles, t is the iteration number, w is the inertia weight, $rand()$ and $Rand()$ are random numbers uniformly distributed in the range $[0, 1]$, c_1 and c_2 are accelerating factors.To control the flying step size of the particles, v_{ij} is constrained in the range $[\vee v_{max}, v_{max}]$ where v_{max} is commonly set as 10%--20% of each search dimension size [10].

Fast convergence of standard PSO sometimes only provides a poor solution. Such inferior convergence is called premature convergence. To prevent premature convergence, a modification referred to as LPSO with a novel operator is proposed in [2]. The novel operator (leap operator) is based on two *hypotheses*:

Hyp. 1. In the terminal iterations, if the global best position $gbest$ is not updated for a number of consecutive iterations, the near optimal area is always almost located and the algorithm tends to be trapped in premature convergence.

Hyp. 2. The worst-fitting particle has the least probability to reach the global optimal solution.

When this operator works, a particle leaps from one position to another, that's why we refer it as *leap operator*. The key mechanism of leap operator is presented by the following formulas:

$$(t \geq \rho * maxiter) \wedge (C \geq \delta) \tag{2.3}$$

where ρ and δ are thresholds. The first half of (2.3) determines in which period of a run process the leaps are performed and the second one determines the leap frequency. t is the current iteration number and C is a variable which records the number of consecutive iterations during which $gbest$ is not updated.

$$X_l = gbest \tag{2.4}$$

$$x'_{lk} = x_{lk} + offset \tag{2.5}$$

In (2.4), X_l is the position of the particle selected to leap. In (2.5), a stochastic $offset$ is added to one of the dimensions of X_l. The initial velocity is assigned with 0 which is presented by

$$V_l = 0 \tag{2.6}$$

ρ and δ are the only parameters which require priori knowlege for initialization. In [2], only empirically recommended values are given, $[0.4, 0.6]$ for ρ and 5 for δ respectively.

3 Modification with time-varying Parameters: TVLPSO

In LPSO, ρ and δ require priori knowledge to initialize. Firstly, such knowledge is not easy to access. Secondly, even though we have the knowledge, it's still difficult to construct a universal model which deals with the relationship between the knowledge and the 2 parameters. So we try not to give the values directly, but we can make them vary during the algorithm's run process. It provides more flexible parameter values, thus reducing the possible impact a bad parameter value may have on a run process.

In [2], variation is added only in the terminal iterations to prevent premature convergence. However, in the early iterations, no variation exists. So here we add some variation in the early iterations to ensure a more widespread exploration. Such variation is adaptive to time. Specifically, the more iterations are implemented, the more variation should be added. This is based on hyp.1. So that a time-varying model is constructed:

$$\delta(t) = \begin{cases} (\delta_i - \delta_f)(\frac{maxiter-t}{maxiter}) + \delta_f & \text{if } t \leq \rho * maxiter \\ \delta_f & \text{otherwise} \end{cases} \tag{3.1}$$

where δ_i is the initial value for δ and δ_f is the final one. $maxiter$ is the maximum iteration number and t is the current iteration.

Besides, a time-varying inertia weight w proposed by Shi and Eberhart [9] is used. The mathematical presentation is as follows:

$$w(t) = (w_i - w_f)(\frac{maxiter - t}{maxiter}) + w_f \tag{3.2}$$

Furthermore, similar time-varying acceleration coefficients introduced in [11] are also used. The mathematical presentation is as follows:

$$c_1(t) = (c_{1i} - c_{1f})(\frac{maxiter - t}{maxiter}) + c_{1f} \tag{3.3}$$

$$c_2(t) = (c_{2i} - c_{2f})(\frac{maxiter - t}{maxiter}) + c_{2f} \tag{3.4}$$

With (3.1) to (3.4), more variation is added to LPSO. Since it's time-varying and based on LPSO, we refer it as TVLPSO (Time-Varying Leap-PSO).

4 Experimental results and statistical analysis

In our experiments, four standard functions widely used in genetic and evolutionary algorithms' tests [4–8] etc. for benchmark tests are selected. For each function, the dimension number is 30 and for each dimension I, xi ϵ [-100,100]. Each function's global optimal solution is 0. They are Sphere(De Jong F1) function, Rosenbrock function, Rastrigin function and Griewank function presented by (4.1), (4.2), (4.3) and (4.4) respectively

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{4.1}$$

$$f_2(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \tag{4.2}$$

$$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{4.3}$$

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{4.4}$$

First, an experiment aiming to determine the values of δ_i and δ_f is implemented. In this experiment, δ_f is assigned with 5 which is the recommended value of δ in [2]. For δ_i, several values are tested (see Table 4.1). c_i, c_f, w_i and w_f are all assigned with values recommended in [9] and [11] ($c_{1i} = 2.5, c_{1f} = 0.5, c_{1i} = 0.5, c_{1f} = 2.5, w_i = 0.9, w_f = 0.4$). Other parameters are all assigned with values recommended in [9] and [2] ($0.4 \leq w \leq 0.9$, $c_1 = c_2 = 2$, $v_{max} = 4$, $\rho = 0.4$). The $maxiter$ is 5000.

For each test, 30 trial runs are implemented. To show the performance of TVLPSO, LPSO and PSO are tested alone with it. All common parameters among them are assigned with the same recommended values. The statistical analysis of the experimental result is also in Table 4.1. For f_1 and f_2, TVLPSO performs the best. For f_3, the average value is slightly worse than that of LPSO. For f_4, PSO performs the best. Fig. **??** and Fig. 4.1 are the box plots of the statistical analysis. They give a visual and more detailed presentation.

5 Conclusion

Based on our earlier work [2], an enhanced modified particle swarm optimizer with time-varying parameters is introduced in this paper. Since PSO is sensitive to its parameter values, experiments are implemented to discuss the values for the new parameters. With benchmark tests, comparison is made with PSO and LPSO. The experimental results demonstrate that TVLPSO has good potential in finding better solutions. However, only numeric tests are implemented. Maybe in future work, more tests can be implemented on various problems.

Table 4.1: Statistical analysis of the experimental results.

Function	Statistical Analysis	PSO	LPSO	TVLPSO with different δ_i values			
				$\delta_{i1} = 25$ $\delta_{f1} = 5$	$\delta_{i2} = 50$ $\delta_{f2} = 5$	$\delta_{i3} = 100$ $\delta_{f3} = 5$	$\delta_{i4} = 200$ $\delta_{f4} = 5$
f_1	Average	1.2711e-77	1.5658e-77	1.8855e-76	4.3950e-77	**8.9710e-78**	6.4752e-77
	Min	3.3278e-81	3.4363e-82	3.7252e-81	7.8563e-82	**4.3246e-82**	4.3246e-82
	Max	3.1046e-76	5.0794e-76	6.4849e-75	1.4362e-75	**1.7292e-76**	7.3587e-76
	SD	4.9074e-77	7.2302e-77	9.3501e-76	2.1087e-76	**2.5666e-77**	1.6462e-76
	CV	3.8608	4.6174	4.9590	4.7979	2.8610	2.5423
	Median	8.8512e-79	3.7822e-79	7.3145e-79	9.9422e-79	1.2843e-78	3.4268e-78
f_2	Average	33.5720	37.5856	**26.9554**	33.4086	29.8224	33.6982
	Min	0.3036	2.1792	**0.0270**	0.8700	0.8158	3.1889
	Max	95.5984	119.9976	93.7678	117.9441	86.3541	87.4896
	SD	28.8191	21.3229	**24.6676**	27.7500	24.5104	25.5365
	CV	0.8584	0.8003	0.9151	0.8306	0.8219	0.7578
	Median	33.3311	3.9798	**20.0846**	21.7383	20.6541	21.0453
f_3	Average	32.7938	**4.2584**	4.6972	4.6166	**4.5768**	4.6565
	Min	15.9193	**0.9950**	**0.9950**	**0.9950**	**0.9950**	3.8368
	Max	51.7378	10.9445	14.9244	11.9395	**9.9496**	12.9344
	SD	8.2707	2.0305	3.1076	2.3074	2.3355	2.7373
	CV	0.2522	0.4768	0.6616	0.1998	0.5103	0.5879
	Median	33.3311	3.9798	3.9798	3.9798	**3.9798**	4.9748
f_4	Average	**0.0086**	0.0089	0.0126	0.0124	0.0096	**0.0090**
	Min	**0**	**0**	**0**	**0**	**0**	**0**
	Max	**0.0270**	0.0295	0.0393	0.0467	0.0418	**0.0394**
	SD	**0.0068**	0.0078	0.0102	0.0088	0.0109	**0.0081**
	CV	0.7935	0.8745	0.8099	0.7116	1.1416	0.8990
	Median	**0.0074**	**0.0074**	0.0099	0.0123	0.0074	**0.0074**

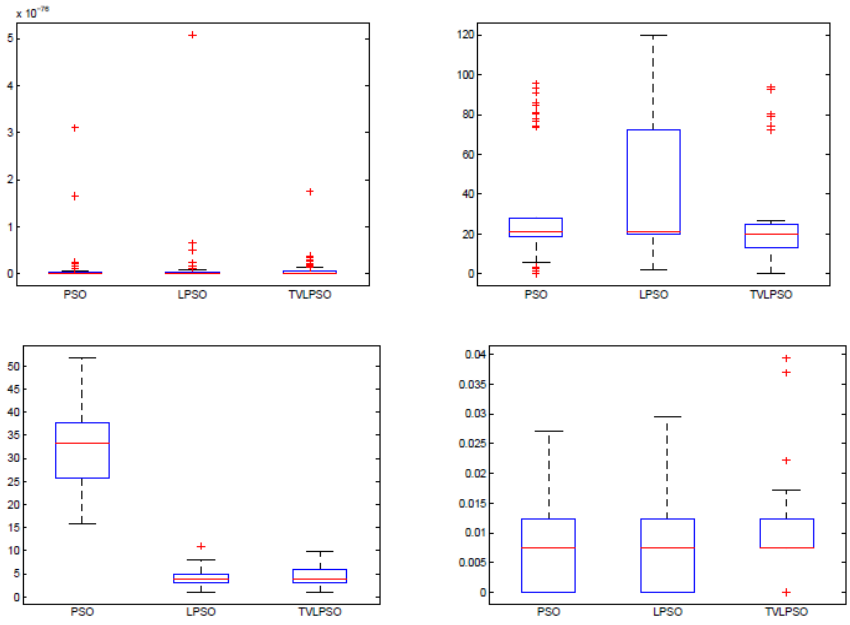

Figure 4.1: The box plots of experimental results on f_1 (δ_{i3}), f_2 (δ_{i1}) ,f_3 (δ_{i3}) and f_4 (δ_{i4}).

References

[1] J. Kennedy and R. C. Eberhart, Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, 1995.
[2] Ran Cheng and Min Yao, A modified Particle Swarm Optimizer with a Novel Operator, AICI 2010, Part II, LNAI 6320, pp. 293–301, 2010.
[3] P.Angeline, Evolutionary optimisation versus particle swarm optimization: Philosophy and performance difference,In Proc. of Evolutionary Programming Conference, San Diago, USA, 1998.
[4] Suganthan P N, Particle swarm optimizer with neighborhood operator, Proc of the Congress on Evolutionary Computation, pp. 1958–1962, 1999.
[5] Angeline P J, Using selection to improve particle swarm optimization, Proc IEEE Int Conf on Evolutionary Computation, Anchorage, pp. 84–89, 1998.
[6] Clerc M, The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization, Proc of the Congress on Evolutionary omputation,Washington DC, pp. 1951–1957, 1999.
[7] Kennedy J, Stererotyping: Improving particle swarm performance with cluster analysis, Proc IEEE Int Conf on Evolutionary Computation, pp. 1507–1512, 2000.
[8] Qian-li Zhang, Xing Li, Quan-an Tran, A modified particle swarm optimization algorithm, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 2005.
[9] Shi, Y.H. and Eberhart, R.C., Parameter selection in particle swarm optimization, in Eiben A, Porto V, Saravanan N, Waagen D. (Eds), Evolutionary Programming VII. San Diego, California,USA: Springer-Verlag, pp. 591–600, 1998.
[10] R.C. Eberhart, Y. Shi, Particle swarm optimization: developments, applications and recourses, Proc of the IEEE Int Conf on Evolutionary Computation, vol. 1, pp. 81–86. Seoul (2001).
[11] Asanga Ratnaweera, Saman K. Halgamuge, Harry C. Watson, Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients, IEEE Transactions on Evolutionary Computation, Vol.8, No.3, June, 2004.
[12] Mohammed EI-Abd, Preventing premature convergence in a PSO and EDA hybrid, in IEEE Congress on Evolutionary Computation, pp. 3061–3066, 2009.

Ran Cheng is a Ph.D. student at the College of Computer Science and Technology, Zhejiang University, Hanghou, China. His research fields are particle swarm optimiztion, computational intelligence and multi-objective optimization. In Fall 2010, Mr. Cheng obtained, with honors, a B.Sc. at the College of Information Science and Engineering, Northeastern University, Shenyang, China.

Min Yao received the Ph.D. degree in Biomedical Engineering and instrument from Zhejiang University, China, in 1995. He is currently a professor in the College of Computer Science and Technology at Zhejiang University, Hangzhou, China. His research interests include computational intelligence, pattern recognition, knowledge discovery and knowledge service.