Improved Particle Swarm Optimization algorithm in dynamic environment

Conference Paper · May 2014

DOI: 10.1109/CCDC.2014.6852707

3 authors:

Changcheng Xiang
Hubei University for Nationalities

Xuegang Tan
Southeast University (China)

Yi Yang

Improved particle swarm optimization algorithm in dynamic environment

Changcheng Xiang[1,2,3], Xuegang Tan[3], Yi Yang[3]

1. Key Laboratory of Biologic Resources Protection and Utilization ,Hubei Minzu University, Enshi,Hubei,445000,P.R. China
E-mail: xcc7426681@126.com

2. College of Forestry and Horticulture, Hubei Minzu University,Enshi,445000,China

3. College of Science, Hubei Minzu University,Enshi,445000,China

Abstract: In this paper,The improved Particle Swarm Optimization in dynamic objective function environment(DOFPSO) is purposed. The dynamic environment will change with the time t. The DOFPSO algorithm discuss that how to determine changes of the time(environment) and how to keep population diversity. The improved algorithm has the ability to fast response the change of environment and could find the best fitness value quickly. The results of experiment indicate that DOFPSO is more effective than particle swarm optimization(PSO) and restart method particle swarm optimization(RMPSO) in the response of change of environment and fast convergence.

Key Words: Particle Swarm Optimization;Dynamic Environment;Convergence

1 Introduction

Most research in evolutionary computation focuses on static optimization problems. However, Many optimization problems in real life are more complex than static state optimization ,and are subject to changing conditions over time, thus optimization in dynamic environments(DE) is a challenging and important task for us. The dynamic optimization problems (DOPs) requires optimization algorithms to not only find the optimal solution in a short time but also track the changing optimal solution over time when the problem changes. Hence, we need to find an adaptive algorithm to search the optimal solution to a changing environment. Many intelligent algorithms are widely used to deal with optimization problems in DE, such as Evolutionary algorithms (EA)[1], genetic algorithm(GA)[2], Immune Algorithm(IA)[3] and etc. However, These algorithm have the disadvantages that solving DE problems is the loss of genetic diversity . To overcome these disadvantages, we should increase the search space to improve the optimal solution. Particle swarm optimization(PSO) are the swarm intelligent stochastic optimization technique, which was designed by Kennedy and Eberhart [4, 5, 6] firstly, and which originated from bird's searching food. PSO has been proved well to many static optimization. The fitness value(objective function) is to evaluate the convergence of the PSO. In other words, the optimal value is not changed, we think that the PSO converges to the best solution in static optimization. However, for the DOPs, the fitness value may be changed with DE.

The work is supported by the fourth Open funfation of Key Laboratory of Biologic Resource Protection and Utilization of Hubei Province (No.PKLHB1332), the natural science foundation of Hubei Province (No.CDZ2010047), and the soft science research project of Hubei Province (No.2012GDA01309).

We known that PSO has great ability to search best fitness value in static state optimization, but it can easily be trapped in a local optimum value when environment changed. So, we should improve the basic PSO algorithm for DOPs. For DOPs , there exist two problems to be solved, one is how to determine changes of the time(environment), the other is how to keep population diversity. In papers[7, 8], the time point was selected to change and some worst particles are initialized again when the environment changed. That is to say that position and velocity of particles is initialized again to keep the population diversity. Blackwell T [9] and Dou Quansheng [10] proposed swarm-core evolutionary particle swarm optimization which is particle swarm are divided into three subswarms and each sub-swarm has different job to keep population diversity. In order to test the environment change, Hu and Eberhart detect the global best value whether changed for several generations[5, 7], likes impulse. Obviously, the method can't response to environmental change timely and accurate. we should design the new method to detect the changed time.

In this paper, we discuss the condition of environmental change which caused by fitness value change with time and improve PSO algorithm to keep the population diversity. Generally speaking, The fitness value of best particle will change when objective function changed. So based on the feature, we check the fitness value of global best particle. If the fitness value of the best particle at the time t is equal to the fitness value at the next time $t+1$, the particles keep the same environment, otherwise it changed. The results show that the method can fast and accurately respond the change of environment. In the first part, we discuss the DOPs briefly and the improved PSO algorithm for Dops. the time changed point (environment) and population diversity are discussed in detail. In the second part, we revised the Generalized Griewank Function as dynamic optimization function, the simulation results show that the improved PSO algorithm has more effective to response of the environment change.

2 Dynamic Optimization and Improve Modified Particle Swarm Optimization

2.1 Dynamic Optimization Problem

The dynamic optimization problems (DOPs) requires optimization algorithms to not only find the optimal solution in a short time but also track the changing optimal solution over time when the problem changes. The objective function would change with optimization procedure[11]. Many problems in real word can be modeled with dynamic optimization problem. Generally, the dynamical optimization problem can be defined as follows[12]:

$$DOP = \begin{cases} optimize & f(x,t), \\ s.t. & x \in F(t) \subseteq X, t \in T \end{cases} \quad (1)$$

where:
- $X \in R^n$, X is the search space.
- t is the time.
- $f : X \times T \to R$, is the objective function that assigns a numerical value $(f(x,t) \in R)$ to each possible solution $(x \in S)$ at time t.
- $F(t)$, is the set of feasible solutions $x \in F(t) \subseteq S$ at time t.

In other words, a DOPs is a problem where the objective function or the restrictions change with time. In this paper, we discuss the problem which caused by the target function changed with time t.

2.2 Standard particle swarm optimization

In static state optimization, the potential solutions, which called particles, are then flown through the search space by following the current optimum particles. The updates of particles based on the following equations[13, 14]:

$$\begin{aligned} v_i(k+1) &= w \times v_i(k) + c_1 \times r_1 \times (p_{id}(k) - x_i(k)) \\ &+ c_2 \times r_2 \times (p_{gd}(k) - x_i(k)) \\ x_i(k+1) &= x_i(k) + v_i(k+1) \end{aligned} \quad (2)$$

Equation (2) calculates a new velocity($v_i(k+1)$) of each particle which based on the particle's previous velocity($v_i(k)$), the particle's position at which the best fitness value has been achieved(p_{id}, or $pBest$) up to now, and the population global location(p_{gd}, or $gBest$) at which the best fitness value of gloab has been achieved at this point. w is the inertia weight, c_1, c_2 are positive constants, and r_1, r_2 are random numbers in the range of (0, 1). Equation (3) updates every particle's position based on the particle's previous position($x_i(k)$) in search space and it's the new velocity($v_i(k+1)$). Search space dimensionality is D, and the ith particle's position is $x_i = (x_{i1}, x_{i2}, ..., x_{ik}, ...x_{iD})$, here, $X_{min} \le x_i \le X_{max}$.

2.3 Improved particle swarm optimization in dynamical Environment

Arming at the Characteristics of standard particle swarm optimization (PSO) is easy to fall into local optimum when target function changed with time t, and restart method particle swarm optimization (RMPSO) couldn't keep previous memory of particles, we put forward the improved particle swarm optimization algorithm in dynamic objective function environment(DOFPSO). The steps of algorithm as follows:
- Step 1: It was set that the search space$[X_{min}, X_{max}]$, rang of particles velocity $[V_{min}, V_{max}]$, initial time t_0, and numbers of particles N and so on.
- Step 2: We initialize the position and velocity of particles with the under equation:

$$x_i(0) = (X_{max} - X_{min}) \times rand() \quad (3)$$

$$v_i(0) = (V_{max} - V_{min}) \times rand() \quad (4)$$

The initial position $x_i(0)$ was treated as the ith particle's local best position, and the fitness value of it was looked as the ith particle's local best fitness value. We find the best value from all particles' fitness values as the F_{pg} called the global best fitness value, and the corresponding position is recognized as p_g which is the population global optimal location.
- Step 3: It would be accessing to the main iterative loop and updating the position and velocity of particle with equation (2),(3).
- Step 4: In order to keep the particles "running" in the search space, we should judge the particles' position. If particles' position are beyond the rang of search space as well as velocity out of the rang of speed, the position and the velocity are Calculated as follows:

$$\begin{aligned} x_i(k) &= X_{max} \times (2 \times rand() - 1); \\ & if \quad x_i(k) > X_{max} or x_i(k) < X_{min} \\ v_i(k) &= V_{max} \times (2 \times rand() - 1); \\ & if \quad v_i(k) > V_{max} or v_i(k) < V_{min} \end{aligned} \quad (5)$$

- Step 5: We find the best local position of ith particle (p_{id} or $pBest$), and find the best global fitness value(F_{pg}), which corresponding position called The global optimal position(p_{gd} or $gBest$).
- Step 6: In order to detect and respond the environment change caused by objective function with time t, we set a accuracy ε. When variance of best fitness value passed the ε, using the p_g we have found recently to detective whether the environment changed. if $f(p_g, t_0)$ is equal to $f(p_g, t_1)$, we hold that the environment not change; Otherwise, it changed. To clarify are two point: firstly, the parameter t_0 is the upper time of environment and t_1 is now; secondly, the variance calculation formula based on the following equations[15, 16]:

$$\overline{F_{pg}} = \frac{1}{n} \sum_{i=1}^{n} F_{pg}(i) \quad (6)$$

$$varFpg = [\sum_{i=1}^{n} (Fpg(i) - \overline{F_{pg}})^2]^{1/2} \quad (7)$$

Here, n is iteration of recent iterate n times, $F_{pg}(i)$ is the ith best global fitness value of n best value we found recently.

- Step 7: We should check out the population diversity, when the environment changed. It is set a region which take for p_g as circle center. The radius of region is R. It was though that the particle in the region, if the norm $r = \|x_i - p_g\|$ less than R. We change the velocity of particles in the region and make them get to other ground of space, if the particles' number in the region are beyond upper limit $\rho N (0 < \rho < 1)$ we set. Them run as follow:

$$v_i(k+1) = p_g - x_i(k) \tag{8}$$

$$x_i(k+1) = x_i(k) + s \times v_i(k+1) \tag{9}$$

Here, s is escape multiple. When particles trap into local optimum, it run away p_g with speed of s times. sectionDOFPSO Algorithm Experiment

3 The Result of Experiment

To evaluate performance of DOFPSO which we modified, we chose Generalized Griewank Function and remoulded it, which have parameter about t, as objective function. It listed as

$$f(x,t) = \frac{1}{4000}\sum_{i=1}^{D}(x_i^2 \sin(t)) - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{t}}) \tag{10}$$

Here, in order to find the minimum value of target function, we set search range dimensionality $D = 2$; search range upper $X_{max} = [10, 10, ..., 10]$, search range low $X_{min} = [-10, -10, ..., -10]$; velocity upper $V_{max} = [10, 10, ..., 10]$, velocity low $[-10, -10, ..., 10]$; number of particles $N = 60$; the inertia weight $w = 0.65$; radius of region which take for p_g as circle center $R = \frac{1}{1000}\|X_{max} - X_{min}\|$; positive constants $c_1 = 0.85$, $c_2 = 5.0$, and the accuracy $\varepsilon = 10^{-5}$.
In this section, some results are presented to illustrate the effectiveness and response of the proposed optimization based on environment change with time t. In the table 1, we list three algorithms' environment change point of time.

We can find that PSO is easy to be trapped into local optimization and no response about the environment change. Although,the RMPSO could solve the response environment problem, the convergence rate and accuracy dissatisfied us. In the table 2, listed the fitness value of three algorithms at environment change points. In figure 1 and 2, DOFPSO's fitness value compared with PSO and RMPSO .

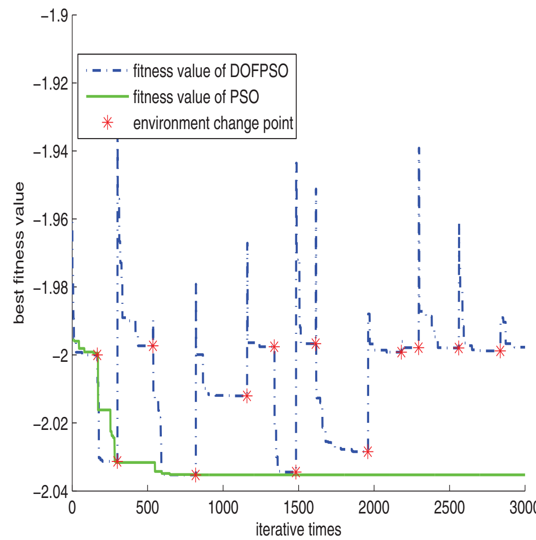

Figure 1: The best fitness values of DOFPSO compared with the best fitness values of PSO.

The particle diversity of algorithm is very important, it keeps search ability about the best fitness value. Mean fitness value in each generation could reflect the particle diversity of algorithm. So we used it to observe the particle diversity of algorithm. The optimal fitness average value is calculated as follows:

table **1** The response of three algorithms about environment change with time

Algorithms	Change point of environment												
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}
PSO	Y	Y	-	-	-	-	-	-	-	-	-	-	-
RMPSO	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
DOFPSO	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

NOTE: - No response to environment change, Y Response to environment change.

table **2** The fitness value of three algorithms at environment change points

Algorithms	**Environment change points of iterative**												
	166	299	535	818	1158	1339	1482	1613	1959	2181	2295	2561	2836
PSO	-2.00	-2.02	-2.03	-2.03	-2.03	-2.03	-2.03	-2.03	-2.03	-2.03	-2.03	-2.03	-2.03
RMPSO	-1.99	-2.03	-1.99	-2.03	-2.01	-1.99	-2.03	-1.99	-2.02	-1.99	-1.99	-1.99	-1.99
DOFPSO	-1.99	-2.03	-1.99	-2.04	-2.01	-1.99	-2.03	-1.99	-2.03	-1.99	-1.99	-1.99	-1.99

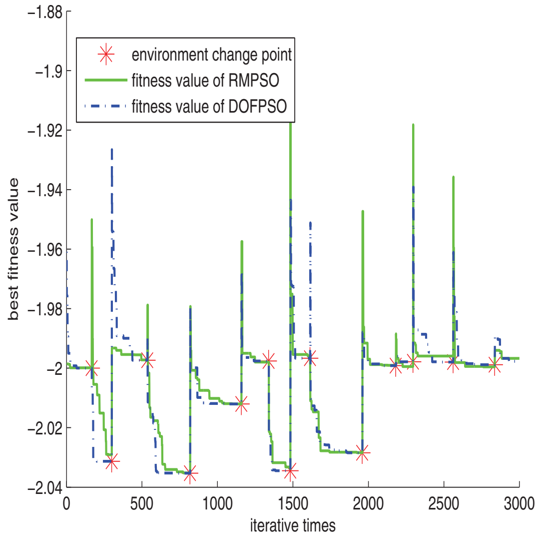

Figure 2: The best fitness values of DOFPSO compared with the best fitness values of RMPSO.

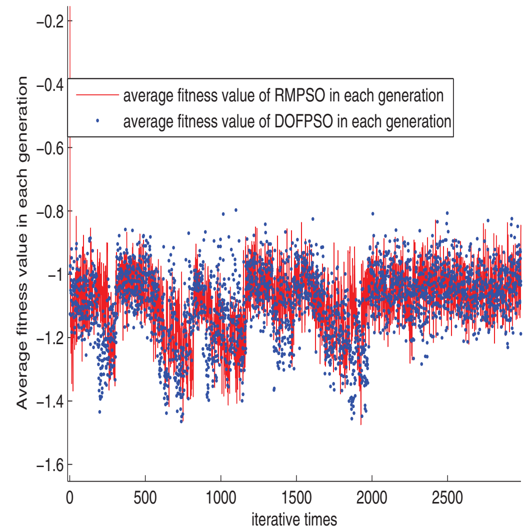

Figure 4: The average fitness values of DOFPSO in each generation compared with the average fitness values of RMPSO in each generation.

$$\overline{f} = \frac{1}{N}\sum_{i=1}^{N}(f(x_i,t)) \tag{11}$$

In figure 3 and 4, we used mean fitness value in each generation of DOFPSO compared with the average fitness values of PSO and RMPSO. Figure 5 list fitness average value in each generation of three algorithm and the best fitness value of DOFPSO.

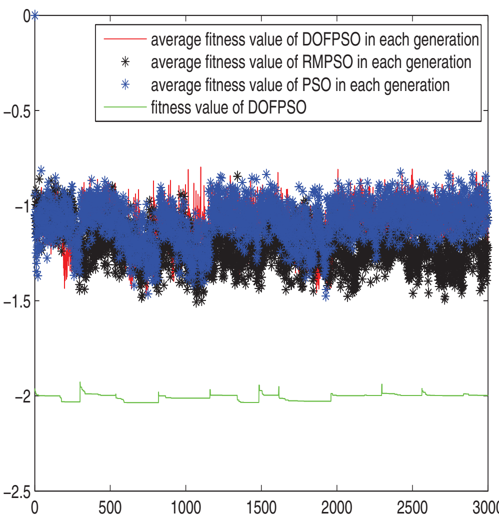

Figure 5: The average fitness values of three algorithm in each generation and best fitness value of DOFPSO.

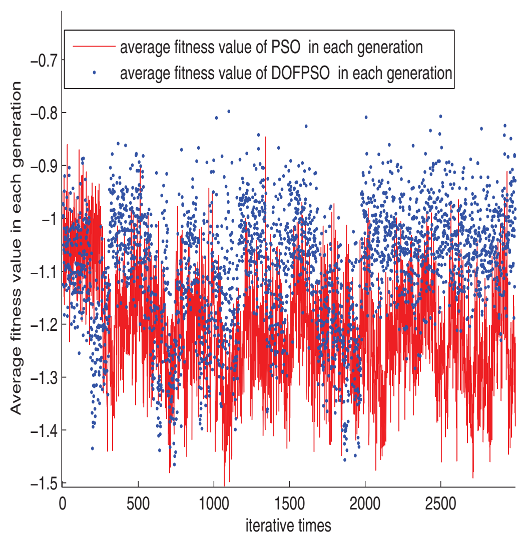

Figure 3: The average fitness values of DOFPSO in each generation compared with the average fitness values of PSO in each generation.

4 Conclusions

In this paper, we find that the improved particle swarm optimization algorithm in dynamic objective function environment (DOFPSO) has the more effective , which could be response the environment change caused by objective function with time and could be converged fast. Compared with Standard particle swarm optimization(PSO), it turned out that DOFPSO could be more sensitive in response of environment change and keep diversity of particles better than PSO. Of course, it is faster convergence and greater accuracy than restart method particle swarm optimization(RMPSO) and keep the diversity of particles as well as RMPSO.

REFERENCES

[1] J. Branke. Evolutionary optimization in dynamic environments. Kluwer Academic Publishers, Massachusetts USA,

2002
[2] Ursem R K. Multinational GAs: Multimodal Optimization Techniques in Dynamic Environments[C]//GECCO. 2000: 19-26.
[3] de Franca F O, Von Zuben F J, de Castro L N. An artificial immune network for multimodal function optimization on dynamic environments[C]//Proceedings of the 2005 conference on Genetic and evolutionary computation. ACM, 2005: 289-296.
[4] Eberhart, R. and Kennedy, J. A New Optimizer Using Particles Swarm Theory, Proc. SiXth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, 1995: 39-43.
[5] Kennedy, J. and Eberhart, R. Particle Swarm Optimization, IEEE International Conference on Neural Networks(Perth, Australia), IEEE Service Center, Piscataway, NJ, IV, 1995: 1942-1948
[6] Shim, Y. H., Eberhart, R. C. Parameter Selection in Particle Swarm Optimization, The 7th Annual Conference on Evolutionary Programming, San Diego, USA, 1998.
[7] Hu X, EberhartR. C., Adapticle swarm optimization: Detection and response to dynamic systems[C], Proceedings of the IEEE Intenational Conference on Evolutionary Computation, Honolulu, Hawaii, USA. Piscataway, NJ.: IEEE Press, 2002: 1666-1670.
[8] Zhan Di, Designing Restart Strategy for Randomized Algorithms and Its Applicationin Solving the TSP. CHINESE J. COMPUTERS, vol.25, No.5, May 2007:514-519.
[9] T Blankwell, J Branke. Multi-swarm optimization in dynamic environments[C]. In: Proc of Applications of Evolutionary Computing, LNCS 3005. Berlin: Springer-Verlag, 2004: 489-500.
[10] Dou Quansheng, Zhou Chunguang, Xu Zhougyu, et al. Swarm-core evolutionary particle swarm optimization in dynamic optimization environments [J]. Journal of Comoputer Research and Development, 2006, 43(1): 89-95 (in Chinese).
[11] Y Jin, J Branke. Evolutionary optimization in uncertain environmentsłA survey[J]. IEEE Trans on Evolutionary Computation, 2005, 9(3): 303-217.
[12] CruzC, Gonz lez J R, Pelta D A. Optimization in dy-namic environments: A survey on problems, methods and measures. Soft Comput(2011) 15, Springer, pp: 1247-1248.
[13] Shi, Y. and Eberhart, R. C. A modified particle swarm optimizer. Proceedings of the IEEE International Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1998, pp:69-73.
[14] Clerc, M., (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proceedings, ICEC, Washington, DC, USA.
[15] Peng ShiGe. Survey on normal distributions, central limit theorem, Brownian motion and the related sto chastic calculus under sublinear expectations. Science in China Series A: Mathematics. Jul, 2009, Vol. 52, No. 7: 1391C1411.
[16] K. Weicker, Performance measures for dynamic environments, in Parallel Problems Solving from Nature , ser. LNCS 2439. Berlin, Germany: Springer-Verlag, 2002, pp: 64-73.