Proceedings of the 2009 IEEE
International Conference on Robotics and Biomimetics
December 19 -23, 2009, Guilin, China

Particle Swarm Optimization Algorithm with Asymmetric Time Varying Acceleration Coefficients

G. Q. Bao and K. F. Mao

***Abstract*—To update the performance of the standard optimization method, a modified particle swarm optimization (MPSO) algorithm is proposed based on the earlier works. An asymmetric time-varying acceleration coefficients adjusting strategy introduced in this paper is to keep the balance between the global search and the local search with the great advantages of convergence property and robustness compared with basic PSO algorithm. The relationship between swarm average velocity and convergence is studied through Benchmark test functions simulation. All the merits mentioned above are demonstrated by the compound gear transmission ratio optimization in transmission systems.**

I. Introduction

The basic particle swarm optimization (PSO) is a population-based optimization method first proposed by Kennedy and Eberhart [1]. Some of the attractive features of PSO include the ease of implementation and the fact that no gradient information is required. This allows the PSO to be used on functions where the gradient is either unavailable or computationally expensive to obtain.

It is clear that there will always be a need for better optimization algorithms, since the complexity of the problems that we attempt to solve is ever increasing. In recent years, many scientific, engineering and economic optimization problems, such as minimizing the losses in power grid by finding the optimal configuration of the components, training a neural network to recognize images of people's face, and reducing the cost of a transverse flux permanent magnet machine, involves to optimize a set of parameters. Numerous optimization algorithms have been proposed to solve these problems with a certain degrees of success [2,7,8,10].

The PSO algorithm is becoming very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution. In this paper, some recent approaches for solving optimization problems through PSO are presented. After its historical background introduction in Section II, an analytic description of the concepts behind PSO is studied in Section III. An improved PSO (MPSO) algorithm based on asymmetric time varying acceleration coefficients c_1 and c_2 is described in Section IV. The benchmark problems and parameter settings are also outlined. Further in section V the application to gear train design is discussed. The experimental results verify the performance of MPSO is sensitive to the c_1 and c_2 and the asymmetrical variation of c_1 and c_2 can guarantee both in the solution quality and algorithm robustness. Finally, in section VI, conclusions and future work are discussed.

Manuscript received July 15, 2009. This work was supported by the National Natural Science Foundation of China (No. 50877034) and the Teaching and Research Award Program for Innovation Talents in Gansu Province of China.

G.Q. Bao and K.F. Mao are with the Electrical Engineering Department, Gansu University of Technology, China (phone: 0086-13993183555; fax: 0086-931-2975382; e-mail: Guangqing.bao@ gmail.com).

II. Particle Swarm Optimization

A. The Particle Swarm Optimization Algorithm

The origins of the PSO were based on the sociological behavior associated with bird flocking. The algorithm maintains a population of particles, where each particle represents a potential solution to an optimization problem. Let D be the size of the swarm. Each particle i can be represented as an object with several characteristics. These characteristics are assigned by the following symbols:

-- x_i: The current position of the particle;
-- v_i :The current velocity of the particle;
-- p_i :The personal best position of the particle i;
-- p_g :The best position discovered by any of the particle so far.

The personal best position associated with particle i is the best position that the particle has visited, yielding the highest fitness value for that particle. For a minimization task, a position yielding a smaller function value is regard as having a higher fitness. The symbol f will be used to denote the objective function that is being minimized. The update equation for p_i and p_g is represented in (1) and (2) respectively, with the dependence on the time step t made explicit.

$$p_i(t+1)=\begin{cases} p_i(t) & \text{if } min\{\ f(x_i(t+1))\ \}\geq f(p_i(t)) \\ x_i(t+1) & \text{if } min\{\ f(x_i(t+1))\ \}\leq f(p_i(t)) \end{cases} \quad (1)$$

$$p_g(t)=min\{f(p_o(t)), f(p_1(t)),..., f(p_D(t))\} \quad (2)$$

The velocity and position of each particle are updated by (3) and (4):

$$v_{iD}(t+1)=wv_{iD}(t)+c_1\varepsilon[\ p_{iD}(t)-x_{iD}(t)\ ]+c_2\eta[\ p_{gD}(t)-x_{iD}(t)\ ] \quad (3)$$

$$x_{iD}(t+1)=x_{iD}(t)+v_{iD}(t+1) \quad (4)$$

Fig. 1 lists the pseudo-code for the basic PSO algorithm. The two *if* statements are the equivalent of applying (3) and (4) respectively.

B. The parameters of PSO

As shown in (3), the algorithm makes use of two

Create and initialize a D-dimensional PSO : S
Repeat:
For each particle $i \in [1..d]$:
If $f(S.x_i) < f(S.p_i)$
Then $S.p_i = S.x_i$
If $f(S.p_i) < f(S.p_g)$
Then $S.p_g = S.p_i$
End for
Perform PSO updates on S using Equations (3,4)
Until stopping condition is true.

Fig.1. Pseudo-code for the original PSO algorithm

independent random sequences, $\varepsilon \sim U(0,1)$ and $\eta \sim U(0,1)$. These sequences used here have effects on the stochastic nature of the algorithm. The value of v_i is clamped to the range [-v_{max}, v_{max}] to reduce the likelihood that the particle might leave the research space. If the search space is defined by the bounds [-v_{max}, v_{max}], then the value of v_{max} is typically set so that $v_{max}=k^* x_{max}$, where $0.1 \le k \le 1.0$ [3]. The role of the Inertia weight w in (3) is considered critical for the PSO's convergence behavior. The simulation results on the benchmark problems illustrate that an inertia weight w starting with a value close to 1 and decreasing to 0.4 through the course of the run will give the PSO the better performance compared with fixed or linearly decreasing w settings.

The parameters c_1 and c_2 are called acceleration coefficients which influence the maximum size of the step that a particle can take in a single iteration. The velocity update step is specified by the D dimension of the velocity vector associated with the ith particle. From (3), it is clear that c_1 regulates the step size in the direction of the personal best position of that particle, and c_2 regulates the maximum step size in the direction of the global best particle.

C. Differences between PSO and evolutionary algorithm(EA) techniques

A better way to model the v_i term is to think of each iteration not as a process of replacing the previous population with a new one, but rather as a process of adapt. This way the x_i values are not replaced, but rather adapted using the velocity vectors v_i. This makes the difference between other EAs and PSO more clearly: the PSO maintains information regarding position and velocity (changes in position); in contrast, traditional EAs only keep track of positions.

In many of Evolutionary Computation (EC) based techniques, the three main operators, *recombination*, *mutation* and *selection*, are involved. Although having no direct recombination operator, the stochastic acceleration of a PSO towards its previous best position, as well as towards the best particle of the swarm, resembles the recombination procedure in EC.

The inertia weight w governs how much of the previous velocity should be retained from the previous time step [4]. The w can be similar to the temperature parameter encountered in Simulated Annealing (SA)[5]. In SA algorithm, the higher the temperature, the greater the probability the algorithm will explore a region outside of the current local minimum. Here, according to some empirical studies, the linearly decreasing w allows the PSO to explore a large area at the start of the simulation run, and to refine the search later by using a smaller inertia weight [6].

The stopping criterion mentioned in Fig.1 depends on the type of problem being solved. Usually the algorithm is run for a fixed number of function evaluations or until a specified error bound is reached [7].

III. Discussion of PSO Convergence

Without a formal model of why the algorithm works, it was impossible to determine what the behavior of the algorithm would be in the general case. Owing to the stochastic nature of PSO, it is not always possible to observe the characteristics predicted by the theoretical model directly, i.e. a stochastic global optimization algorithm may require an infinite number of iterations to guarantee that it will find the global minimum. So the probability of observing this algorithm locate a global minimum in a finite number of iterations is very small. Despite this problem, it is still possible to see whether the algorithm is really making progress toward its goal, or whether it has become trapped in a local minimum. So, the focus of the work here is the convergence behavior of PSO.

The movement of the particles is to keep track of its coordinates in the problem space. At the end of a training iteration, PSO changes the position of each particle toward its p_i and p_g. The individual velocity and position are updated by the following equation:

$$v(k+1)=wv(k)+c_1(p_i-x(k))+c_2(p_g-x(k)) \tag{5}$$

$$x(k+1)=x(k)+v(k+1) \tag{6}$$

From the basic iterative representation above, the following is derived:

$$v(k+2)=wv(k+1)+c_1(p_i-x(k+1))+c_2(p_g-x(k+1)) \tag{7}$$

$$x(k+2)=x(k+1)+v(k+2) \tag{8}$$

Based on (6) and (7), (8) can be written:

$$\begin{aligned} &x(k+2)+(-w+c_1+c_2-1)x(k+1)+wx(k) \\ &=c_1 p_i+c_2 p_g \end{aligned} \tag{9}$$

Assuming a continuous process, (9) becomes a classical inhomogeneous linear ordinary differential equation. The secular equation of (9) is:

$$\lambda^2+(-w+c_1+c_2-1)\lambda+w=0 \tag{10}$$

Define $\Delta=(-w+c_1+c_2-1)^2-4w$ and set $c=c1+c2$. The Eigen value discussion about the sign of Δ can be made as follows:

1) $\Delta=0$

$$\lambda=\lambda_1=\lambda_2=\frac{-(-w+c_1+c_2-1)}{2}$$

$$x(k)=(A_0+A_1k)\lambda^k$$

$$A_0 = x(0)$$

$$A_1 = \frac{(1-c)x(0) + wv(0) + c_1 p_i + c_2 p_g}{\lambda} - x(0)$$

2) $\Delta > 0$

$$\lambda_{1,2} = \frac{-(-w+c_1+c_2-1)\pm\sqrt{\Delta}}{2}$$

$$x(k) = A_0 + A_1\lambda_1^k + A_2\lambda_2^k$$

$$A_0 = \frac{c_1 p_i + c_2 p_g}{c}$$

$$A_1 = \frac{\lambda_2(x(0)-A_0)-(1-c)x(0)-wv(0)-c_1p_i-c_2p_g+A_0}{(1-c)x(0)+wv(0)+c_1p_i+c_2p_g-x(0)}$$

$$A_2 = \frac{(1-c)x(0)+wv(0)+c_1p_i+c_2p_g-A_0-\lambda_1(x(0)-A_0)}{(1-c)x(0)+wv(0)+c_1p_i+c_2p_g-x(0)}$$

3) $\Delta < 0$

$$\lambda_{1,2} = \frac{w-c_1-c_2+1\pm i\sqrt{-\Delta}}{2}$$

$$x(k) = A_0 + A_1\lambda_1^k + A_2\lambda_2^k$$

$$A_0 = \frac{c_1 p_i + c_2 p_g}{c}$$

$$A_1 = \frac{\lambda_2(x(0)-A_0)-(1-c)x(0)-wv(0)-c_1p_i-c_2p_g+A_0}{(1-c)x(0)+wv(0)+c_1p_i+c_2p_g-x(0)}$$

$$A_2 = \frac{(1-c)x(0)+wv(0)+c_1p_i+c_2p_g-A_0-\lambda_1(x(0)-A_0)}{(1-c)x(0)+wv(0)+c_1p_i+c_2p_g-x(0)}$$

Basically the term convergence means the magnitude of the changes in the positions of the particles in the swarm diminishes over time. A sequence $\{x_i(k)\}$ is said to converge into a minimum x^* if $\lim_{k\to+\infty} x_i(k) = x^*$. Below, $\|\ \|$ is the 2-norm, the conditions guarantee the PSO to coincide with the minimum point is: $\|\lambda_1\| < 1$ and $\|\lambda_2\| < 1$. So,

1) If $\Delta = 0$

$w^2 + c^2 - 2wc - 2w - 2c + 1 = 0$, $0 \le w < 1$.

2) If $\Delta > 0$

$w^2 + c^2 - 2wc - 2w - 2c + 1 > 0, c > 0$, $2w - c + 2 > 0$.

3) If $\Delta < 0$

$w^2 + c^2 - 2wc - 2w - 2c + 1 < 0$, $w < 1$.

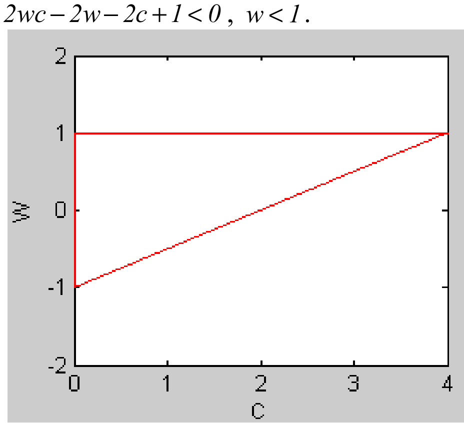

Fig.2. Convergence Area of PSO

To summary what discussed above, the feasible ranges of coefficients to ensure PSO convergence is shown in Fig.2.

IV. Analysis of a Modified PSO (MPSO)

A. *Motivation of MPSO*

Locating global minimum is a very challenging task for all minimization methods [8]. An extended study of the acceleration parameter in the first version of PSO, is given in Kennedy, 1998.Generally, in population-based optimization methods, it is desirable to encourage the individuals to wander through the entire search space, without clustering around local optima, during the early stages of the optimization. On the other hand, during the latter stages, it is very important to enhance convergence toward the global optima, to find the optimum solution efficiently. Properly fine-tuning of the parameters c_1 and c_2 may result in faster convergence and alleviation of local minima.

Considering those concerns, we propose an asymmetrical time varying acceleration coefficients as a new parameter automation strategy for the modified PSO (MPSO), different from the symmetrical time varying acceleration coefficients studied in many literatures. The objective of this development is to enhance the global search in the early part of the optimization and to encourage the particles to converge toward the global optima at the end of the search.

Under this new development, we reduce the cognitive component and increase the social component, by changing the acceleration coefficients c_1 and c_2 with iteration procedure. Additionally, the acceleration coefficients c_1 and c_2 have different rate of slop in order to realize the fine-control of the particles movement. With a large cognitive component and small social component at the beginning, particles are allowed to move around the search space, instead of moving toward the population best. At the same time, a small cognitive component and a large social component allow the particles to converge to the global optima in the latter part of the optimization.

In this MPSO, we suggest this method should be run with a time varying inertia weight factor starting from 1 and gradually decreasing towards 0.4. The modification of c_1 and c_2 can be mathematically represented as follows:

$$c_1 = c_{1\max} - k \times (c_{1\max} - c_{1\min}) / k_{\max} \qquad (11)$$

$$c_2 = c_{2\min} + k \times (c_{2\max} - c_{2\min}) / k_{\max} \qquad (12)$$

where k is the present iteration numbers, k_{max} is the maximum iteration numbers. Fig.3 is the varying curve of acceleration coefficients c_1 and c_2 (Maximum 200 iterations).

B. *Experimental Setting*

To validate the MPSO, it is firstly used to solve the deliberately designed mathematical functions. The two well-known non-linear functions taken from [9] are tested using different selection of c_1 and c_2 for comparison purpose. The first function is Sphere function described by (13) and the second one is the generalized Rastrigrin function described by (14), where $x = [x_1, x_2, ..., x_n]$ is the n-dimensional real-valued vector. The plots of 3-dimensional function $f_1(x)$ and $f_2(x)$ are shown in Fig.4and Fig.5, where the *X*-axis and

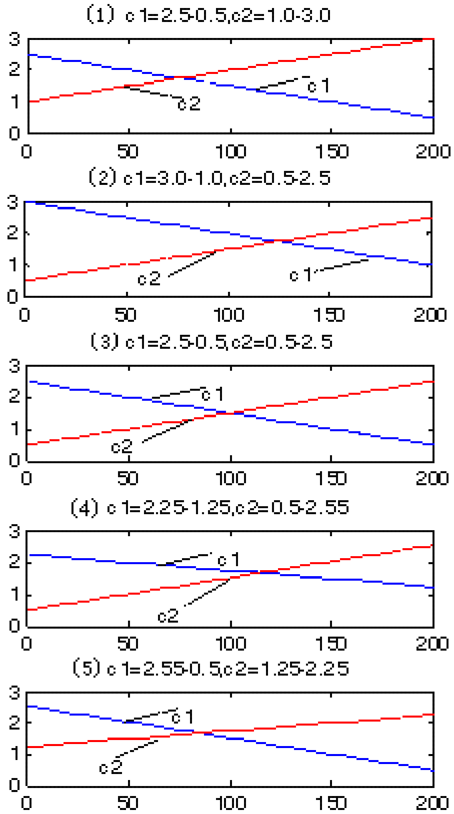

Fig.3. Time-varying acceleration coefficients c_1 and c_2

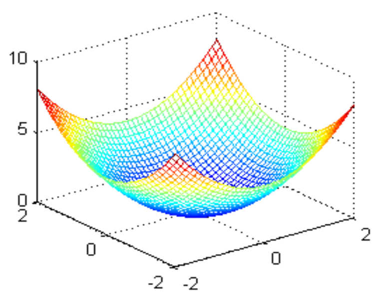

Fig.4. Plot of Sphere function (n=3)

the *Y*-axis stand for the number of iterations and fitness values, respectively. The fitness function is define by $Fitness=lg(f_i(x)), (i=1,2)$.

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{13}$$

$$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{14}$$

C. Simulation Study

The parameter details of c_1 and c_2 in Fig.5 and Fig.6 are as follows:

For curve 1, c_1 and c_2 are constant with $c_1 = c_2 = 2$;

From curve 2 to curve 6, the corresponding time-varying acceleration coefficients c_1 and c_2 are given from Fig.3 (1) to Fig.3 (5). Among them, Fig.3 (3) is the symmetrical time varying acceleration coefficients while the others are asymmetrical time varying acceleration coefficients. Namely, in MPSO the falling rate of c_1 and the rising rate of c_2 are different.

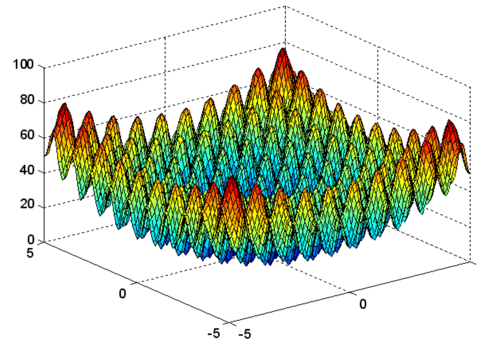

Fig.5. Plot of Rastrigrin function (n=3)

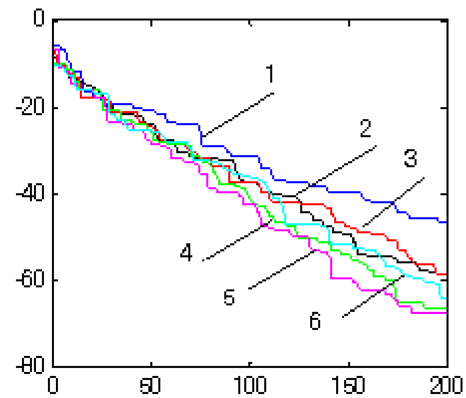

Fig.6. The convergence characteristic for mean fitness value of $f_1(x)$.

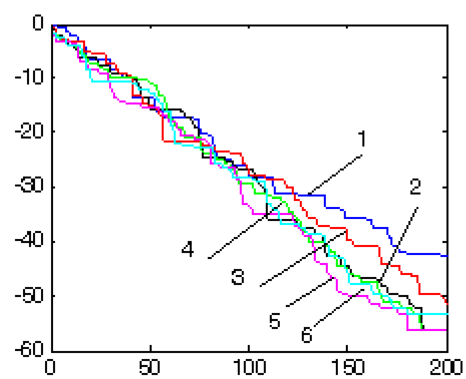

Fig.7. The convergence characteristic for mean fitness value of $f_2(x)$.

From the simulations results, it has been observed that MPSO is capable of converging to a high quality solution at the early iterations (about 150 iterations). Compared curve1 with other curves in Fig.6 and Fig.7, it is obvious the convergence rate of MPSO is faster than that of the basic PSO, which means the MPSO is capable of escaping from the areas of local minima incorporating the global minimum effectively. The MPSO with Fig.3 (4) acceleration coefficients provides stable convergence and thus the best probability of success for the optimization.

V. APPLICATION TO GEAR TRAIN DESIGN

A. Overview of particle swarm optimization

Multi-objective Optimization (MO) problems are very common, especially in engineering applications, due to the multi-criteria nature of most real-world problems. Design of potential function (Skinner and Broughton, 1995), *X*-ray diffraction pattern recognition (Paszkowicz, 1996), curve fitting (Ahonen et al., 1997) are such applications.

In contrast to the single objective optimization case where the optimal solution is clearly defined, in MO there is a whole set of trade-offs. These points are optimal solutions for the MO problem when all objectives are considered simultaneously. The objective functions f may be conflicting with each other, so most of the time it is impossible to obtain for all objectives the global minimum at the same point [10].

B. Problem formulation

Empirical results of MPSO are obtained using various synthetic benchmark functions with well-know characteristics. The second experiment is performed on a real-world problem to act as a gear optimization. The gear ratio for the compound gear train arrangement is shown in Fig.8.

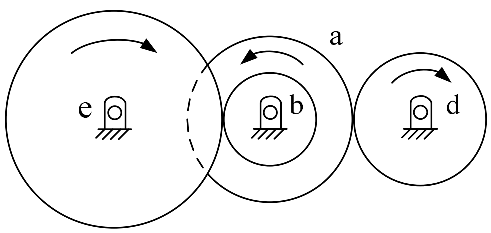

Fig.8. Gear train design

The gear ratio for a reduction gear train is defined as the ratio of the angular velocity of the output shaft to that of the input shaft. In order to produce the desired overall gear ratio, the compound gear train is constructed out of two pairs of the gearwheels, d-a and b-e. The overall gear ratio *n* between the input and output shafts can be expressed as:

$$n = \frac{z_d z_b}{z_a z_e} \tag{15}$$

Here *z* denotes the number of teeth on each gearwheel. According to the experience, it is desirable to produce a gear ratio to 1/6.931 as close as possible. For each gear, the number of teeth must be between 12 and 60. The design variables are denoted by a vector $Z=[z_d, z_b, z_a, z_e]^T$, and hence $[z_d$, z_b, z_a and z_e are the numbers of teeth of gears d,b, a and e respectively, which must be integers

The problem is formulated as:

$$\min f(z)=(\frac{1}{6.931}-\frac{z_d z_b}{z_a z_e})^2, \tag{16}$$

Subject to $12 \le z_i \le 60 \quad (i=d,b,a,e)$

C. Results and Discussions

In the gear train design procedure, particles should be highly confined to the defined search space as input parameters outside the search space are not practically viable. Therefore, whenever a dimension of a particle moves away from the predefined search space it is replaced by corresponding random value inside the search space.

Table 1 list the various gear train solutions and compares the results with different acceleration coefficients as shown in Fig.3. It can be observed from Table 1 the solution with bold writings is the best solution. It is obvious that there are four global optima by inspecting (16). As PSO works with a population of possible solutions rather than an only solution, it is easy to find multiple global optima based on one solution and the symmetry of (16) for this problem:

$$Z_1 = [z_d, z_b, z_a, z_e]^T = [16,19,43,49]^T$$
$$Z_2 = [z_d, z_b, z_a, z_e]^T = [19,16,43,49]^T$$
$$Z_3 = [z_d, z_b, z_a, z_e]^T = [16,19,49,43]^T$$
$$Z_4 = [z_d, z_b, z_a, z_e]^T = [16,19,49,43]^T$$

Actually it is almost impossible to have optimization tasks with multiple global optima which can be detected so simply. However, by using a sufficiently large population, it is capable of obtaining all four alternative solutions within a single trial.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, the MPSO an effective particle swarm optimization algorithm with asymmetrical time-varying acceleration coefficients is applied for the alleviation of the local minima problem for the PSO. Compared with symmetrical time-varying acceleration coefficients studied in many literatures, this technique eliminates undesirable local minima but preserves the global ones.

This MPSO is also applied to the gear train design procedure to find the right number of teeth on each gearwheel for optimal gear ratio. Competitive results observed validate that the MPSO is superior to the existing methods of this kind for finding the best solution, both in the solution quality and algorithm robustness. Future research should concentrate mainly on developing self-adaptation techniques for the coefficients modification and penalty functions to reduce the number of unusable modifications and further application area to scientific, engineering and economic optimization problems.

REFERENCES

[1] J.Kennedy and R.C. Eberhart. "Particle swarm optimization," In Proceedings of IEEE International Conference on Neural Networks, volume IV, pp.1942-1948.

[2] D.H.Wolpert and W.G.Macready. "No free lunch theorems for optimization," IEEE Transactions on Evolutionary Computation,(4) 1997, pp.67-82.

[3] D.Corne, M.Dorigo, and F.Glover. New Ideas in Optimization, McGraw Hill, 1999, pp.379-387.

[4] Y.Shi and R.C.Eberhart. "A modified particle swarm optimizer," In IEEE International Conference of Evolutionary Computation, Anchorage, Alaska, May 1998.

[5] R.H.J.M.Otten and L.P.P.P.Van Ginneken. The Annealing Algorithm. Kluwer, Boston, USA, 1989.

[6] Y.Shi and R.C.Eberhart. "Empirical study of particle swarm optimization," In IEEE International Conference of Evolutionary Computation, Indianapolis, USA, July 1999, vol.3, pp.1945-1950.

[7] J.Salerno. "Using the particle swarm optimization technique to train a recurrent neural model," In Proceedings of the Ninth IEEE International Conference on Tolls with Artificial Intelligence, 1997, pp.87-93.

[8] Y.Fukuyama and H.Yoshida. "A particle swarm optimization for reactive power and voltage control in electric power system," In Proceedings of IEEE Congress on Evolutionary Computation, Seoul, Korea,2001, pp.87-93.

[9] Angeline,P.J.(1998). "Evolutionary optimization versus particle swarm optimization: philosophy and performance difference," 1998 Annual Conference on Evolutionary Programming, San Diego.

[10] G.Q.Bao and J.Z.Jiang. "Optimal design for cogging torque reduction of transverse flux permanent motor using PSO algorithm," Power Electronics and Motion Control Conference, IPEMC 2004, ISBN: 7-5605-1869-9, vol.1, Xi'an, China, pp.260-263.

Table 1
Summary of the optimal solutions for the gear train design

c_1,c_2 / *Results*	$c_1=c_2=2$	c_1=2.5-0.5 c_2=1.0-3.0	c_1=3.0-1.0 c_2=0.5-2.5	c_1=2.5-0.5 c_2=0.5-2.5	c_1=2.55-0.5 c_2=1.25-2.25	***c_1=2.25-1.25*** ***c_2=0.5-2.55***
z_d	27	19	19	18	16	19
z_b	16	17	17	19	19	16
z_a	45	42	45	42	43	49
z_e	60	49	52	50	50	43
$f(z)$	$2.5*10^{-4}$	$1.6*10^{-4}$	$1.3*10^{-5}$	$9.7*10^{-6}$	$8.3*10^{-6}$	$2.6*10^{-12}$
Gear Ratio	0.160000	0.156948	0.140741	0.141171	0.141395	0.144283
Error(%)	10.9	8.78	2.45	2.15	1.999	***0.0013***