Journal of
Marine Science and Technology

Volume 24 | Issue 4 Article 17

STUDY ON THE NEW VARIANT OF PARTICLE SWARM METHOD FOR OPTIMIZATION DESIGN

Jinn-Tong Chiu
Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C., cjt7725@gmail.com

Chih-Chung Fang
Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

Recommended Citation
Chiu, Jinn-Tong and Fang, Chih-Chung (2016) "STUDY ON THE NEW VARIANT OF PARTICLE SWARM METHOD FOR OPTIMIZATION DESIGN," *Journal of Marine Science and Technology*: Vol. 24: Iss. 4, Article 17.
DOI: 10.6119/JMST-016-0325-2
Available at: https://jmstt.ntou.edu.tw/journal/vol24/iss4/17

Journal of Marine Science and Technology, Vol. 24, No. 4, pp. 832-841 (2016)
DOI: 10.6119/JMST-016-0325-2

STUDY ON THE NEW VARIANT OF PARTICLE SWARM METHOD FOR OPTIMIZATION DESIGN

Jinn-Tong Chiu and Chih-Chung Fang

Key words: particle swarm optimization, spring, optimization.

ABSTRACT

A new variant of Particle Swarm Optimization (PSO) is developed to improve the performance of PSO, which has been widely used in various fields for optimization. The proposed PSO incorporates a space partitioning technique in grid method with PSO. In the searching process of the new algorithm, three position vectors are introduced to enhance the exploration of the particles in the population of PSO and hence helpful for a global optimization problem of interest.

First, the proposed variant of PSO is verified by applying it to the seven benchmark functions and thereafter proved from the results as a robust one. Next, we applied the algorithm to the optimization design of the M-type spring used in the 3C equipments. With the maximum stress as the objective function of the designing product and the thrust as the constraint, we obtain from the computation the designing parameter set of the spring, which gives the designing spring a more uniform stress distribution and a reduction of the maximum stress by around 9.2% helpful to increase the lifetime of the initiated product.

I. INTRODUCTION

Particle Swarm Optimization algorithm (PSO) is inspired by the ecological behavior of birds. It is argued by Acan and Gunay (2005) that PSO can find the global optimum in most cases for low-dimensional problems and for uni-modal problems. However, for high-dimensional problems or problems with multiple extreme points, it usually fails to find the global optimum. Hence, improvement on PSO's performance has been proposed since it was developed.

Suganthan (1999) proposed dynamically increasing neighboring particles for linearly decreasing inertia weights (L.PSO). This applies the idea of a linearly decreasing inertia weight function to recognition of acceleration constant C_1 and social acceleration constant C_2. It improves the capability of global searching and search accuracy. Starting from basic mathematical and analytical ideas, Kennedy and Clerc (2000) proposed particles swarm optimization with constriction factor (PSO-CF), which effectively confines the search trajectory of each particle and does not limit their maximal speed $v_{\max}$. Shi and Eberhart (2001) used PSO-CF parameters and their characteristics to adjust the inertia weight function in PSO-IW and applied it to optimizations of dynamic systems. Chatterjee and Siarry (2004) proposed non-linear inertia weight, which selects inertia weights according to the relationship between the inertia weights of the previous generation and next generation, thereby increasing the activity (diversity) of the particle swarm. Liu et al. (2005) combined PSO with a chaotic system. First, they improved the inertia weight according to the relationship between each particle and the population average. They then added a chaotic local search (CLS) to increase the activity and accuracy of PSO. S. Fan and Erwie (2006) proposed combining PSO with the simplex algorithm. It replaced the original particle movement pattern in PSO with the movement pattern of the simplex algorithm, using the particle diffusivity in the simplex algorithm to avoid PSO results falling in local optima. Yin et al. (2010) proposed a few Cyber Swarm algorithms. They adopted the adaptive learning-and-memorizing strategy to increase the swarm activity and the search accuracy. Mohammed El-Abd (2011) combined two swarm intelligence algorithms and proposed the Artificial Bee Colony PSO, which utilizes the excellent performance of the Artificial Bee Colony on multimodal functions to address the drawbacks of PSO on multimodal problems. Wang et al. (2013) paper proposed a hybrid PSO algorithm, called DNSPSO, which employs a diversity enhancing mechanism and neighborhood search strategies to achieve a trade-off between exploration and exploitation abilities. Zhao et al. (2015) proposed a compact PSO, which has excellent performance with less hardware requirement and plays an essential role to obtain the optimal scaling factors.

From the above studies, we can see that there are three main directions to improve the PSO algorithm: (1) change the mathematical model and calculation of PSO; (2) improve the model's parameters, such as inertia weight ω and acceleration constants C_1 and C_2; (3) incorporate a strong local search mechanism during PSO search. The first two strategies both enhance the swarm activity of PSO and increase the population diversity by different methods, in order to prevent PSO from premature

Paper submitted 05/11/15; revised 11/02/15; accepted 03/25/16. Author for correspondence: Jinn-Tong Chiu (e-mail: cjt7725@gmail.com).

Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

convergence. The third strategy is to further search the good solutions to prevent potentially best particles from being absorbed by others, which will cause search failure.

Different from the above three strategies improving the PSO, we in the study attempt to propose a variant of PSO by adopting the Grid method (Jasbir, 1989), which shrinks a given original search zone gradually into a sufficiently small region where the extreme value of a problem lies inside. Besides, we introduce three different values out of the results by the Grid method, and those values would replace the worst three particles in performance among the population in PSO. The replacement is found helpful for strengthening the particle's exploration in the ensuing evolution of PSO and hence heightens the likelihood of obtaining the global optimal of the problem of interest.

The verification of the proposed modified PSO will be processed by applying it to seven benchmark functions, uni-modal and multi-modal. Further, we will practice the modified PSO in the design of an M type spring to find the best one among all the sets of the relevant designing variables of the spring capable of enduring the possible longest lifetime. The M type spring is used in a 3C equipment such as usb flash drive, slider phone or remote controller.

II. MODIFIED PARTICLE SWARM OPTIMIZATION

1. Iterative Grid Method

Grid Method is also known as one-dimensional search. In a fixed search range $[a,b]$, we select a few equally-spaced points, $\alpha_1, \alpha_2, \ldots, \alpha_{nd}$ such that the search range is divided into $(nd+1)$ subintervals. Next, the function $f(x)$ is evaluated at these points, and the point with the minimal value is denoted by f_m, i.e., $f_m = min\{f(a), f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_{\text{nd}}), f(b)\}$. Because the point α_{m} corresponding to f_m must lie between the left and right neighbors $\alpha_{\text{m-1}}$ and $\alpha_{\text{m+1}}$ of the point, we then take [α_{m-1} , α_{m+1}] as the new search range and repeat the above process to shrink the search range until a predefined accuracy is reached.

We generalize the Grid Method to handle n-dimensional problems with several steps. First we randomly pick a vector $\vec{X}_{ref} = \{x_1, x_2, \ldots, x_n\}$ in the search range and apply the original Grid Method to each dimension, each time fixing the values in the other n-1 dimensions. The obtained results are then substituted for the corresponding dimension in $\vec{X}_{ref}$ and we then have the optimal solution for the n-dimensional problem.

The above method, which successively grid searches for each variable, is also called Iterative Grid Method and is computationally simple. For an optimization problem in which variables are not correlated, it can easily find solutions close to the global optimum.

2. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a stochastic optimization technique that is motivated by the behavior of a flock of birds. A PSO population is called a swarm, which is imagined to be flying in the n-dimensional search space defined by a given optimization problem. Each individual in the swarm is known as a particle, and each particle represents a solution to the problem. The problem's objective (or fitness) function is expressed in terms of these particles to measure how close the computed solution comes to the goal.

Initially, the positions and the velocities of all of the particles are randomly assigned. Each particle keeps tracks of its coordinates in a defined search space generation by generation. We define $S_i^k = (s_{i1}, s_{i2}, \ldots, s_{in})^k$ to denote the position of a particle i at the k-th generation, with a corresponding velocity denoted by $v_i^k = (v_{i1}, v_{i2}, \ldots, v_{in})^k$. Each generation generates two different types of solutions: the particle best and the globally best solutions. The particle's best solution is the location that a particle so far achieved is closest to the exact solution, and usually denoted as the *pbest*. The globally best solution is the best among all particles' *pbests* and termed as the *gbest*.

Given the positions and velocities of all of the particles, the velocity of particle i at the $(k + 1)$th generation is determined by three vectors and is expressed mathematically in Eq. (1). Its position is updated by Eq. (2).

$$v_{ij}^{k+1} = wv_{ij}^k + c_1 \times rand() \times (s_{ij}^{pbest} - s_{ij}^k) + c_2 \times rand() \times (s^{gbest} - s_{ij}^k) \quad (1)$$

$$s_{ij}^{k+1} = s_{ij}^k + v_{ij}^{k+1} t;\ j = 1, \ldots, n\ ; \quad (2)$$

The first term on the right-hand side of Eq. (1) describes the particle's momentum with an inertial parameter w balancing the global and local searches. The second term, which is the difference between the S_i^{pbest} (also known as *pbest*) and the particle's current position, is multiplied by a learning factor, C_1, and a random number rand() that represents the particle's self-cognition ability. Similarly, the third term, which is the difference between the S^{gbest} (also known as *gbest*) and the particle's current position, is multiplied by another learning factor, C_2, and a random number, rand(), indicating the particle's sociability. The two different factors, C_1 and C_2 boost the particle's self-cognition ability and sociability. The random number rand() is chosen from a uniform distribution within the range (0, 1). The t is search time step and be assigned to 1. The search process is illustrated in Fig. 1.

A single PSO run proceeds as follows:

(1) Create a swarm of N_p particles that are distributed within an n-dimensional search space. Randomly initialize each particle's position and velocity, with both expressed as n-vectors. Specify the values of the parameters ω, C_1, C_2, and the maximum generation number required by the PSO.

(2) Define the problem's objective (fitness) function. Evaluate

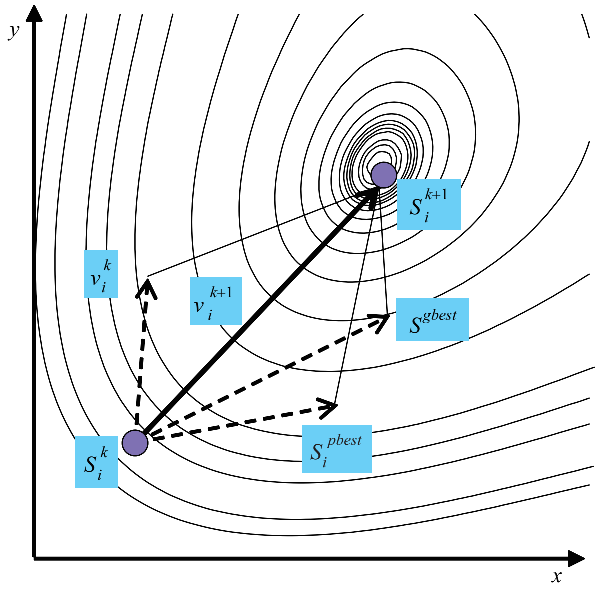

Fig. 1. Particle speed and location update.

the objective function value for each particle and for each generation.

(3) Compute each particle's velocity and position using Eqs. (1) and (2). Set the two different best positions of each particle, *pbest* and *gbest,* to $\min_{i=1,N_p}\{(f_i(pbest), f(gbest)\}$.

(4) Check whether the objective function value at the current generation meets the criterion of whether the system has reached the maximum generation number. If yes, then go to Step 5; otherwise, go to Step 2.

(5) Halt.

3. The Modified Particle Swarm Optimization, MPSO

Among all of the types of well-known heuristic computation algorithms, the PSO is a simple, fast and easily implemented evolution algorithm; however, its performance usually provides an unreliable result when a complicated problem is solved, such as a multi-modal problem or a problem with a large number of variables. Therefore, many variants of PSOs have been developed to improve its performance on complicated problems by modifying the parameters defined in the PSO or by introducing local search methods or cooperating with other computational algorithms.

As described previously the Iterative Grid Method is well suited for separable problems, the results by the method possibly give information on the location of the problem's solution lies. Hence we in the study attempt to incorporate the Iterative Grid Method with the PSO to improve the PSO's performance for non-separable or multi-modal problems with a large number of variables. The developed algorithm is thus called the modified Particle Optimization (MPSO).

In the development of the proposed MPSO algorithm, we introduce three creative positions, which are denoted by $\vec{X}_{avg}^{pbest}$, $\vec{X}_{igm}$, and $\vec{X}_{Disturb}$, to replace the three poorest fitness particles among the population of particles in the PSO. This replacement aims to improve both the exploitation and the exploration of the particles when particles in the evolution stay in a stagnant situation.

The best average position $\vec{X}_{avg}^{pbest}$ represents the position of the center point of the population and is determined by averaging all of the particle's current positions, expressed as follows:

$$\vec{X}_{avg}^{pbest} = \frac{1}{N_p}\sum_{t=1}^{p} S_i^{pbest} \tag{3}$$

The position $\vec{X}_{igm}$ is determined only from applying the iterative grid method. Technically, the converged $\vec{X}_{igm}$ is close to the exact solution for a separable problem but hardly for a non-separable one. In spite of the position obtained in violation of the problem's non-separable characteristics, it possibly provides a tendency of variation for individual components among the population in the evolution process. Next, to recover the ignorance of the interdependence among the variables while obtaining the $\vec{X}_{igm}$, we create another disturbance position $\vec{X}_{Disturb}$, which is determined by the positions $\vec{X}_{igm}$, $\vec{X}_{avg}^{pbest}$ and $\vec{S}^{gbest}$. Note that the last two positions are obtained with the consideration of mutual interaction among the variables.

Thus, we take the center position $\vec{X}_{avg}^{pbest}$ as a reference point in the determination of the position $\vec{X}_{Disturb}$. Then, each component of $\vec{X}_{Disturb}$ is determined through several procedures that are described below.

Starting with the first component of $\vec{X}_{avg}^{pbest}$, we create two virtual vectors of iterative grid position $\vec{T}_{igm}$ and best position $\vec{T}_{gbest}$. The first component of $\vec{T}_{igm}$ is taken from the first component of $\vec{X}_{igm}$, and the remaining components of the vector $\vec{T}_{igm}$ are all the same as those of vector $\vec{X}_{avg}^{pbest}$. Similarly, the first component of the vector $\vec{T}_{gbest}$ is taken from the first component of S^{gbest}, and the remaining components of $\vec{T}_{gbest}$ are all the same as those of the $\vec{X}_{avg}^{pbest}$. Next, we check the function values of the three positions $\vec{X}_{avg}^{pbest}$, $\vec{T}_{igm}$ and $\vec{T}_{gbest}$; the first component of the position with the minimum function value is selected to be the first component of $\vec{X}_{Disturb}$. Following the similar procedure, all of the remaining components of $\vec{X}_{Disturb}$ are produced.

Note that the replacement is launched once for each number of evolution generations, denoted by N_{rplc}, reaches the value set in experimenting PSO. The flowchart of the MPSO is shown in Fig. 2.

Table 1. Seven benchmark functions.

	Function	Math. Expression	Domains of variables	Exact Solution
F1	Sphere	$F_1(x)=\sum_{i=1}^{n} x_i^2$	$-100 \le x_i \le 100$ $i = 1, 2, \ldots, n$	$x_i = 0, \quad i = 1, 2, \ldots, n$ $f(\vec{x}) = 0$
F2	Quadric	$F_2(x)=\sum_{i=1}^{n}(\sum_{j=1}^{n} x_j)^2$	$-100 \le x_i \le 100$ $i = 1, 2, \ldots, n$	$x_i = 0, \quad i = 1, 2, \ldots, n$ $f(\vec{x}) = 0$
F3	Rosenbrock	$F_3(x)=\sum_{i=1}^{n-1}\left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right]$	$-30 \le x_i \le 30$ $i = 1, 2, \ldots, n$	$x_i = 1, \quad i = 1, 2, \ldots, n$ $f(\vec{x}) = 0$
F4	Ackley	$F_4(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$-32 \le x_i \le 32$ $i = 1, 2, \ldots, n$	$x_i = 0, \quad i = 1, 2, \ldots, n$ $f(\vec{x}) = 0$
F5	Griewank	$F_5(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \le x_i \le 600$ $i = 1, 2, \ldots, n$	$x_i = 0, \quad i = 1, 2, \ldots, n$ $f(\vec{x}) = 0$
F6	Rastrigrin	$F_6(x) = \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	$-5.12 \le x_i \le 5.12$ $i = 1, 2, \ldots, n$	$x_i = 0, \quad i = 1, 2, \ldots, n$ $f(\vec{x}) = 0$
F7	Schwefel	$F_7(x) = \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	$-500 \le x_i \le 500$ $i = 1, 2, \ldots, n$	$x_i = -420.9687, \quad i = 1, 2, \ldots, n$ $f(\vec{x}) = -418.98 \times n$

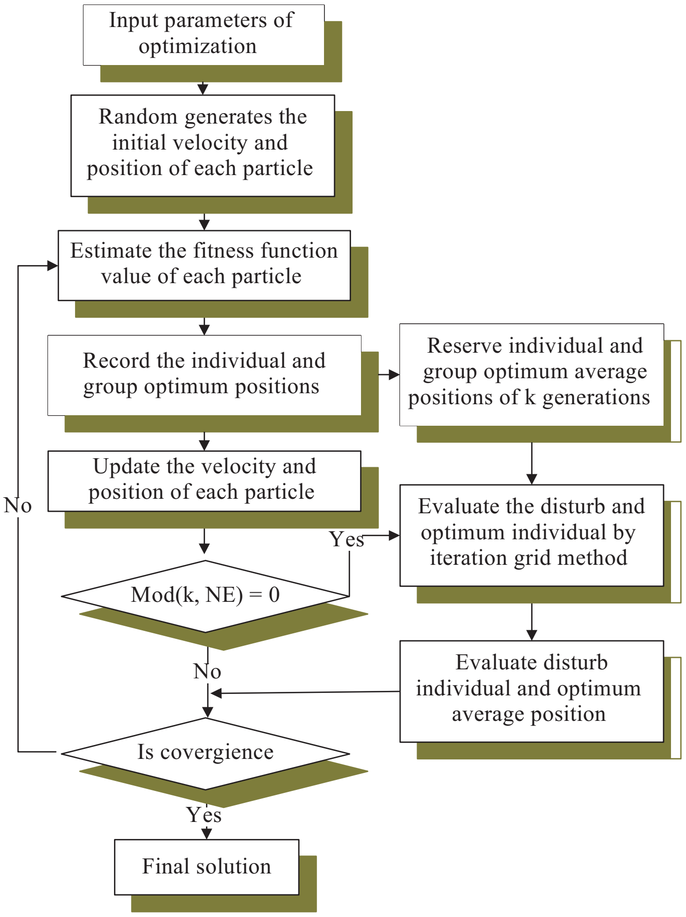
Input parameters of optimization
Random generates the initial velocity and position of each particle
Estimate the fitness function value of each particle
Record the individual and group optimum positions
Reserve individual and group optimum average positions of k generations
Update the velocity and position of each particle
Evaluate the disturb and optimum individual by iteration grid method
Yes
Mod(k, NE) = 0
No
Evaluate disturb individual and optimum average position
No
Is covergience
Yes
Final solution

Fig. 2. Modified PSO algorithm flow chart.

Table 2. The results for the seven functions, each with 30 variables.

	PSO			MPSO		
function	f_{av}	f_b	success rate	f_{av}	f_b	success rate
F1	8.37E-07	5.89E-07	100%	8.76E-07	6.81E-07	100%
F2	3.46E+03	9.26E-07	40%	9.15E-07	0.00E+00	100%
F3	1.82E+04	1.00E-06	3.3%	1.20E-06	9.88E-07	100%
F4	1.30E+01	8.46E-07	23.3%	6.48E-07	8.88E-16	100%
F5	9.03E-03	7.72E-07	43.3%	6.94E-07	0.00E+00	100%
F6	5.55E+00	8.67E-07	83.3%	7.41E-07	0.00E+00	100%
F7	6.71E+03	8.11E+03	0	-1.26E+04	-1.26E+04	100%

Success rate: convergence times of 30 tests.

Table 3. The results for the seven functions, each with 50 variables.

	PSO			MPSO		
function	f_{av}	f_b	success rate	f_{av}	f_b	success rate
F1	9.32E-07	7.46E-07	100%	7.22E-07	0.00E+00	100%
F2	1.91E+04	9.99E-07	3.33%	9.62E-07	0.00E+00	100%
F3	2.18E+04	9.98E-07	10%	9.99E-07	9.95E-07	100%
F4	1.89E+01	9.63E-07	3.33%	8.88E-16	8.88E-16	100%
F5	1.44E-02	8.90E-07	43.33%	8.79E-07	0.00E+00	100%
F6	1.98E+01	9.21E-07	50%	6.44E-07	0.00E+00	100%
F7	-1.06E+04	-1.33E+04	0	-2.09E+04	-2.09E+04	100%

Success rate: convergence times of 30 tests.

Table 4. The results for the seven functions, each with 100 variables.

	PSO			MPSO		
function	f_{av}	f_b	success rate	f_{av}	f_b	success rate
F1	1.71E+04	9.97E-07	16.66%	0.00E+00	0.00E+00	100%
F2	6.70E+04	2.50E+04	0	9.98E-08	0.00E+00	100%
F3	5.15E+07	9.99E-07	6.66%	1.00E-06	9.97E-07	100%
F4	1.99E+01	1.99E+01	0	8.88E-16	8.88E-16	100%
F5	5.82E-02	7.08E-07	40%	5.22E-07	0.00E+00	100%
F6	1.07E+02	2.77+01	0	0.00E+00	0.00E+00	100%
F7	-2.13E+04	-2.48E+04	0	-4.18E+04	-4.18E+04	100%

Success rate: convergence times of 30 tests.

III. CASES AND DISCUSSIONS

1. Comparison with the Original PSO

To demonstrate the effectiveness of the proposed MPSO, we applied it on seven benchmark problems, which are displayed in Table 1. The functions of F2 and F3 are non-separable functions, and the rest are separable. Furthermore, the first three functions F1, F2, F3 are uni-modal and the rest are multi-modal.

In the computation, the relevant parameters were given values as follows:

The number of particles in the population Np = 80; the weight coefficient $\omega = 0.6$; the learning coefficients $C_1 = 1.5$ and $C_2 = 2.0$; the number of subintervals $n_{di} = 9$; the replacement rate $N_{rplc} = 250$; the maximum number of generations $N_{max} = 50{,}000$; and the convergence criteria $\varepsilon = 10^{-6}$. There are three different numbers of variables that were used for the comparison: $n = 30$, $n = 50$, and $n = 100$.

Because the proposed algorithm MPSO is developed for improving the PSO algorithm, the results determined by MPSO are compared with those of the original PSO. To avoid a bias in the results, we performed each function through 30 independent runs. To present the algorithm's performance, we define a parameter f_{avg}, which averages the final solutions of the 30 runs, and a parameter f_b, which represents the best value among the 30 final function values. Obviously, the less difference between f_b and f_{avg}, the more reliable the algorithm is for the function tested. Moreover, we define the success rate as the rate of the number of final solutions that met the convergence criterion ε over the total of 30 runs.

Therefore, the results obtained respectively by both MPSO and PSO for all of the seven functions with 30 variables, 50 variables and 100 variables are presented, respectively, in Tables 2-4. The results shown in the three Tables indicate that the proposed MPSO algorithm work much successfully and is hardly affected by types of functions (separable or non-separable, uni-modal or multi-modal) and the number of the variables. In regard to the original PSO, except the F1, which is a separable and uni-modal function, the algorithm cannot attain a success rate of 100% for all the rest of functions, particularly for the F7, which is a multi-modal function, a success rate of zero for all the three different variables. The results clearly illustrate the modified PSO outperforms the original PSO with a great success.

2. Simulation Analysis of the M Type Spring

According to the statistics (Lin, 2007), electronic products have a life cycle of about 2~5 years. Their springs must withstand 40,000~70,000 slides (about 40~50 times per day). Therefore, the springs' anti-fatigue capability should be at least 70,000 times. When designing the springs in 3C equipment, both different spring strengths and install positions will affect the phone's lifetime.

Yihui Tsai (2008) applied the Taguchi method to analyze the M type springs in 3C equipment for the optimal geometry sizes of the 3C equipment springs to extend their lifetime. Zhang (2009) used Genetic Algorithm to find the optimal install position such that no lateral force is induced when the lid slides.

Therefore, to further demonstrate the effectiveness of the proposed algorithm on practical application, we select a M type spring used in the 3C equipment. In Fig. 3 is shown the

Table 5. Material Properties.

Stainless Steel Alloy SUS304	
Young's Modulus (MPa)	100000
Poisson Ratio	0.29
Yield Stress (MPa)	2200

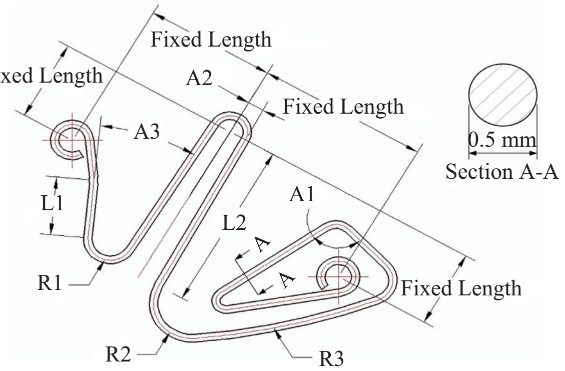

Fig. 3. M type spring geometry.

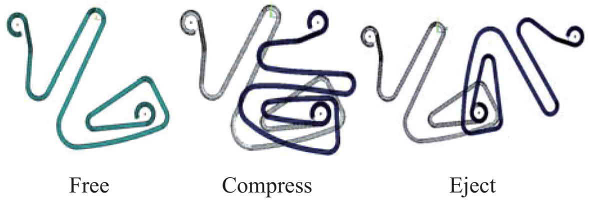

Fig. 4. M type spring movement in a semi-automatic 3C equipment.

geometry of the M type spring. The sectional diameter of the spring is 0.5 mm. There are eight variables to describe the performance of the product. The properties of the material used for the spring are listed in Table 5.

3. Description of the Spring Movement

The spring movement in the 3C equipment has three continuous states: (1) free state, (2) compress state, and (3) eject state, as shown in Fig. 4.

Damages to malleable material are often due to shear stress. Then von Mises criterion considers the shear damage and can be used to determine if the material is in linear elastic region or plastic region. Assuming $\sigma_1, \sigma_2, \sigma_3$ are the principal stresses in 3D space, we can use the following formula to calculate von Mises stress:

$$\sigma_{eqv} = \sqrt{\tfrac{1}{2}[(\sigma_1 - \sigma_2)^2 + (\sigma_1 - \sigma_3)^2 + (\sigma_3 - \sigma_2)^2]} \qquad (4)$$

If the yield strength of the tensile test is S_y, then when $\sigma_{eqv} < S_y$, the material is in the linear elastic region and will not yield, satisfying Hooke's Law.

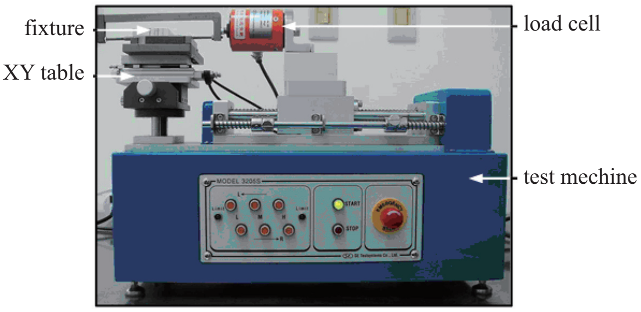

Fig. 5. Horizontal electric tester (Chen, 2007).

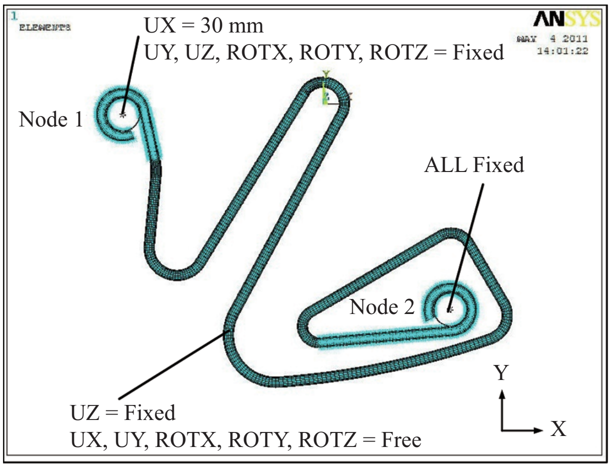
UX = 30 mm
UY, UZ, ROTX, ROTY, ROTZ = Fixed
Node 1
ALL Fixed
Node 2
UZ = Fixed
UX, UY, ROTX, ROTY, ROTZ = Free
Y
X

Fig. 6. Boundary conditions.

When the spring in a 3C equipment undergoes forced displacement, we observe the contact forces of the spring in each direction. As in Chen (2007), the experiment equipment is a horizontal electric test with fixtures, an XY mobile station and a Load Cell, as shown in Fig. 5.

When simulating the M type spring movement, it will have large deformation and contact behavior, which can easily cause divergence and the failure of optimization. Therefore, we use two different analysis elements for simulation: (1) solid element used for analysis and (2) beam element for simplifying the analysis model. We use ANSYS APDL (2012) to create a finite element model for the M type spring and provide element type, element size, and boundary condition for geometry nonlinear analysis.

The analysis model consists of three parts. The M type spring is in the X-Y plane. The lid and base bolt are simulated by two cylinders with rigid surfaces. The contact boundary condition applies between the spring and two bolts. The boundary condition is shown in Fig. 6. We push Node 1 30 mm in X direction while Node 2 is fixed. The direction of spring Z is also fixed. There are 100 steps, each advancing 0.3 mm. We use static mode to simulate the deformation of the M type spring. For solid element analysis, we use the element named Solid 185 8-Node; for contact element, we use Conta175/Target170, element number: 30030, number of nodes: 31934.

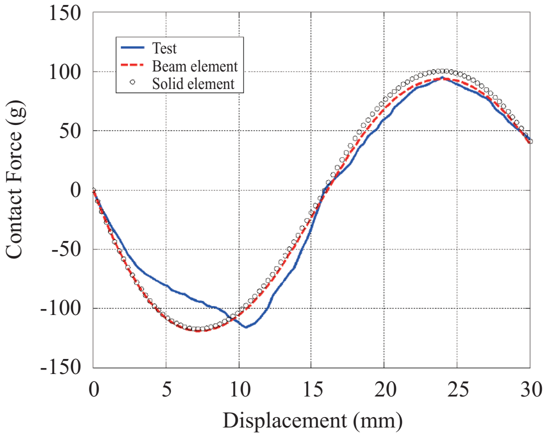

Fig. 7. Displacement-reaction force in X direction.

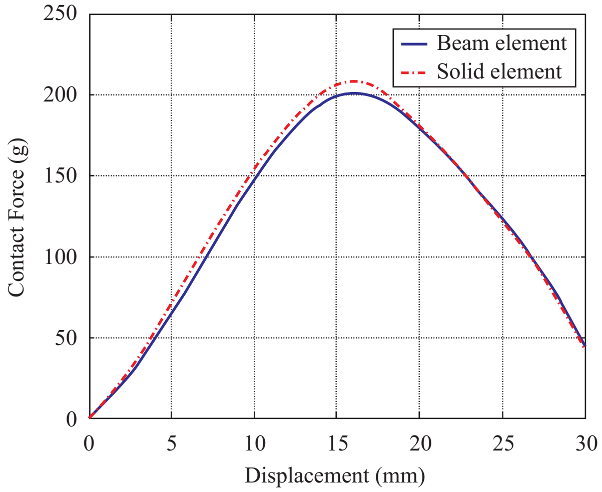

Fig. 8. Displacement-reaction force in Y direction.

Fig. 7 shows the beam element analysis, solid element analysis and physical measurements of X-directional displacement and counter force using ANSYS. The horizontal axis is the displacement of Node1 δ_x from 0 mm to 30 mm. The vertical axis shows the X-directional counter force, measured by grams (g). The solid curve is the physical measurement, the dashed curve is the beam element analysis and the dotted curve is the solid element analysis.

Fig. 8 shows the solid analysis and beam analysis in the Y-directional displacement and counter force. The former has a maximal counter force of 208.3 g on Y direction, while the latter has 201.3 g, with an error of 3.36%. From Figs. 7 and 8, we can see that the simulated values have a similar trend to the experiment values.

Table 6 compares the simulated and experimental results. The counter force, elastic force and positions where force direction changes are obtained from the two element analyses are all within reasonable error ranges, which are: 2.59%, 1.27%, and 1.93% for beam analysis; 1.04%, 5.57%, and 1.93% for solid analysis. We observe stress distribution in four regions of the M type spring, namely region A, B, C and D.

Table 6. Comparison of ANSYS Simulation and physical measurements.

	Counter force	Elastic force	Position where the force direction changes
Experiment Result	116.0 g	95.0 g	16.04 mm
Beam Simulation result	119.01g	93.79g	16.35 mm
Solid simulation result	117.21g	100.30g	16.35 mm
Error			
Beam and experiments	2.59%	1.27%	1.93%
Solid and experiments	1.04%	5.57%	1.93%

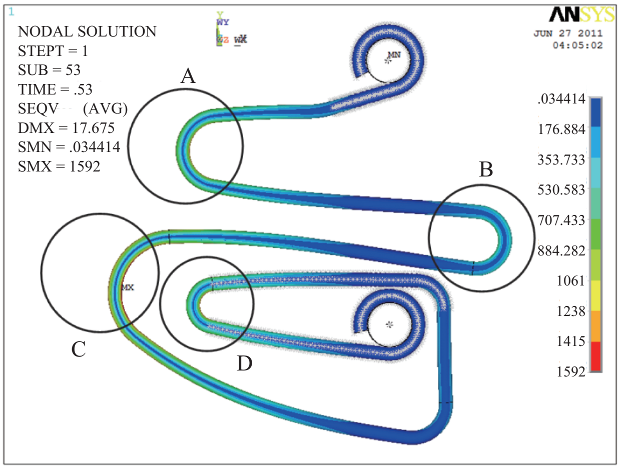

Fig. 9. The stress distribution when the slider goes to 15.9 mm by solid element.

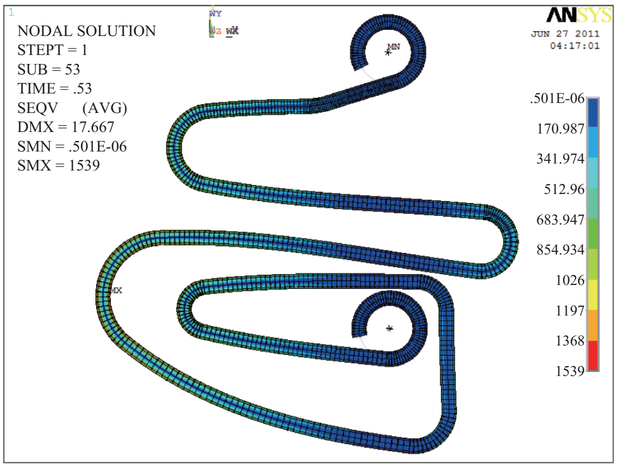

Fig. 10. The stress distribution when the slider goes to 15.9 mm by beam element.

Figs. 9 and 10 show the von Mises stress distribution in

Table 7. Initial values and ranges of the parameters for M type spring.

Design Variables	A1	A2	A3	L1	L2	R1	R2	R3
Initial Value	90	1.5	26	3.8	11	1.35	2.3	40
Lower Bound	88	1	21	2.5	11	1.1	2.1	30
Upper Bound	105	2	27	5.5	13	1.6	2.6	50

A: A1 and A3 Angle (deg), A2 Distance (mm),
R: Radius (mm), L: Length (mm).

solid analysis and beam analysis when the slider moves to 15.9 mm, which is also when the maximal stress occurs. The stress concentrates at the rounded corner in Region C. The former stress is 1592 MPa and the latter is 1539 MPa. Since the stress obtained from beam element calculation is closer to the average, it could be smaller than the stress obtained from solid element calculation. The solid element analysis shows that the stresses in Regions A, B, and D are 1292 MPa, 735 MPa and 1375 MPa, respectively. We can see that the stress on Region B is not high so the stress cannot be evenly distributed between the regions. In the following design optimization for the M type spring, we use beam element analysis during the searching process and then use solid element analysis as the basis for the final solution.

IV. DESIGN OPTIMIZATION FOR THE M TYPE SPRING BY THE PROPOSED ALGORITHM

1. Characteristics of the Geometry Design Parameters for the M Type Spring

The stress on the M type spring mainly concentrates at the corners. There are 8 variables for the geometry design optimization of the M type spring. The initial value and search range for each are listed in Table 7.

The relation between the maximum concentrated stress created on the M type spring and its geometry parameters cannot be formulated by mathematical equations to determine if the parameters are correlated. In the parameters' search ranges, we take the initial parameters as references, and repeatedly divide each search range by 11 points to analyze the M type spring movement. The characteristic curves of the maximal concentrated stresses on these 11 points are depicted in Figs. 11 and 12. From the curves in Figs. 11 and 12, we can see that the geometry parameters are correlated, which means they are not independent variables.

2. Mathematical Model for the Design Optimization of the M Type Spring

Consider the relation between the sliding piece and the spring. The thrust and elastic force are set to ±10% of original spring. The objective function is to lower the maximum von Mises stress $\sigma_{\max}$. The mathematical model of this problem is as follows.

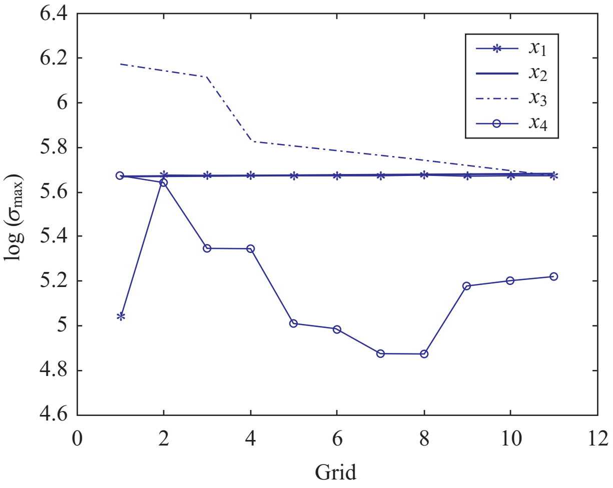

Fig. 11. The maximal concentrated stress for the 1st~4th parameters.

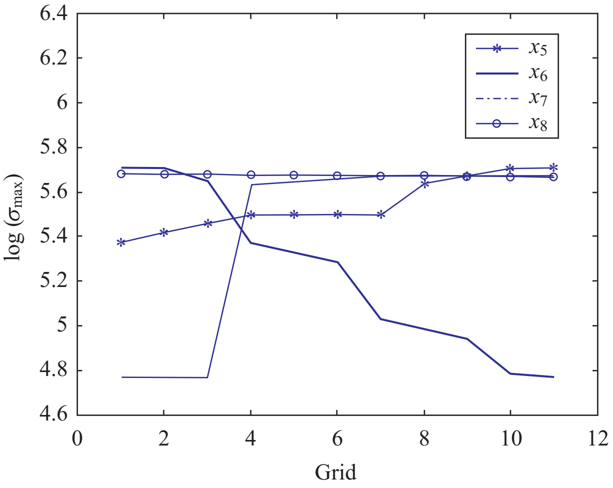

Fig. 12. The maximal concentrated stress for the 5th~8th parameters.

$$\text{Minimize: } f(x) = \sigma_{\max} \tag{5}$$

Subject to:

$$g_1 = {|PF-116|}\big/{116} \le 10\%,\ g_2 = {|BO-95|}\big/{95} \le 10\% \tag{6}$$

Where: *PF* is the maximum thrust force and *BO* is the maximum elastic force.

We use a penalty function to deal with the constraints and transform the above problem into an unconstrained optimization problem, as follows.

$$f(x) = \sigma_{\max} + R \times \{\min[0, g_1] + \min[0, g_2]\} \tag{7}$$

R is the penalty factor. In this paper, *R* = 200000.

Table 8. optimal value of design variables.

Design	A1	A2	A3	L1
Optimal	88 deg	2 mm	26.2 deg	2.9 mm
Design	L2	R1	R2	R3
Optimal	12.2 mm	1.6 mm	2.1 mm	31.3 mm

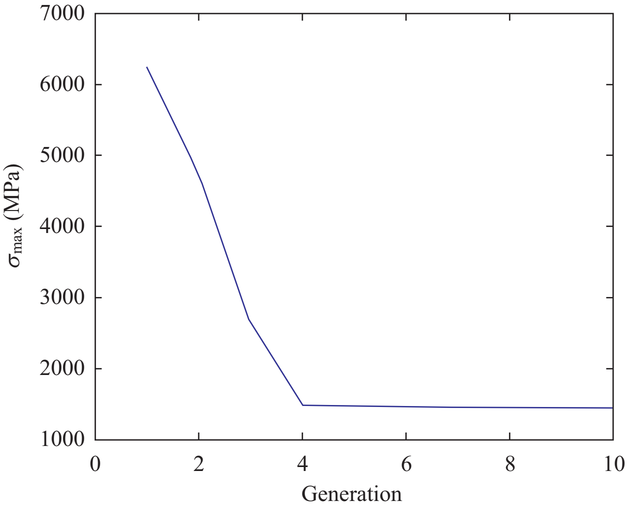

Fig. 13. Convergence of optimal result for M type spring.

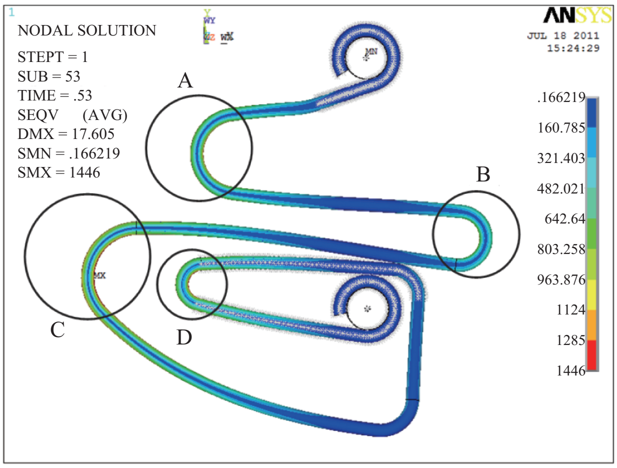

Fig. 14. von Mises stress distribution when the slider moves to 15.9 mm.

MPSO parameters are:

(a) Number of populations (NP) = 25.
(b) Inertia weight (w) = 0.6.
(c) Acceleration constants $C_1 = 1.5$, $C_2 = 2$.
(d) Number of dividing points (N_d) = 5.
(e) Range shrink rate of iterative grid method (ρ) = 0.98.
(f) Maximum iterations of iterative grid method ($CGM_{iter\max}$) = 1.
(g) The generations that apply iterative grid method (N_E) = 2.
(h) The convergence criteria is that maximum iterations ($iter_{\max}$) = 20 or $\sigma_{\max} \leq 1150$ Mpa.

Table 9. The maximal stress before and after optimization in each region and the relative improvements.

Unit: MPa	Region A	Region B	Region C	Region D
Initial (ratio to the maximal stress)	1292.8 (0.81)	735.6 (0.46)	1592 (1)	1375.3 (0.86)
Optimal (ratio to the maximal stress)	1163.8 (0.80)	874.4 (0.604)	1446 (1)	1374.1 (0.95)
Improvement	-10.0%	18.8%	-9.2%	-0.08%

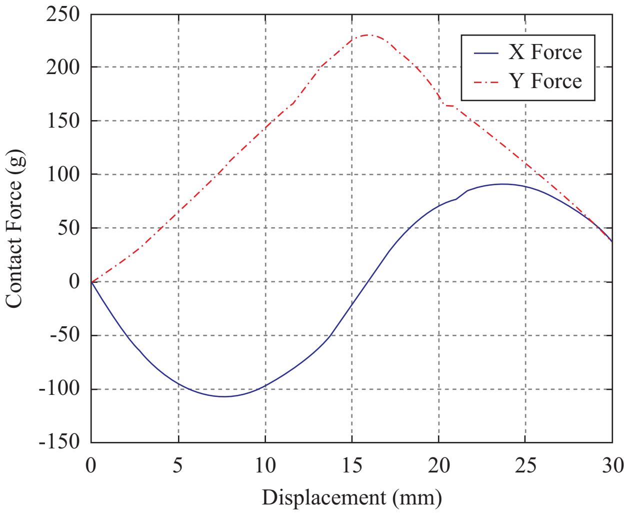

Fig. 15. Displacement and reaction force of the slider spring in X and Y direction.

(i) The search range of each design variable is listed in Table 7.

Table 8 shows the determined optimal design parameters. Fig. 13 illustrates the search convergence. After six generations, the evolving converges and cannot be updated. Fig. 14 shows the von Mises stress diagram for the optimal design. The M type spring has maximum stress at 15.9 mm. The maximum stress concentrates on the inner surface of the rounded corner in region C with a magnitude of 1446 MPa, which is 9.2% lower than the original spring.

Table 9 shows the stresses before and after optimization in each region and the improvements. We can see from the table that all the high stress regions have decreased in stress and the stress distributions are more uniform. Fig. 15 shows the reaction force curves of the M type spring in X and Y direction. The maximal thrust is 107.2 g, the maximal elastic force is 91.6 g, and the maximal counter-force in Y direction is 230.4 g. The volume of the spring goes from 16.34 mm^3 to 16.48 mm^3, which is about 0.85% increase, as shown in Fig. 16.

V. CONCLUSION

By incorporating a grid method with the PSO, we developed

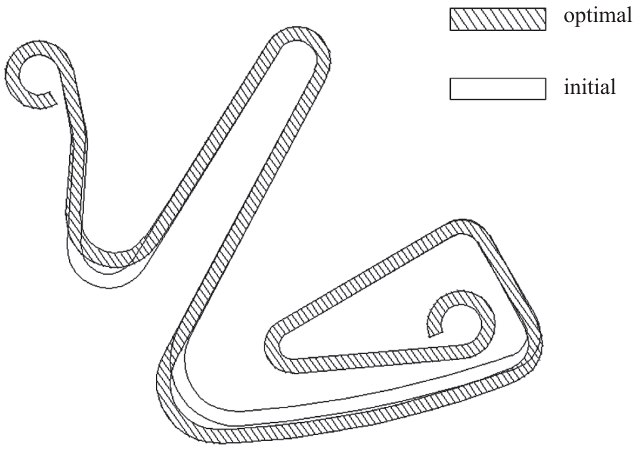

Fig. 16. Comparison of Initial and Optimal Shape.

a new variant of the original PSO. The developed algorithm is a robust one giving a success rate of 100% when it is applied on each of seven tested benchmark functions. It overcomes the difficulty met by the original PSO when a non-separable or a multi-modal function is solved or when the number of variables of the problem is increased.

To verify the effectiveness of the proposed algorithm in a practical application, the optimal design of a M type spring used in the 3C equipments was selected for the demonstration. There are eight variables in the geometry design of the spring. The optimal solution of the eight variables of the spring giving the maximum strength is determined by the algorithm. In the calculation, the ANSYS APDL is used to create the finite element model for the M type spring and determine the relevant forces, such as the counter-force, the elastic force and directions of forces.

The geometry parameter optimization for the M type spring shows that when the thrust is larger than 105 g and elastic force larger than 90 g, the maximum von Mises stress decreases about 9.2% from 1592 MPa to 1446 Mpa.

REFERENCES

Acan, A. and A. Gunay (2005). Enhanced particle swarm optimization through external memory support. IEEE, 235-242.

ANSYS, Inc. (2012). Release 14.5 Documentation for ANSYS.

Chatterjee, A. and P. Siarry (2004). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Computers and Operations Research 33, 859-871.

Chen, Y.-J. (2007). Stress analysis and dimension parameter design of slide hinge. Master's Thesis, Department of Mechanical Engineering, National Taiwan University of Science and Technology.

Clerc, M. and J. Kennedy (2000). The particle swarm: explosion, stability, and convergence in a multimodal complex space. Proceedings of the Congress of Evolutionary Computation, Washington, D.C., IEEE, Piscataway, NJ,USA, 6, 58-73.

Eberhart, R. C. and Y. Shi (2001). Tracking and optimizing dynamic systems with particle swarms. Proceedings of the Congress on Evolutionary Computation 1, Soul, IEEE Press, 94-100.

Jasbir, S. A. (1989). Introduction to Optimum Design, McGraw Hill, Inc.

Lin, M.-C. (2007). The research of slide phone mechanism. Master's Thesis, Department of Material and Manufacture Engineering, Feng Chia University.

Liu, B., L. Wang, Y.-H. Jin, F. Tang and D.-X. Huang (2005). Improved particle swarm optimization combined with chaos, SCI Chaos. Solitons and Fractals 25, 1261-1271.

Mohammed, EI-Abd (2011). A hybrid ABC-SPSO algorithm for continuous function optimization. Swarm Intelligence (SIS), 2011 IEEE Symposium on 978-1-61284-053-6, IEEE Press.

S. Fan, S.-K. and E. Zahara (2006). A hybrid simplex search and particle swarm optimization for unconstrained optimization. European Journal of Operational Research 181, 527-548.

Suganthan, P. N. (1999). Particle Swarm Optimiser with Neighbourhood Operator, Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, IEEE, Piscataway, NJ, USA, 3, 1958-1962.

Tsai,Y.-H. (2008). Life analysis for spring of slide hinge. Master's Thesis, Department of Mechanical Engineering, National Taiwan University of Science and Technology.

Wang, H., H. Sun, C. Li, S. Rahnamayan and J.-S. Pan (2013). Diversity enhanced particle swarm optimization with neighborhood search. Information Sciences 223, 119-125.

Yin, P. Y., F. Glover, M. Laguna and J.-X. Zhu (2010). Cyber swarm algorithms-improving particle swarm optimization using adaptive memory strategies. European Journal of Operational Research 201(2), 377-389.

Zhang, J.-M. (2009). Optimal design of spring structure on slide mechanism of mobile using genetic algorithms. Master's Thesis, Department of Mechanical and Automation Engineering, National Kaohsiung First University of Science and Technology.

Zhao, M., J.-S. Pan and S.-T. Chen (2015). Optimal SNR of audio watermarking by wavelet and compact PSO methods. Journal of Information Hiding and Multimedia Signal Processing 6(5), 833-846.