Mass-spring-damper Motion Dynamics-based Particle Swarm Optimization

Ki-Baek Lee and Jong-Hwan Kim

Abstract— **Mass-spring-damper motion dynamics-based particle swarm optimization (MMD-PSO) is a novel optimization paradigm based on motion dynamic model which consists of mass, spring and damper. In MMD-PSO some particles, which are located fitter places than other particles, drop their anchor and connect springs and dampers between the anchors and all the particles. These connections influence the movements of the particles so as to proceed to fitter places attracted by the anchors. To demonstrate the effectiveness of MMD-PSO, several experiments are carried out on numerical optimization problems with complex test functions. The results show that proposed MMD-PSO is more powerful than original PSO and PSO mass-spring analogy in terms of robustness and convergence speed with no tuning parameters.**

I. Introduction

Problems which need global optimization frequently arise in practical engineering applications. These problems usually require controller parameters satisfying desired well-known time domain specification such as rise time, settling time and percent-overshoot. The optimization process starts from modeling the system under control and designing an objective function based on the time-domain specification each value of which item depends on the controller parameters. The objective function can be successfully modeled in a simple problem; however, it is often difficult or impossible to get it in a closed form in case of the function is nonlinear and non-differentiable.

In this case, stochastic optimization techniques such as evolutionary programming (EP), evolution strategies (ES) and genetic algorithm (GA), can be applied because of their robustness compared to the traditional optimization methods such as calculus-based and enumerative strategies [1], [2], [3], [4]. However, in spite of their advantages, such evolutionary algorithms (EAs) require considerable computing power and need various considerations and tunings for implementation. Particle swarm optimization (PSO), which models a bird flock or a fish swarm, can be also used for this purpose [5], [6]. PSO shows faster convergence speed even with less number of computations than GA [7]. However, conventional PSO also requires design parameters to be tuned.

The goal of this paper is to upgrade the conventional PSO such that it can find global solution as fast as possible without any tuning parameters. Proposed PSO algorithm is based on the motion dynamics of mechanical system modelled by mass-spring-damper. After particles are initially distributed, they explore the search space following the mass-spring-damper motion dynamics model. Anchors are placed on the several current fittest positions, and the particles are pulled by springs and dampers connected to the anchors. As generation proceeds, eventually the particles gather to the optimal solution. Its computation time is relatively shorter than that of other algorithms. In addition, proposed algorithm shows robustness against local minimum problems. The effectiveness of the algorithm is verified by carrying out experiments on well-known test functions.

The remainder of this paper is organized as follows. In Section II, MMD-PSO along with its terminology is described in detail. Section III presents the experiment results on test functions including De Jong functions, Griewangk's function, Rastrigin's function, Ackley's function, and Schwefel's (sine root) function. The results are compared with those by original PSO and PSO mass-spring analogy [7], [8]. Finally, conclusions and further works follow in Section IV.

Ki-Baek Lee and Jong-Hwan Kim are with the Department of Electrical Engineering and Computer Science, Korea Avdanced Institute of Science and Technology, Daejon, Republic of Korea, (phone: +82-42-869-5448; email: {kblee, johkim}@rit.kaist.ac.kr).

II. Proposed PSO Algorithm

A. Terminology

Let us assume that the i-th particle has the mass, m_i, which follows the following Newton's second law:

$$F_i = m_i \ddot{x}_i \tag{1}$$

where F_i is the force of the i-th particle and x_i is the position of the i-th particle.

An anchor is a heavy object used to attach something at a specific point. Also, let us introduce a concept of anchor, which is a heavy object at a specific fixed point used to fasten something securely. In this paper, it is assumed that anchors have infinite mass and do not move.

A spring is a flexible elastic object used to store mechanical energy. Springs follow the Hooke's law, described as

$$F_{si} = -k_x d_{i0} \tag{2}$$

where F_{si} is the force exerted by the spring on the i-th particle, d_{i0} is the distance from the i-th particle to the origin and k_x is the modulus of elasticity. Assume springs have zero mass, and do not break.

A damper is a device that deadens, restrains, or depresses the motion of the particle. It is used to represent viscosity and satisfies the following equation:

$$F_{di} = -k_v \dot{x}_i \tag{3}$$

978-1-4244-1823-7/08/$25.00©2008 IEEE

where F_{di} is the force exerted by the damper on the i-th particle, x_i is the position of the i-th particle and k_v is the modulus of viscosity. Dampers are also assumed to have zero mass.

B. Configuring search space and fitness values

Search space $\mathbb{S}$ is defined as

$$\mathbb{S} \subset \mathbb{R}^n$$

and the position of the i-th particle in the search space is represented by the following vector:

$$x_i = [x_{i_1} \;\; x_{i_2} \;\; x_{i_3} \;\; \ldots \;\; x_{i_n}]^T \in \mathbb{S}.$$

At the position of the i-th particle, the fitness value f_i is defined as

$$f_i = f(x_i). \tag{4}$$

The greater the fitness value, the fitter the position of the fitness value. The higher the fitness value, the fitter the position of the particle. Therefore, the goal of the searching process is to find the position of the particle with the highest fitness value.

C. Dynamic motion model of the particle

Figure 1 shows a particle connected to an anchor with a spring and a damper. Let us assume that there are N particles

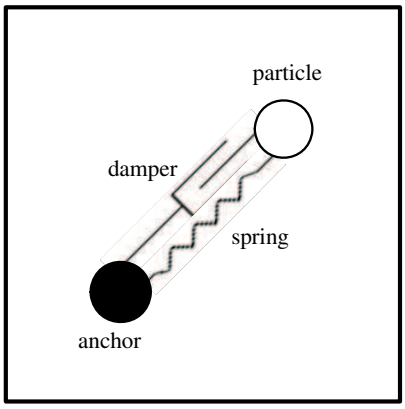


Fig. 1. Particle,anchor,spring and damper

and M anchors. Net force F_i acting on the i-th particle can be derived by (2) and (3) as follows:

$$\begin{aligned} F_i &= F_{si} + F_{di} \\ &= -\sum_{j=1}^{M} (k_j^a d_{ij} + k_i \dot{x}_i) \end{aligned} \tag{5}$$

where k_j^a is the modulus of elasticity of the spring from the j-th anchor, k_i is the modulus of viscosity of the damper to the i-th particle and d_{ij} is the distance between the i-th particle and the j-th anchor. The distance is defined as

$$d_{ij} = x_i - x_j^a$$

where x_j^a is the position of the j-th anchor. Then, (5) can be rewritten as

$$F = \sum_{j=1}^{M} \left(k_j^a (x_j^a - x_i) - k_i \dot{x}_i \right). \tag{6}$$

By substituting (6) to (1), the acceleration of the i-th particle is calculated as

$$\ddot{x}_i = \frac{\sum_{j=1}^{M} \left((k_j^a (x_j^a - x_i) - k_i \dot{x}_i \right)}{m_i}. \tag{7}$$

Above equation can be rewritten in a discrete time representation as follows:

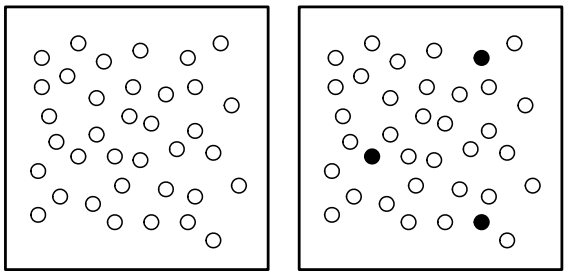
$$v_i[k+1] - v_i[k] = \frac{\sum_{j=1}^{M} \left(k_j^a (x_j^a[k] - x_i[k]) - k_i v_i[k] \right)}{m_i} \tag{8}$$

where $v_i[k]$ is the velocity of the i-th particle at k-th generation and $x_i[k]$ is the position of the i-th particle at k-th generation. By setting the value of mi unit, the update rules for Mass-spring-damper motion dynamics-based particle swarm optimization can be derived as

$$\begin{cases} v_i[k+1] &= v_i[k] + \sum_{j=1}^{M} \left(k_j^a (x_j^a[k] - x_i[k]) - k_i v_i[k] \right) \\ x_i[k+1] &= x_i[k] + v_i[k+1]. \end{cases} \tag{9}$$

D. Procedure of the proposed PSO algorithm

N particles are first distributed at random according to uniform distribution in the search space. Figure 2(a) shows the initial positions of N particles.



(a) Initial positions of N particles (b) Selected M anchors (black circles) among N particles

Fig. 2. Particles and anchors

After initialization, the fitness at each position can be evaluated by (4). M best positions are selected and anchors are dropped at their position. Figure 2(b) shows the visualization of N particles and M anchors, where black circles indicate the anchors. As soon as anchors are dropped, springs and dampers are connected between each anchor and all the particles. In this paper, the modulus of elasticity of the spring connected from the j-th anchor, $j = 1, 2, \cdots, M$, is defined as

$$k_j^a = f_j^a \tag{10}$$

where f_j^a is the fitness value at the position of the j-th anchor. The physical meaning of (10) is that the anchor at

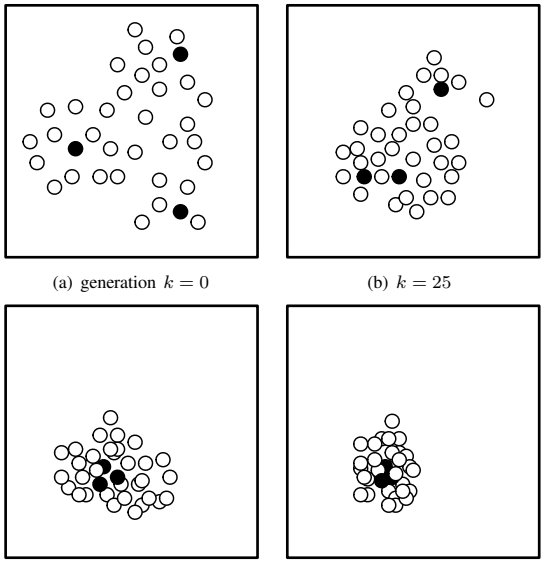
high fitness position gets strong springs. In other words, the anchor pull the particles strongly proportional to the fitness of its position. Similarly, the modulus of viscosity of the i-th particle, $i = 1, 2, \cdots, N$, is defined as

$$k_i = f_i \tag{11}$$

where f_i is the fitness value at the position of the i-th particle. (11) implies that the particle resists external forces as much as its fitness. Therefore, (9) can be rewritten as

$$\begin{cases} v_i[k+1] = v_i[k] + \sum_{j=1}^{M} \left(f_j^a (x_j^a[k] - x_i[k]) - f_i v_i[k] \right) \\ x_i[k+1] = x_i[k] + v_i[k+1] \end{cases} \tag{12}$$

In each generation fitness values are calculated and new M best positions are selected to drop an anchor. This process is repeated until certain termination criterion is satisfied. Trajectories of the particles can be obtained by the update rules, (12). As these update rules show, there are no tuning parameters because fitness values are directly used for them. Figure 3 shows the movements of the particles as generation passes by.



(a) generation $k = 0$ (b) $k = 25$ (c) $k = 50$ (d) $k = 75$

Fig. 3. Movements of the particles

The generic pseudo-code of the proposed PSO is described as follows:

1) **initialize** the positions of N particles
2) **repeat**
   a) evaluate the particles for fitness values
   b) select M best positions
   c) drop an anchor at each of M position
   d) **for** each anchor a_i **do**
      i) **for** each particle p_j **do**
         A) connect a spring and a damper between a_i and p_j
         B) calculate the force exerted on p_j
         C) calculate the displacement of p_j
         D) **update** the position of p_j
3) **until** termination criterion is met

III. Experiment with complex test functions

A. Environment configuration

In this experiment, search space $\mathbb{S}$ was defined as

$$\mathbb{S} \subset \mathbb{R}^2$$

and the position vector x was represented as

$$x = [x_1 \; x_2]^T \in \mathbb{S}.$$

The number of particles, N was set to 100 (10×10), 169 (13×13) and 225 (15×15). The number of anchors, M was set to 4. The methods to be compared were original PSO and PSO mass-spring analogy (PSO-MSA). Update rules of original PSO and PSO-MSA were used, respectively, as follows:

- original PSO

$$\begin{cases} v_i[k+1] = & w \cdot v_i[k] + c_0 \cdot (x_i^{pBest} - x_i[k]) \\ & + c_1 \cdot (x^{gBest} - x_i[k] \\ x_i[k+1] = & x_i[k] + v_i[k+1] \end{cases}$$

where w, c_0, and c_1 are the tuning parameters of original PSO, v_i the velocity of the i-th particle, x_i the position of the i-th particle, x_i^{pBest} the best position of the i-th particle, and x^{gBest} the current global best position of the particles.

- PSO-MSA

$$\begin{cases} v_i[k+1] = & \alpha \cdot v_i[k] + C \cdot (x_i[k]) \\ x_i[k+1] = & \alpha v_i[k+1] + (1 - C) x_i[k] \end{cases}$$

where α and c are the tuning parameters of PSO-MSA. The values of tuning parameters of each method were set as shown in Table I. Note that MMD-PSO did not require any additional tuning parameters. All the three algorithms was terminated after one hundred generations.

Function	**Tuning parameters**		
Original PSO	W	C_0	C_1
	0.5	0.5	0.5
PSO-MSA	α	C	
	0.5	0.5	
MMD-PSO	None		

TABLE I
TUNING PARAMETER SETTING

B. Test functions

Following seven functions were selected as test functions:

1) First De Jong function [9]

$$f(x_1, x_2) = {x_1}^2 + {x_2}^2$$

$$(-10.0 < x_1, x_2 < 10.0)$$

2) Second De Jong function (Rosenbrock's saddle) [10]

$$f(x_1, x_2) = 100 \cdot ({x_1}^2 - x_1)^2 + (1 - x_1)^2$$

$$(-2.0 < x_1, x_2 < 2.0)$$

3) Fifth De Jong function (Shekel's Foxholes) [11]

$$f(x_1, x_2) = \frac{1}{0.002 + \sum_{i=1}^{25} \frac{1}{(i-1)+\sum_{j=1}^{2}(x_j - a_{i,j})^6}}$$

$$(-50.0 < x_1, x_2 < 50.0)$$

$a_{i,1}$ = (-32, -16, 0, 16, 32) for i = 1,2,3,4,5
$a_{i,1} = a_{i \bmod 5,1}$
$a_{i,2}$ = (-32, -16, 0, 16, 32) for i = 1,6,11,16,21
$a_{i,2} = a_{i+k,2}$, k=1,2,3,4

4) Griewangk's function [12]

$$f(x_1, x_2) = \frac{{x_1}^2}{4000} + \frac{{x_2}^2}{4000} - cos(x_1) \cdot cos(\frac{x_2}{2}) + 1$$

$$(-10.0 < x_1, x_2 < 10.0)$$

5) Rastrigin's function [12], [13]

$$f(x_1, x_2) = 20.0 + {x_1}^2 - 10.0 \cdot cos(2\pi x_1)$$

$$+{x_2}^2 - 10.0 \cdot cos(2\pi x_2)$$

$$(-6.0 < x_1, x_2 < 6.0)$$

6) Ackley's function [14], [15]

$$f(x_1, x_2) = -20.0 \cdot e^{-0.2\sqrt{\frac{{x_1}^2+{x_2}^2}{2}}}$$

$$-e^{\frac{cos(2\pi x_1)+cos(2\pi x_2)}{2}} + 20.0 + e$$

$$(-50.0 < x_1, x_2 < 50.0)$$

7) Schwefel's (sine root) function [16]

$$f(x_1, x_2) = 2 \cdot 418.9829 - x_1 sin(\sqrt{|x_1|})$$

$$-x_2 sin(\sqrt{|x_2|})$$

$$(-500.0 < x_1, x_2 < 500.0)$$

C. Experiment results

In experiments, each algorithm was performed 50 times for each test function with the same number of generations of 100. Table II shows the summary of the experimental results. For each case of the number of particles, MMD-PSO is dominant. With sufficiently large number of particles, MMD-PSO is dominant as shown in Table II(c). In addition to the best solutions, the worst solutions of the MMD-PSO were the smallest for all the test functions. In particular, in the cases of second De Jong function, Griewangk's function, Rastrigin's function and Schewefel's function, original PSO and PSO-MSA tended to approach the local minima. However, MMD-PSO successfully avoided the local minima and could reach the global solution. Mean and median values of MMD-PSO were much smaller than those of other methods. Relatively small variance of MMD-PSO also verifies that it is much robuster than other ones.

Figure 4 shows the comparison results from the view point of convergence speed. For three test functions such as first De Jong, fifth De Jong, and Ackley's function, the algorithms showed similar convergence speed. Thus, the rusults for the rest four functions such as second De Jong, Griewangk's, Rastringin's, and Schewefel's function, were compared as shown in Figure 4. In the figure, X and Y axes represent the number of generations and the best value for each test function at that generation, respectively. Smaller values mean better results since the functions are fit to minimization problems. In other words, the faster it reduces, the better the result is. Among the four figures, Figure 4(a), Figure 4(c) and Figure 4(d) show that MMD-PSO has considerably faster convergence rate than the others. In the case of Figure 4(b), MMD-PSO and original PSO showed similar converged rate.

From the results, it can be concluded that MMD-PSO can find global solution more accurately than original PSO and PSO-MSA. In addition, MMD-PSO shows better convergence property than the others.

IV. Conclusions

This paper proposed a novel optimization method, MMD-PSO, based on motion dynamics model, which consisted of mass, spring, damper. For verifying the performance of MMD-PSO, seven complex test functions were used. The functions were first De Jong function, second De Jong function, fifth De jong function, Griewangk's function, Rastrigin's function, Ackley's function and Schwefel's (sine root) function. Experimental results demonstrated that the performance of MMD-PSO was competitive. Most of all, MMD-PSO did not require any tuning parameters which are indispensable to original PSO and PSO-MSA. In addition, MMD-PSO could approach the global best with better solution accuracy than other comparison methods. The convergence rate of MMD-PSO was also faster than others.

References

[1] D. E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*.

TABLE II

COMPARISON RESULTS OF TEST FUNCTIONS WITH METHODS OF ORIGINAL PSO, PSO-MSA, AND PROPOSED MMD-PSO

(a) $N = 100$

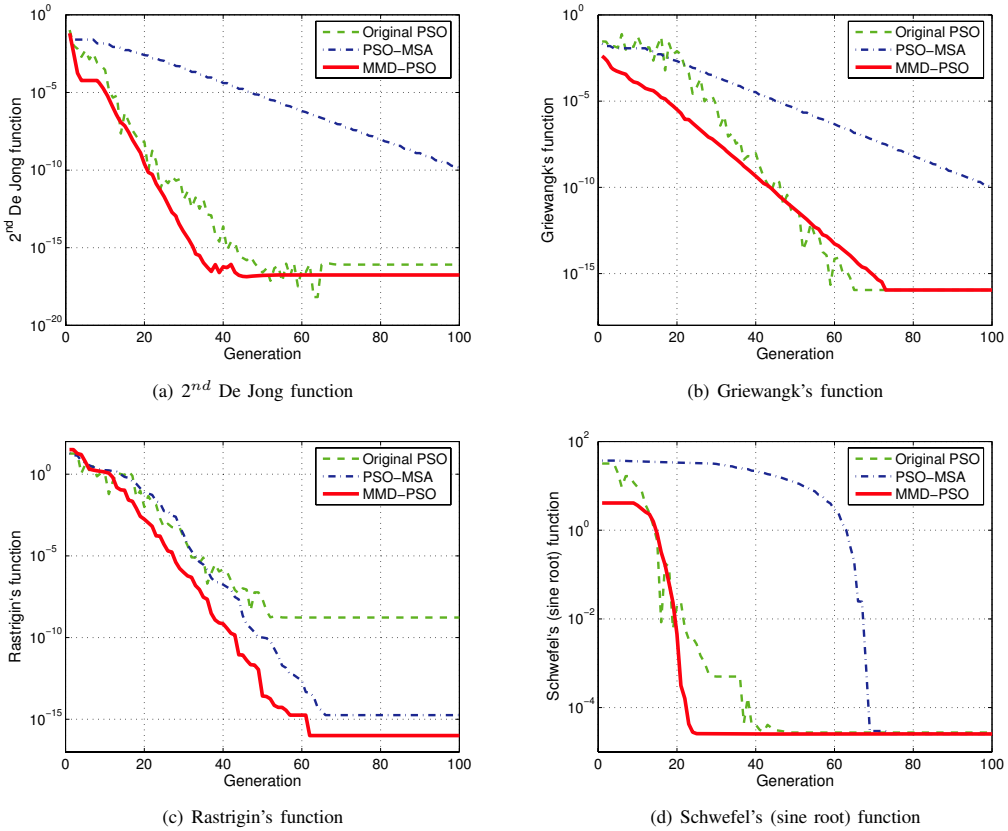
	Method \ Metric	Worst	Best	Mean	Median	Variance
1^{st} De Jong	Original PSO	1.10e-16	7.46e-18	6.81e-17	6.69e-17	1.12e-33
	PSO-MSA	1.31e-15	1.09e-16	1.71e-16	1.10e-16	7.17e-32
f_1	**MMD-PSO**	7.31e-16	4.40e-18	1.36e-16	1.01e-16	3.47e-32
2^{nd} De Jong	Original PSO	3.09e-16	4.03e-18	6.13e-17	4.86e-17	4.38e-33
	PSO-MSA	5.44e+03	1.09e-16	4.86e+02	1.59e-16	2.27e+06
f_2	**MMD-PSO**	1.08e-16	3.25e-17	8.16e-17	9.05e-17	7.16e-34
5^{th} De Jong	Original PSO	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
	PSO-MSA	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
f_3	**MMD-PSO**	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
Griewangk's	Original PSO	1.23e-02	1.11e-16	6.66e-03	9.86e-03	3.20e-05
	PSO-MSA	1.23e-02	1.11e-16	3.70e-03	1.11e-16	2.72e-05
f_4	**MMD-PSO**	9.86e-03	1.00e-20	1.48e-03	1.11e-16	1.31e-05
Rastrigin's	Original PSO	1.09e+00	1.00e-20	1.04e-01	1.00e-20	1.04e-01
	PSO-MSA	7.43e-02	1.00e-20	3.71e-03	2.63e-14	2.76e-04
f_5	**MMD-PSO**	7.71e-13	1.00e-20	4.26e-14	1.00e-20	2.97e-26
Ackley's	Original PSO	5.89e-16	5.89e-16	5.89e-16	5.89e-16	1.03e-62
	PSO-MSA	7.67e-07	1.21e-10	8.68e-08	1.99e-09	3.68e-14
f_6	**MMD-PSO**	7.21e-13	5.89e-16	3.66e-14	5.89e-16	2.59e-26
Schwefel's	Original PSO	1.19e+02	2.55e-05	3.03e+01	2.55e-05	2.74e+03
	PSO-MSA	1.19e+02	2.55e-05	3.56e+01	2.55e-05	3.12e+03
f_7	**MMD-PSO**	1.19e+02	2.55e-05	2.37e+01	2.55e-05	2.37e+03

(b) $N = 169$

	Method \ Metric	Worst	Best	Mean	Median	Variance
1^{st} De Jong	Original PSO	2.71e-15	7.46e-18	2.14e-17	1.09e-16	3.46e-31
	PSO-MSA	1.31e-15	1.52e-17	1.60e-16	1.10e-16	7.39e-32
f_1	**MMD-PSO**	6.10e-16	3.30e-17	1.35e-16	1.08e-16	1.88e-32
2^{nd} De Jong	Original PSO	5.28e-12	4.03e-18	1.64e-13	6.76e-17	1.30e-33
	PSO-MSA	7.44e+03	2.91e-17	5.86e+02	1.10e-16	3.52e+06
f_2	**MMD-PSO**	4.78e-14	3.53e-17	2.49e-15	1.07e-16	1.14e-28
5^{th} De Jong	Original PSO	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
	PSO-MSA	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
f_3	**MMD-PSO**	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
Griewangk's	Original PSO	1.23e-02	1.11e-16	5.08e-03	4.95e-03	2.74e-05
	PSO-MSA	9.92e-03	1.11e-16	1.98e-03	1.11e-16	1.64e-05
f_4	**MMD-PSO**	9.86e-03	1.00e-20	9.86e-04	1.11e-16	9.22e-06
Rastrigin's	Original PSO	9.35e-01	1.00e-20	4.67e-02	1.00e-20	4.37e-02
	PSO-MSA	9.95e-03	1.00e-20	4.97e-03	2.63e-14	4.95e-06
f_5	**MMD-PSO**	7.71e-13	1.00e-20	5.74e-14	1.00e-20	3.47e-26
Ackley's	Original PSO	5.89e-16	5.89e-16	5.89e-16	5.89e-16	1.03e-62
	PSO-MSA	7.67e-07	1.21e-10	1.48e-07	1.06e-08	6.28e-14
f_6	**MMD-PSO**	9.41e-13	5.89e-16	1.20e-13	5.89e-16	8.62e-26
Schwefel's	Original PSO	1.19e+02	2.55e-05	1.84e+01	2.55e-05	2.33e+03
	PSO-MSA	1.19e+02	2.55e-05	2.42e+01	2.55e-05	2.36e+03
f_7	**MMD-PSO**	1.18e+02	2.55e-05	1.78e+01	2.55e-05	1.88e+03

(c) $N = 225$

	Method \ Metric	Worst	Best	Mean	Median	Variance
1^{st} De Jong	Original PSO	1.10e-16	9.46e-18	5.56e-17	5.24e-17	8.81e-34
	PSO-MSA	1.11e-16	1.09e-16	1.10e-16	1.10e-16	3.23e-37
f_1	**MMD-PSO**	1.10e-16	3.20e-18	7.12e-17	6.81e-17	1.19e-33
2^{nd} De Jong	Original PSO	1.09e-16	1.55e-18	5.03e-17	4.81e-17	1.30e-33
	PSO-MSA	1.09e+04	1.09e-16	7.58e+02	1.10e-16	6.58e+06
f_2	**MMD-PSO**	1.08e-16	1.20e-17	8.00e-17	9.05e-17	8.94e-34
5^{th} De Jong	Original PSO	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
	PSO-MSA	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
f_3	**MMD-PSO**	9.98e-01	9.98e-01	9.98e-01	9.98e-01	5.19e-32
Griewangk's	Original PSO	1.23e-02	1.11e-16	6.78e-03	9.86e-03	3.31e-05
	PSO-MSA	1.23e-02	1.11e-16	2.22e-03	1.11e-16	2.10e-05
f_4	**MMD-PSO**	1.11e-16	1.00e-20	1.05e-16	1.11e-16	6.16e-34
Rastrigin's	Original PSO	9.95e-01	1.00e-20	9.95e-02	1.00e-20	9.38e-02
	PSO-MSA	5.13e-02	1.00e-20	2.56e-03	2.13e-14	1.31e-04
f_5	**MMD-PSO**	1.00e-20	1.00e-20	1.00e-20	1.00e-20	9.53e-34
Ackley's	Original PSO	5.89e-16	5.89e-16	5.89e-16	5.89e-16	1.03e-62
	PSO-MSA	4.38e-07	9.21e-11	4.03e-08	1.73e-09	1.06e-14
f_6	**MMD-PSO**	5.89e-16	5.89e-16	5.89e-16	5.89e-16	1.02e-62
Schwefel's	Original PSO	1.18e+02	2.55e-05	2.44e+01	2.55e-05	2.33e+03
	PSO-MSA	1.19e+02	2.55e-05	2.97e+01	2.55e-05	2.78e+03
f_7	**MMD-PSO**	1.18e+02	2.55e-05	1.18e+01	2.55e-05	1.33e+03



(a) 2^{nd} De Jong function

(b) Griewangk's function

(c) Rastrigin's function

(d) Schwefel's (sine root) function

Fig. 4. Comparison results from the view point of convergence speed ($N = 225$)

[2] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2nd ed. Piscataway, NY: IEEE Press, 2000.

[3] H.-P. Schwefel, *Evolution and Optimum Seeking*. New York: Wiley Inter-Science, 1995.

[4] J.-H. Kim and H. Myung, "Evolutionary programming techniques for constrained optimization problems," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 119–128, May 1997.

[5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 2001.

[6] F. Heppner and U. Grenander, "A stochastic nonlinear model for coordinated bird flocks," in *The Ubiquity of Chaos*, E. S. Krasner, Ed., Washington, DC: AAAS, 1990.

[7] B. Brandstatter and U. Baumgartner, "Particle swarm optimization-mass-spring system analogon," *IEEE Trans. on Magnetics*, vol. 38, no. 2, Mar. 2002.

[8] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *in Proc. IEEE Int. Conf. of Neural Networks*, vol. IV, Perth, Australia, 1995, pp. 1942–1948.

[9] K. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan, 1975.

[10] H. Rosenbrock, "An automatic method for finding the greatest or the least value of a function," *The Computer Journal*, vol. 3, no. 3, pp. 175–184, 1960.

[11] J. Shekel, "Test functions for multimodal search techniques," in *Fifth Annual Princeton Conf. on Information Science and Systems*, 1971.

[12] A. Törn and A. Zilinskas, "Global Optimization," *Lecture Notes in Computer Science*, vol. 350, 1989.

[13] H. Mühlenbein, D. Schomisch, and J. Born, "The Parallel Genetic Algorithm as Function Optimizer," *Parallel Computing*, vol. 17, no. 6-7, pp. 619–632, 1991.

[14] D. H. Ackley, *A connectionist machine for genetic hillclimbing*. Boston: Kluwer, 1987.

[15] T. Bäck, *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.

[16] H.-P. Schweful, *Numerical optimization of computer models*. Chicester: Wiley & Sons, 1981.