Visualizing the Search Process of Particle Swarm Optimization

Yong-Hyuk Kim
Department of Computer Science and Engineering
Kwangwoon University
Wolge-dong, Nowon-gu
Seoul, 139-701, Korea
yhdfly@kw.ac.kr

Kang Hoon Lee
Department of Computer Science and Engineering
Kwangwoon University
Wolge-dong, Nowon-gu
Seoul, 139-701, Korea
kang@kw.ac.kr

Yourim Yoon
School of Computer Science and Engineering
Seoul National University
Sillim-dong, Gwanak-gu
Seoul, 151-744, Korea
yryoon@soar.snu.ac.kr

ABSTRACT

It is a hard problem to understand the search process of particle swarm optimization over high-dimensional domain. The visualization depicts the total search process and then it will allow better understanding of how to tune the algorithm. For the investigation, we adopt Sammon's mapping, which is a well-known distance-preserving mapping. We demonstrate the usefulness of the proposed methodology by applying it to some function optimization problems.

Categories and Subject Descriptors

D.2.12 [**Software Engineering**]: Interoperability—*Data mapping*; G.1.2 [**Numerical Analysis**]: Approximation—*Nonlinear approximation*; I.3.m [**Computer Graphics**]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

Particle swarm optimization, visualization, data mapping

1. INTRODUCTION

Particle swarm optimization (PSO) is a new evolutionary algorithm [9, 12], of which the original intent was to simulate the choreography of a bird flock (see Figure 1) graphically. In PSO, each potential is seen as a particle with certain velocity flying through the domain. Each particle adjusts its flying according to the flying experiences of itself and its companions. PSO finds optimal regions of search spaces through the interaction of particles in population. It has been successfully applied to a number of optimization problems. Some studies showed that it often dominates genetic algorithms (GAs) [7, 13]. Since its appearance, PSO has gone through some improvements such as the inertia weighting factor and fine tuning [8, 22, 21].

Figure 1: Bird flock (photograph by Manuel Fresti)

The visualization method greatly enhances understanding and improves intuition. Understanding the search process of PSO is not an easy problem. However, its visualization is interesting and helps us to understand the search process. Visualization is one of the most basic tools for studies of search spaces. For the visualization of population-based search, the most popular method is the fitness flow over time as in many evolutionary algorithm papers. In this paper, we propose a new visualization technique for PSO, primarily using Sammon's mapping [19].

The remainder of this paper is organized as follows. In Section 2, we introduce previous visualization attempts for population-based evolutionary search such as genetic algorithms and particle swarm optimization. We also briefly review particle swarm optimization and Sammon's mapping. In Section 3, we present our approach based on Sammon's mapping. Sample experiments are presented in Section 4. Finally, we give some discussion in Section 5.

2. PRELIMINARIES

2.1 Prior Work Related to Population-Based Search

Traditionally, the fitness-iteration plotting has been popular for the visualization of evolutionary process. A great

GECCO'09, July 8–12, 2009, Montréal, Québec, Canada.

number of papers include these plottings to visualize their search process.

Although fitness-iteration plot is one of the most common methods, it is not the only one. Other methods have been proposed in various forms. Dybowski *et al.* [5] proposed a visualization method of GAs by using Sammon's mapping. There have been several studies about GA visualization using Sammon's mapping [3, 14, 18]. They presented initial studies about small or particular problems. An extensive survey of GA visualization techniques appeared in [10]. In this paper, we focus only on the visualization of PSO using Sammon's mapping.

The most intuitive visualization of PSO is to plot the positions of all the particles of the swarm per iteration. This method yields an animation of particles. In the case that the domain of PSO is 2 or 3-dimensional space, such visualization is straightforward. But when the domain is high-dimensional as in most cases, it is not so easy problem. We will treat such general cases neatly in this paper.

We also mention existing work [1, 16, 23] on diversity measures that are often used for combinatorial optimization problems which have finite domains. In particular, the *resampling ratio* is used to measure the number of times the same solution is recomputed, and the *similarity ratio* is used to measure the average diversity of a set of solutions. This kind of measures could be adapted to deal with continuous domains, e.g., by splitting continuous domains into discrete interval sets, and could be used to provide an insight into a PSO search process by quantifying intensification and diversification. There have been other studies for PSO visualization. There was a study using some derived statistics to represent the convergence status of the swarm [2]. We can also find an enhanced histogram visualization technique in [20]. However, these methods have made limited success.

2.2 Particle Swarm Optimization

In this subsection, we briefly describe the standard PSO algorithm. A swarm is a population of particles. Each particle is treated as a point in m-dimensional bounded space X. Each particle i has position $\boldsymbol{x}_i$, velocity $\boldsymbol{v}_i$, and the best previous position $\boldsymbol{b}_i$ ($\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{b}_i \in X$). Let $\boldsymbol{g}$ be the global best particle. The performance of each particle is measured according to a predefined fitness function. Initially, each particle has a random position ($\boldsymbol{x}_i \in U[\boldsymbol{x}_{min}, \boldsymbol{x}_{max}]$) and velocity zero, i.e., $\boldsymbol{v}_i = \boldsymbol{0}$. The particles are updated by the following formula.

$$\boldsymbol{v}_i \leftarrow w \cdot \boldsymbol{v}_i + c_1 \cdot \boldsymbol{r}_1 \cdot (\boldsymbol{b}_i - \boldsymbol{x}_i) + c_2 \cdot \boldsymbol{r}_2 \cdot (\boldsymbol{g} - \boldsymbol{x}_i)$$

$$\text{and} \quad \boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i,$$

where w, c_1, and c_2 are constants, and $\boldsymbol{r}_1$, $\boldsymbol{r}_2 \in U[\boldsymbol{0}, \boldsymbol{1}]$. Then, $\boldsymbol{b}_i$s and $\boldsymbol{g}$ are updated. The inertia weight w provides a balance between global and local exploration. The acceleration constants c_1 and c_2 represent the weights of the stochastic acceleration terms that pull each particle $\boldsymbol{x}_i$ toward $\boldsymbol{b}_i$ and $\boldsymbol{g}$, respectively. Small values make particles roam far from target regions before being tugged. But, large values result in abrupt movement toward that regions.

2.3 Sammon's Mapping

Sammon's mapping [19] is a mapping technique for transforming a dataset from a high-dimensional (say, m-dimensional) input space onto a low-dimensional (say, d-dimensional) output space (with $d < m$). The basic idea is to arrange all the data points on a d-dimensional output space in such a way that minimizes the distortion of the relationship among data points. The distances between the data points in the output space resemble the distances in the input space defined by given metric as faithfully as possible. The resultant output space depicts clusters of the input space as groups of data points mapped close to each other in the output space.

Sammon's mapping tries to preserve distances of the original domain. This is achieved by minimizing an error criterion which penalizes the differences of distances between points in the input space and the output space. Consider a dataset of n objects. If we denote the distance between two points x_i and x_j in the input space by δ_{ij} and the distance between x'_i and x'_j in the output space by d_{ij}, then Sammon's stress measure E is defined as follows:

$$E = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta_{ij}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}.$$

The stress range is [0,1] with 0 indicating a lossless mapping. This stress measure can be minimized using any minimization technique. Sammon [19] proposed a technique called pseudo-Newton minimization, a steepest-descent method. The complexity of Sammon's mapping is $O(n^2 m)$ because of computing its stress function. There were several studies about Sammon's mapping [4, 6, 17].

3. PROPOSED APPROACH

In this section, we present a new visualization method. The method produces an animation of particles by plotting the projected 2-dimensional (2D) positions of all the particles in the swarm per iteration. In the case that the domain of PSO has two or three dimension, since the representation itself of each particle means its position in output space, the visualization of the search process of PSO is a quite easy task. But most optimization problems have higher-dimensional domain. In such general cases, the visualization of particles becomes a non-trivial task. We focus on the visualization of particles on high-dimensional space.

Figure 2 depicts a sketch of our visualization. All the particles together with the global best are presented in 2D space. Their traces are also indicated. As a PSO search proceeds, we can check the convergence of particles immediately through this visualization. In the figure, seven particles are converging toward the global best during four iterations.

To transform particles from high-dimensional space onto 2D space, we adopt a well-known distance-preserving mapping, Sammon's mapping [19] given in Section 2.3. First, when a PSO starts, we map initial random particles onto 2D space by using the original Sammon's mapping. Then, we repeat the following procedure until the PSO terminates. Since each particle has its current position and next one reflecting its current velocity, to make the continuous movement of each particle, we map all the particles of next generation together with all current-generation ones onto 2D space by a variant of Sammon's mapping. Briefly speaking, this variant mapping fixes the 2D positions of particles of current generation and determines only those of next generation. Then, each particle can move without discontinuity. However, the variant mapping may produce large error because of fixing a half of total points. We can check the success degree of the variant mapping with Sammon's stress

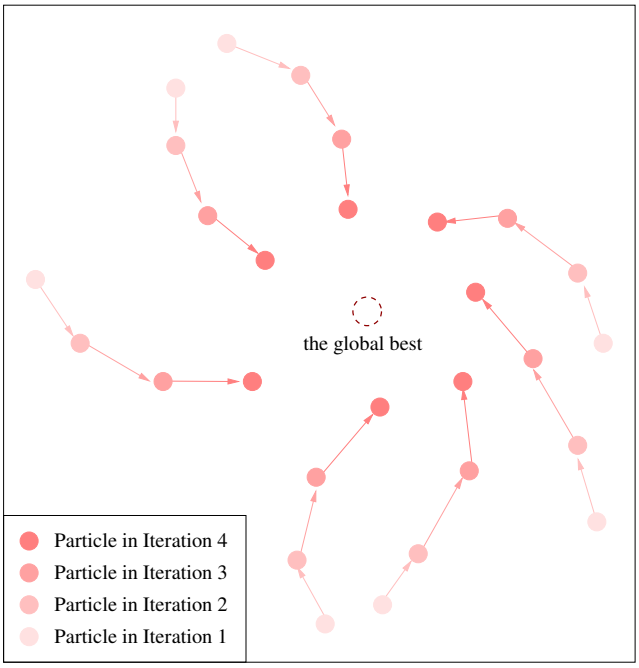


Figure 2: Schematic form of the proposed visualization

Map initial n particles onto 2D output space
by the original Sammon's mapping;
Output n particles of current generation onto 2D display;
do
{
Map n particles of next generation
together with those of current generation
onto 2D output space by our variant mapping†;
Efface n particles of current generation
from 2D display leaving their traces;
Output next n particles onto 2D display;
Replace n particles of of current generation
with those of next generation;
} **until** (PSO stops);

† See Figure 4.

Figure 3: Framework of the proposed visualization

measure as well. The closer to 0 the measure, the more successful the mapping. In most cases, we could obtain successful mappings with about 0.1 or below as the Sammon's stress measure. Figure 3 gives a pseudo-code of our visualization procedure and Figure 4 shows the details of our variant of Sammon's mapping.

4. SIMULATION

As a sample problem to apply our visualization, we chose the function optimization problem. We used three test functions (F_{Sphere}, $F_{\text{Rastrigin}}$, and F_{Schwefel}) to minimize, which are given in Table 1. They are from [24] and have different levels of difficulty. F_{Sphere} is a unimodal function and

variantMapping(n, n particles of current generation
+ n particles of next generation)
{
for each $i = 1, 2, \ldots, 2n$
for each $j = 1, 2, \ldots, 2n$
$\delta_{ij} \leftarrow$ distance between point i and point j;
$\delta_{max} \leftarrow \max_{i,j} \delta_{ij}$;

for each $i = n+1, n+2, \ldots, 2n$
$(x_i, y_i) \in U[-\delta_{max}, \delta_{max}] \times [-\delta_{max}, \delta_{max}]$;

repeat T times
for each $i = n+1, n+2, \ldots, 2n$
$(e_x^1, e_y^1, e_x^2, e_y^2) \leftarrow (0, 0, 0, 0)$;
for each $j = 1, 2, \ldots, 2n$ $(j \neq i)$
$d_{ij} \leftarrow \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$;

if $d_{ij} \neq 0$, $\delta_{ij} \neq 0$, and $d_{ij} \neq \delta_{ij}$ **then**
// calculation of derivatives
$e_x^1 \leftarrow e_x^1 + (x_i - x_j)\left(\frac{1}{d_{ij}} - \frac{1}{\delta_{ij}}\right)$;
$e_y^1 \leftarrow e_y^1 + (y_i - y_j)\left(\frac{1}{d_{ij}} - \frac{1}{\delta_{ij}}\right)$;
$e_x^2 \leftarrow e_x^2 + \left(\frac{1}{d_{ij}} - \frac{1}{\delta_{ij}}\right) - \frac{(x_i - x_j)^2}{d_{ij}^3}$;
$e_y^2 \leftarrow e_y^2 + \left(\frac{1}{d_{ij}} - \frac{1}{\delta_{ij}}\right) - \frac{(y_i - y_j)^2}{d_{ij}^3}$;
if $e_x^2 \neq 0$ and $e_y^2 \neq 0$ **then**
$(x_i^t, y_i^t) \leftarrow (x_i, y_i) + \Delta \cdot \left(\frac{e_x^1}{|e_x^2|}, \frac{e_y^1}{|e_y^2|}\right)$;
else
$(x_i^t, y_i^t) \leftarrow (x_i, y_i)$;

// correction of n particles of next generation
for each $i = n+1, n+2, \ldots, 2n$
$(x_i, y_i) \leftarrow (x_i^t, y_i^t)$;

// Sammon's stress measure
$E \leftarrow \frac{1}{\sum_{i=1}^{2n-1} \sum_{j=i+1}^{2n} \delta_{ij}} \sum_{i=1}^{2n-1} \sum_{j=i+1}^{2n} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}$;
}

∗ We set Δ to be 0.2 in our experiments.
∗ We set T to be 1,000 in our experiments.
∗ n is the number of particles.
∗ Output positions of first n points are fixed during the run of this algorithm.

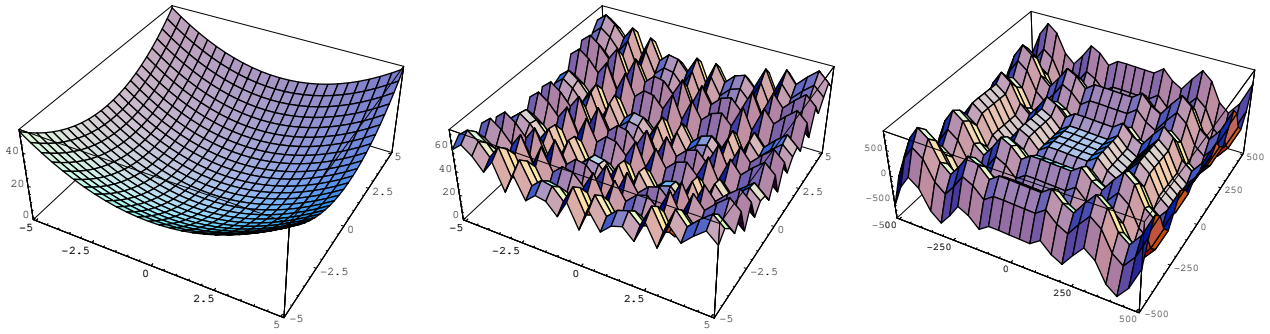
Figure 4: Pseudo-code of our variant Sammon's mapping

so very easy to find the minimum. $F_{\text{Rastrigin}}$ has the intermediate level of difficulty. F_{Schwefel} is the hardest among three functions. To give some intuition about test functions, the bottom of Table 1 also shows function graphs for 2-dimensional cases using Mathematica developed by Wolfram Research.

We used two PSOs varying movement strength. The parameters of PSO for slow movement (PSO-S) are as follows:

Table 1: Test Functions

Function	m	Range of x_i: $[l_i, u_i]$	Optimum	Optimal value
$F_{\text{Sphere}} = \sum_{i=1}^{m} {x_i}^2$	100	$[-5.12, 5.11]$	$(0, 0, \ldots, 0)$	0.00
$F_{\text{Rastrigin}} = \sum_{i=1}^{m} ({x_i}^2 - 10\cos(2\pi x_i) + 10)$	20	$[-5.12, 5.11]$	$(0, 0, \ldots, 0)$	0.00
$F_{\text{Schwefel}} = \sum_{i=1}^{m} -x_i \sin(\sqrt{\lvert x_i \rvert})$	10	$[-512, 511]$	$(420.968746, \ldots, 420.968746)$	-4189.828873



(a) F_{Sphere} $(m = 2)$ (b) $F_{\text{Rastrigin}}$ $(m = 2)$ (c) F_{Schwefel} $(m = 2)$

parameter	value
inertia weight: w	0.01
acceleration constant: c_1	0.01
acceleration constant: c_2	0.01
# of particles: n	20
# of iterations	1500

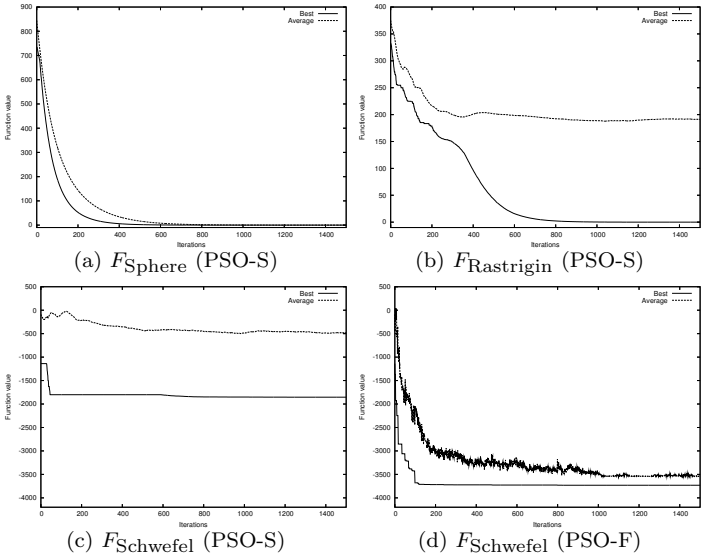
Those of PSO for fast movement (PSO-F) are in the following.

parameter	value
inertia weight: w	0.9
acceleration constant: c_1	1.2
acceleration constant: c_2	1.2
# of particles: n	20
# of iterations	1500

We applied PSO-S to the former two functions, F_{Sphere} and $F_{\text{Rastrigin}}$. Both of PSO versions, PSO-S and PSO-F, are applied to the hardest function, F_{Schwefel}.

Figure 5 shows traditional visualization results about our test runs. Each shows the best and average qualities over iterations. We can see that the best and average qualities of particles are improved as evolutionary step increases. We may guess the convergence degree of particles from the gap between the best and the average. But we do not clearly check the convergence, the interaction between particles, and the area searched by PSO, directly from these figures.

We visualized the two-dimensional motion of a set of particles that was acquired from our PSO experiments, by using a simple rendering routine based on OpenGL (see Figure 6). For each PSO experiment, we determine its viewing region according to the lowest and highest values of the two-dimensional locations that has been passed by moving particles throughout the entire evolutionary process. For each evolutionary step, each particle is drawn as a small filled circle '•' at its two-dimensional coordinate in the viewing region. We also display the positions of the global best and



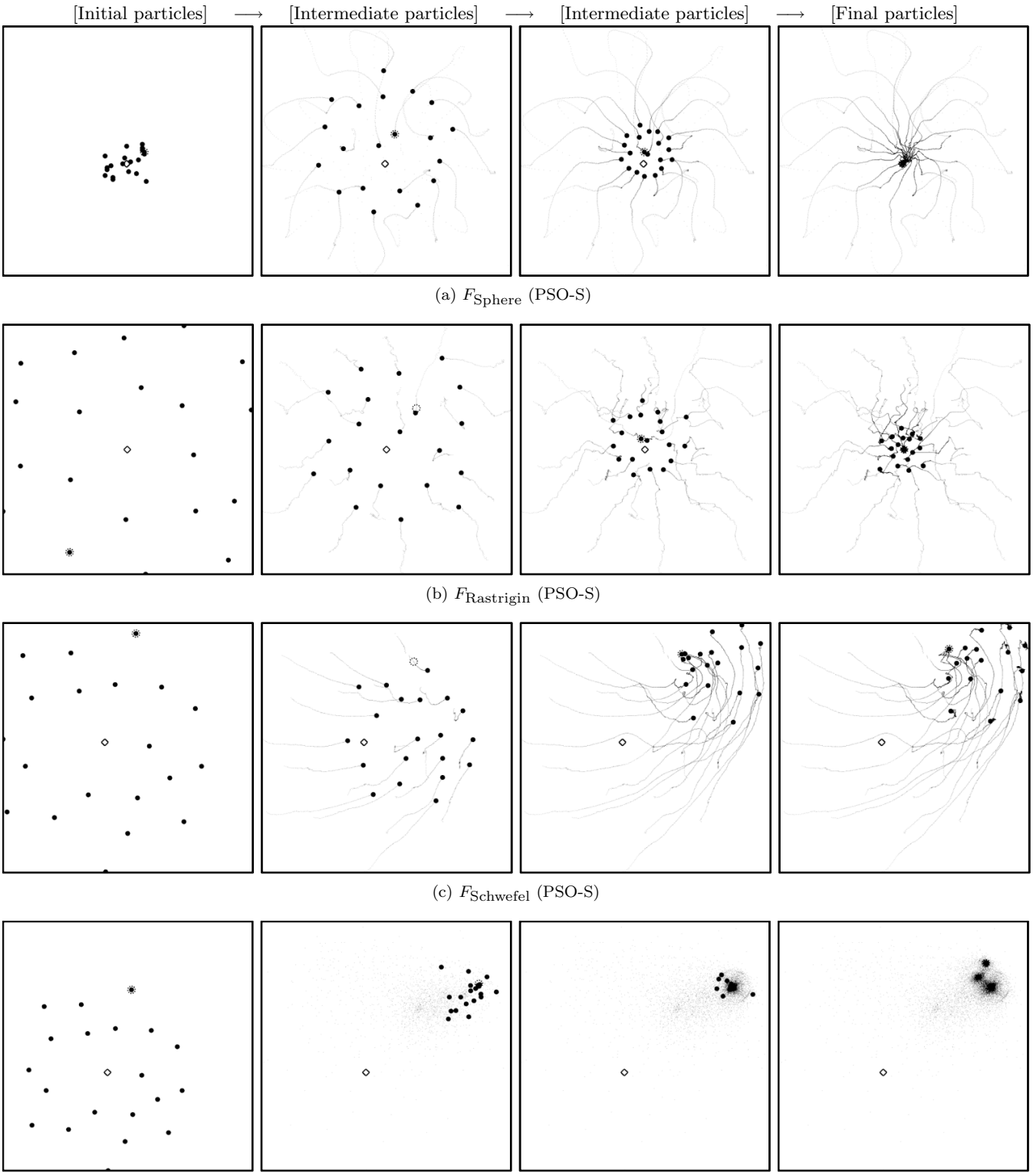
(a) F_{Sphere} (PSO-S) (b) $F_{\text{Rastrigin}}$ (PSO-S)

(c) F_{Schwefel} (PSO-S) (d) F_{Schwefel} (PSO-F)

Figure 5: Traditional visualization: the best and average qualities according to iteration

the global optimum as a dotted circle and an empty diamond '◇', respectively.

Additionally, our visualization tool depicts which regions of search space have been intensively explored by means of color intensities varying across the viewing region. To do so, we partition the two-dimensional region into a grid of small rectangular cells, and then count the number of visits to each cell by any particles incrementally through evolutionary steps. At an arbitrary step, both the global exploration of search space and the individual trajectory of each particle can be represented by displaying every cell that has been visited at least once as a colored rectangle with its intensity



(a) F_{Sphere} (PSO-S)

(b) $F_{\text{Rastrigin}}$ (PSO-S)

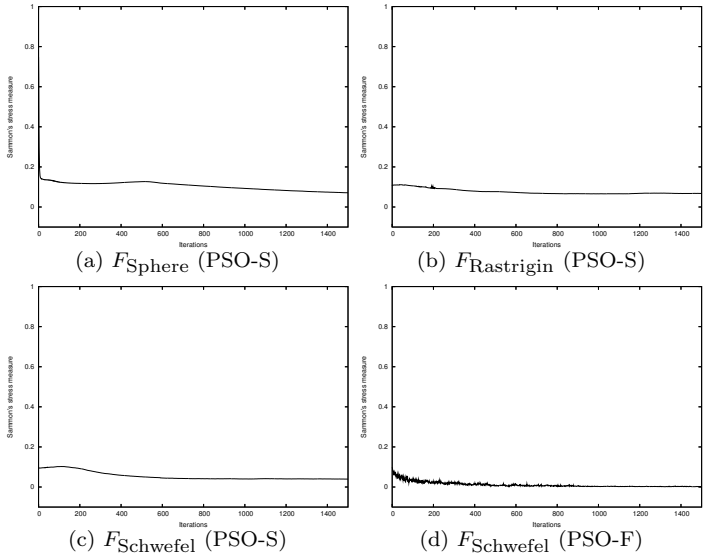
(c) F_{Schwefel} (PSO-S)

(d) F_{Schwefel} (PSO-F)

∗ Filled circle (•): each particle, dotted circle: the global best, and empty diamond (◇): the optimum.

∗ Website for video clips: `http://cg.kw.ac.kr/kang/pso`.

Figure 6: Visualizing the search process of particle swarm optimization



(a) F_{Sphere} (PSO-S) (b) $F_{\text{Rastrigin}}$ (PSO-S)

(c) F_{Schwefel} (PSO-S) (d) F_{Schwefel} (PSO-F)

Figure 7: Sammon's stress measure according to iteration

determined by the number of visits.

Figure 6(a) shows a sequence of images in which twenty particles explore a one-hundred dimensional search space to pursue an optimal value that minimizes F_{Sphere}. Due to the simple, unimodal shape of the function, all of those particles navigate along continuous and smooth trajectories to arrive at the global optimum within 1,500 evolutionary steps.

Figure 6(b) visually demonstrates the increase of difficulty in finding the global optimum of $F_{\text{Rastrigin}}$ even in a lower, twenty-dimensional search space. A significant amount of high-frequency oscillation takes place in the moving trajectories of particles because of a large number of local minima distributed regularly in the search space. After 1,500 iterations, just a few of those particles discover the global minimum, while the remaining particles still oscillate around the neighboring local minima.

Both PSO-S and PSO-F fails in reaching the global optimum of the most challenging test function, F_{Schwefel} in a ten-dimensional space, in which the large distance between the global minimum and the second best local minimum often guides particles along the wrong direction. In Figure 6(c), particles eventually are trapped around a set of irregularly distributed local minima on their misguided ways. The faster moving speed of PSO-F partially addresses this problem by jumping into the nearby regions of lower values, but fails to approach the global optimum. Figure 6(d) exhibits such a process effectively through the successive generation of densely crowded regions.

Each in Figure 7 shows Sammon's stress measure according to evolutionary step. We can see that it is about 0.1 or below in most cases. It means that the mappings were well done with little loss.

For more visually appealing results, you can find a collection of video clips that captures the image sequences for every experiment from the following Website:
`http://cg.kw.ac.kr/kang/pso`.

5. DISCUSSION

In this paper, we depicted the search process of particle swarm optimization by effectively visualizing the motion of a set of particles on invisible high-dimensional space. Not only the proposed visualization pursues minimizing the information loss in the aspect of preserving distances between particles, but it also keeps the continuity of each particle. Through sample experiments, we could see the views that particles are smoothly converging toward the global best, they are moving or jumping by forming a cluster, and so on.

For sample tests, we chose the standard PSO algorithm relying only on the global best and the local best. However, we can also apply our visualization method to other improved PSO algorithms with new features, e.g., the neighborhood best [8], Gaussian mutation [11], chaos [15], self-adaptation [25], and so on, and then watch some new particle-moving patterns appearing from the added features.

We expect that the proposed visualization will give better understanding of how to tune the PSO algorithm by helping detect some troubles occurring during the search.

6. ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their valuable suggestions in improving this paper. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2008-331-C00048).

7. REFERENCES

[1] R. Burke, S. Gustafson, and G. Kendall. A survey and analysis of diversity measures in genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 716–723, 2002.

[2] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1951–1957, 1999.

[3] T. D. Collins. Genotypic-space mapping: Population visualization for genetic algorithms. The Knowledge Media Institute, The Open University, Milton Keynes, UK, Technical Report KMI-TR-39, 30th September 1996.

[4] D. De Ridder and R. Duin. Sammon's mapping using neural networks: a comparison. *Pattern Recognition Letters*, 18(11–13):1307–1316, 1997.

[5] R. Dybowski, T. Collins, and P. Weller. Visualization of binary string convergence by Sammon mapping. In *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 377–383, 1996.

[6] W. Dzwinel. How to make Sammon mapping useful for multidimensional data structures analysis. *Pattern Recognition*, 27(7):949–959, 1994.

[7] R. C. Eberhart and Y. Shi. Comparison between genetic algorithms and particle swarm optimization. In *Proceedings of the 7th International Conference on Evolutionary Programming VII*, pages 611–616, 1998.

[8] R. C. Eberhart and Y. Shi. Particle swarm optimization: Developments, applications and resources. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 81–86, 2001.

[9] R. C. Eberhart, Y. Shi, and J. Kennedy. *Swarm*

Intelligence (The Morgan Kaufmann Series in Artificial Intelligence). Morgan Kaufmann, 2001.

[10] E. Hart and P. Ross. GAVEL - a new tool for genetic algorithm visualization. *IEEE Transactions on Evolutionary Computation*, 5(4):335–348, 2001.

[11] N. Higashi and H. Iba. Particle swarm optimization with Gaussian mutation. In *Proceedings the IEEE Swarm Intelligence Symposium*, pages 72–79, 2003.

[12] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[13] J. Kennedy and W. M. Spears. Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 78–83, 1998.

[14] Y.-H. Kim and B.-R. Moon. New usage of Sammon's mapping for genetic visualization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 1136–1147, 2003.

[15] B. Liua, L. Wanga, Y.-H. Jina, F. Tangb, and D.-X. Huanga. Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals*, 25(5):1261–1271, 2005.

[16] R. W. Morrison and K. A. D. Jong. Measurement of population diversity. In *Proceedings of the 5th European Conference on Artificial Evolution*, pages 31–41, 2001.

[17] E. Pekalska, D. De Ridder, R. Duin, and M. Kraaijveld. A new method of generalizing Sammon mapping with application to algorithm speed-up. In *Proceedings of the Fifth Annual Conference of the Advanced School for Computing and Imaging*, pages 221–228, 1999.

[18] H. Pohlheim. Visualization of evolutionary algorithms - set of standard techniques and multidimensional visualization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 533–540, 1999.

[19] J. W. Sammon, Jr. A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.

[20] B. R. Secrest and G. B. Lamont. Visualizing particle swarm optimization - Gaussian particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 198–204, 2003.

[21] Y. Shi and R. C. Eberhart. A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 69–73, 1998.

[22] Y. Shi and R. C. Eberhart. Parameter selection in particle swarm optimization. In *Proceedings of the 7th International Conference on Evolutionary Programming*, pages 591–600, 1998.

[23] C. Solnon and S. Fenet. A study of ACO capabilities for solving the maximum clique problem. *Journal of Heuristics*, 12(3):155–180, 2006.

[24] S. Tsutsui and D. E. Goldberg. Search space boundary extension method in real-coded genetic algorithms. *Information Sciences*, 133(3-4):229–247, 2001.

[25] Z. Wu and J. Zhou. A self-adaptive particle swarm optimization algorithm with individual coefficients adjustment. In *Proceedings the International Conference on Computational Intelligence and Security*, pages 133–136, 2007.