COMPARISON OF PARTICLE SWARM AND EVOLUTIONARY PROGRAMMING AS THE GLOBAL CONFORMATION OPTIMIZER OF CLUSTERS

MASAO IWAMATSU

Department of Physics, General Education Center, Musashi Institute of Technology
Setagaya-ku, Tokyo 158-8557, Japan
E-mail: iwamatsu@ph.ns.musashi-tech.ac.jp

Received (received date)
Revised (revised date)

The particle swarm optimization (PSO) algorithm and two variants of the evolutionary programming (EP) are applied to the several function optimization problems and the conformation optimization of atomic clusters in order to check the performance of these algorithms as a general-purpose optimizer. It was found that the PSO is superior to the EP though the PSO is not equipped with the mechanism of self-adaptation of search strategies of the EP. However, the PSO cannot find out the global minimum for the atomic cluster though it can find out the global minimum for similar multi-modal benchmark functions of the same size. The size of the cluster which can be handled by the PSO and the EP is limited, and is similar to the one amenable to the popular simulated annealing. The result for benchmark functions only serves as an indication of the performance of the algorithm.

Keywords: particle swarm; evolutionary programming; conformation optimization

1. Introduction

It is well known that many technological problems can lead to problems of global optimization. Hence, an improved global optimization method is highly desirable even in physics. In physics and chemistry, many theoretical studies have been devoted to the goal of global conformation optimization of atomic and molecular clusters. The main difficulty associated with global optimization of clusters is the exponential increase in the search space, which results from the increasing size of the clusters. In fact, it has been proved that the determination of the ground state structure of clusters, which interact even under two-body central force, belongs to the class of NP-hard problems for which no known algorithm is guaranteed to find the global minimum in polynomial time. Therefore traditional local optimization methods may not be useful for such problems.

Various metaheuristics for the numerical optimization of multi-modal functions with continuous variables in multi-dimensional space have been used to solve the problem of global conformation optimization of clusters. For these problems, the simulated annealing (SA) has been widely employed. Biologically-inspired
population-based metaheuristics called evolutionary algorithms (EA), such as the genetic algorithm (GA) and the evolutionary programming (EP) have also become popular. There is, however, a cumbersome problem of temperature scheduling in SA.\textsuperscript{9} The generation of trial vector is also very problematical in SA.\textsuperscript{9} Although the GA is more powerful than the SA, it is usually augmented with traditional local optimization methods.\textsuperscript{5,6} the EP\textsuperscript{10,11} is free from these two defects of the SA and the GA, because it has the mechanism of self-adaptation of search strategy and does not use any gradient-based local search. Unfortunately, the EP is successful only for relatively small problems.\textsuperscript{7,8} Although the EP is also a population-based metaheuristics, it relies solely on the mutation of each individuals and information sharing among the population is incomplete compared to the GA.

The particle swarm (PSO)\textsuperscript{12} is also a population-based general-purpose metaheuristics. Like the EP, the PSO is also free from the above-mentioned two defects of the SA and the GA. Although the PSO has no mechanism of self-adaptation of search strategy, it provides the mechanism to share the information among the population.

Since the detailed comparison of the performance of the EP and the PSO in realistic problem is scarce,\textsuperscript{13} we compare in this article the performance of the EP and the PSO by the function optimization of four benchmark functions as well as by studying the lowest energy structure of atomic clusters as a benchmark problem in physics.

2. Algorithm

The particle swarm (PSO) algorithm used here is the one, which is realized by a C-program written by Shi which is accessible on the Internet.\textsuperscript{14} In short, the algorithm is a population-based algorithm with each individual $i$ is characterized by the pair of vectors which specify the "position" $x$ at the discrete time $t$ with function value $f(x)$ in the $n$-dimensional space and associated "velocity" $v$. Every individual adjusts their velocity using their individual experience about their best position so far as well as the information about the best position of the group they belong. Thus the population search for the global minimum as if the swarm of fishes or birds forages for food. The algorithm is as follows for the function optimization (minimization) problem:

\texttt{LOOP}
\begin{verbatim}
for $i=1$ to number of individuals
    If $f(x_i) > f(p_i)$ then do
        For $d=1$ to dimensions
            $p_{i,d} = x_{i,d}$ (personal best)
        Next $d$
    End do
End $i$
\end{verbatim}
For $j=$ indexes of neighbors
If $f(p_j) > f(p_g)$ then $g = j$ (group best)
Next $j$
For $d=1$ to dimensions

\[ v_{id}(t) = w v_{id}(t - 1) + \phi_1 (p_{id} - x_{id}(t - 1)) + \phi_2 (p_{gd} - x_{id}(t - 1)) \]

If $v_{id}(t) > v_{\text{max}}$ then
\[ v_{id}(t) = v_{\text{max}} \]
else if $v_{id}(t) < -v_{\text{max}}$
\[ v_{id}(t) = -v_{\text{max}} \]
end if
\[ x_{id}(t) = x_{id}(t - 1) + v_{id}(t) \]
Next $d$
Next $i$
Until criterion

where $\phi_1$ and $\phi_2$ are the (real) random number between $[0, 2]$, and $w$ is called inertia weight. In our code by Shi, it starts from $w_{\text{initial}}$ and decrease linearly until $w_{\text{final}}$ at the last step of the iteration.

\[ w_{\text{initial}} \rightarrow w_{\text{final}} \quad (1) \]

which seems to mimic the temperature scheduling of SA. We choose $w_{\text{initial}} = 0.9$ and $w_{\text{final}} = 0.4$ which is suggested in the original code by Shi. Our neighbors are the whole population, therefore the indexes of neighbors $j$ runs from 1 to the number of individuals. We choose the maximum velocity $v_{\text{max}} = 0.7$. This value is chosen from preliminary experiments but is not optimized.

In short, each particle in the PSO changes its velocity and position according to the following "equation of motion"

\[ v_{id}(t) = w v_{id}(t - 1) + \phi_1 (p_{id} - x_{id}(t - 1)) + \phi_2 (p_{gd} - x_{id}(t - 1)) \quad (2) \]
\[ x_{id}(t) = x_{id}(t - 1) + v_{id}(t) \quad (3) \]

to seek for the global minimum.

In the evolutionary programming (EP), on the other hand, the individual $i$ at the time $t - 1$ is characterized by the pair of vectors which specify the "position" $\mathbf{x}$ and the typical footstep size $\eta$ of the random walk in the multi-dimensional space. The latter parameter is called the "strategy parameter". Each individual produces new position $\mathbf{x}'$ and new foot step $\eta'$ from the present position by a random walk using the given footstep. Thus, each individual produces single offspring. The new generation at $t$ is selected from the union of the present individual and his/her offspring by tournament selection. The EP algorithm is schematically written as follows:

\[ \mathbf{v}_{id}(t) = w \mathbf{v}_{id}(t - 1) + \phi_1 (p_{id} - \mathbf{x}_{id}(t - 1)) + \phi_2 (p_{gd} - \mathbf{x}_{id}(t - 1)) \]
\[ \mathbf{x}_{id}(t) = \mathbf{x}_{id}(t - 1) + \mathbf{v}_{id}(t) \]
LOOP
FOR i=1 to number of individuals
FOR d=1 to dimensions
\[ x'_{id}(t-1) = x_{id}(t-1) + \eta_{id}(t-1)N_d(0, 1) \] (produce offspring)
\[ \eta'_{id}(t-1) = \eta_{id}(t-1) \exp \left( \tau' N(0, 1) + \tau N_d(0, 1) \right) \]
If \( \eta'_{id}(t-1) < \epsilon \) then do
\[ \eta_{id}(t-1) = \epsilon \]
End do
Next d
Next i
Choose next generation \((x_i(t), \eta_i(t))\) from union of \((x_i(t-1), \eta_i(t-1)) \cup (x'_i(t-1), \eta'_i(t-1))\) by tournament selection
Until criterion

where \( N(0, 1) \) describes the mutation of the individual and is given by the Gaussian random number with average 0 and standard deviation 1. \( N_d(0, 1) \) indicates that the new random number is drawn anew for each component \( d \). \( \tau \) and \( \tau' \) is usually chosen as \( \tau = 1/\sqrt{2\sqrt{n}} \) and \( \tau' = \sqrt{2n} \), where \( n \) is the dimension. \( \epsilon \) (= \( 10^{-4} \)) is the threshold to prevent the premature convergence. \(^{15}\)

In contrast to the PSO, the EP does not realize direct mechanism to share the information among individuals. The information spreads through the population only by the tournament selection. However, the EP realizes the mechanism of self-adaptation of search space by tournament selection.

In order to improve the performance of this classical evolutionary programming (CEP) using Gaussian mutation described by the code

\[ x'_{ij} = x_{ij} + \eta_{ij} N_d(0, 1) \] (4)

where \( N_d(0, 1) \) is the Gaussian random number, Yao et al. \(^{11}\) proposed the fast evolutionary programming (FEP), which is described by the mutation

\[ x'_{ij} = x_{ij} + \eta_{ij} C_d(0, 1) \] (5)

where \( C_d(0, 1) \) is the Cauchy random number. They \(^{11}\) showed that the performance of the EP is improved for difficult multi-modal functions by introducing the FEP. Further generalization of the EP by using more general Lévy-type mutation is also proposed. \(^{16}\)

In the next section, we will compare the performance of two variant of the EP (CEP, FEP) and the PSO for four benchmark functions as well as for the conformation optimization problem of Lennard-Jones cluster.

3. Numerical results and discussions
3.1. Function optimization of benchmark functions

We have tested the PSO and the EP algorithm for three types of benchmark functions\textsuperscript{11,12}

1) Uni-modal function

2) Multi-modal function with numerous minima

3) Multi-modal function with very few minima

with two variants of EP and PSO

a) Classical evolutionary programming (CEP) with Gaussian mutation\textsuperscript{10}

b) Fast evolutionary programming (FEP) with Cauchy mutation\textsuperscript{11}

c) Particle swarm optimization (PSO) realized by Shi.\textsuperscript{14}

The number of individuals in the population is 100 for the EP and the PSO. The initial values for $x_i$ of individual $i$ are uniformly distributed about the entire search space where the function to be optimized is defined for both the EP and the PSO. The initial velocities $v_i$ of the PSO are also uniformly distributed over the interval $-v_{\text{max}} \leq v_{id} \leq v_{\text{max}}$, where we used $v_{\text{max}} = 0.7$. The initial step lengths $\eta_{id}$ of the EP are all set to $\eta_{id} = 1.0$. Numerical experiments are conducted 50 times and the average of 50 bests and the best among 50 bests are recorded.

![Graph showing the evolution of the average and best fitness (function value) for the sphere function. The PSO shows the best performance.](image)
Fig. 1 shows the evolution of the fitness (function value) of sphere function:

\[ f_1(x) = \sum_{i=1}^{n} x_i^2, \quad (-100 \leq x_i \leq 100) \] (6)

which is the uni-modal function with the global minimum \( f_1(0, 0, \ldots, 0) = 0 \). We choose the dimension \( n=33 \) in order to compare the results with cluster composed of 13 atoms later. We should note in passing that only 33 variables are independent in 13-atom cluster because 6 degree of freedom should be fixed by the translation and rotation degrees of freedom.

Since it is customary to compare the performance between various evolutionary algorithms not by the CPU time but by the generation or the number of function evaluations to avoid the hardware-dependence and programmer-dependence of the performance index, we have shown in Fig. 1 the function value as the function of iteration steps, which is equivalent to the generation of the EP and the time step of the PSO, and is directly proportional to the number of function evaluations in both the EP and the PSO.

Experiments are conducted 50 times, 5000 steps for each. The results are averaged and the best trial is extracted. Again, we have taken the custom of EC community to fix the computation resources to 5000 steps since we intend to study the average performance of each algorithm and do not want to pursue the true global minimum. The results in Fig. 1 clearly shows that the average function value of 50 samples satisfies PSO<CEP<FEP at the later stage, while the best of 50 samples shows PSO<CEP<FEP. Therefore the PSO algorithm gives the lowest function value and shows best performance on average as well as in terms of the ability of locating global minimum among three.

Figure 2 shows the results for Griewank function:

\[ f_2(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1, \quad (-600 \leq x_i \leq 600) \] (7)

which is the multi-modal function with many local minima. Experiments are conducted 50 times, 5000 steps for each method. The results clearly shows that the average of 50 samples satisfies PSO<CEP<FEP at the later stage, while the best of 50 samples shows PSO<CEP<FEP. Therefore the PSO algorithm again shows best performance among three.

Figures 3 and 4 show the results for Ackley function:

\[
\begin{align*}
\text{f}_3(x) &= -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + c, \\
&\quad (-32 \leq x_i \leq 32)
\end{align*}
\] (8)
Fig. 2. The evolution of the fitness for the Griewank function. The PSO shows the best performance.

Fig. 3. The evolution of the fitness for the Ackley function. The PSO shows the best performance.
which is the multimodal function with numerous minimum, and Shekel function:

\[ f_4(x) = -\sum_{i=1}^{5} \left[ (x - a_i)(x - a_i)^T + c_i \right], \quad (0 \leq x_i \leq 10) \] (9)

which has only a few minimum, where \( x, a_i \) and \( c \) are 4 dimensional vector. The values for \( a \) and \( c \) are given, for example, in the reference. In these cases, again, the PSO shows the best performance.

Table I shows the best results and average obtained at the last step of the run. We also list the CPU time necessary to conduct 50 trials. The program was compiled by gcc 2.96 compiler with O2 option under the Red-Hat Linux 7.3. It was executed using a PC equipped with the Pentium IV 2 GHz. The results in Table I clearly indicate the superiority of the PSO over the EP in terms of not only the quality of the solution but by the CPU time consumed except Shekel function. The difference of CPU time between the FEP and the CEP comes mainly from the overhead of generating the Gaussian and the Cauchy random numbers. However, other factors such as the efficiency of tournament selection which depends on the uniformity of the population may affect the performance difference between the FEP and the CEP.

3.2. Conformation optimization of atomic clusters

The problem we consider in this subsection is the global optimization of the conformation of atomic clusters, for which not only the popular metaheuristics such
Table 1: The best, average and standard deviation for the benchmark functions at the last step of the run. The CPU times necessary to conduct 50 trials by the Pentium IV 2GHz operating under Red Hat Linux 7.3 are also shown.

<table>
<thead>
<tr>
<th>Function (Global min.)</th>
<th>Steps</th>
<th>Algorithms</th>
<th>Best value</th>
<th>Average</th>
<th>Standard Dev.</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere (0.0)</td>
<td>1000</td>
<td>CEP</td>
<td>3.48035</td>
<td>85.4363</td>
<td>123.866</td>
<td>3m42.250s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FEP</td>
<td>0.0264094</td>
<td>1.98270</td>
<td>5.48032</td>
<td>4m16.770s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>0.0000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0m25.670s</td>
</tr>
<tr>
<td>Griewank (0.0)</td>
<td>5000</td>
<td>CEP</td>
<td>5.82395e-9</td>
<td>0.781703</td>
<td>1.65689</td>
<td>20m41.500s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FEP</td>
<td>8.27435e-8</td>
<td>0.0452723</td>
<td>0.0478312</td>
<td>23m24.360s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>0.0000000</td>
<td>0.0226850</td>
<td>0.0254040</td>
<td>4m41.290s</td>
</tr>
<tr>
<td>Ackley (0.0)</td>
<td>5000</td>
<td>CEP</td>
<td>4.54117</td>
<td>9.93885</td>
<td>2.87620</td>
<td>19m34.680s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FEP</td>
<td>9.23827e-4</td>
<td>1.04946e-3</td>
<td>6.24384e-5</td>
<td>25m05.380s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>0.0000000</td>
<td>0.0000000</td>
<td>0.000000</td>
<td>4m41.480s</td>
</tr>
<tr>
<td>Shekel (~-10)</td>
<td>5000</td>
<td>CEP</td>
<td>-0.0903624</td>
<td>-0.0385105</td>
<td>0.0128108</td>
<td>9m1.930s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FEP</td>
<td>-0.216453</td>
<td>-0.125382</td>
<td>0.0514568</td>
<td>6m59.420s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>-9.99225</td>
<td>-8.26486</td>
<td>2.34829</td>
<td>2m09.320s</td>
</tr>
</tbody>
</table>

as SA, and Tabu-search but the population-based heuristics such as the GA or the EP have been adopted.

The system consists of N atoms, and the optimum arrangement (conformation) of atoms corresponds to the global minimum of the total potential energy \( f_{\text{LJ}} \) of the N-atom cluster defined by

\[
f_{\text{LJ}}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} u(r_{ij})
\]

where \( u \) is the pair potential that represents the inter-atomic interaction between two atoms \( i \) and \( j \) in the cluster. In this work we used the Lennard-Jones potential

\[
u(r_{ij}) = \begin{cases} 
\frac{1}{r_{ij}^{12}} - \frac{1}{r_{ij}^{6}} & \text{if } r_{ij} \leq r_{ij}^{c} \\
0 & \text{otherwise}
\end{cases}
\]

where

\[
r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}
\]

is the distance between two atoms \( i \) at \((x_i, y_i, z_i)\) and \( j \) at \((x_j, y_j, z_j)\). Therefore, the conformation optimization problem is nothing more than just the global optimization problem of multi-modal function \( f_{\text{LJ}}(\mathbf{x}) \) (total potential energy). We choose this conformation optimization problem of the Lennard-Jones clusters as the target problem in addition to the standard benchmark functions of the previous subsection because it seems the simplest yet still realistic problem. Furthermore, the most likely candidates of global minima are known from brute-force calculation.

In order to test the three algorithms the CEP, the FEP and the PSO to the problem of conformation optimization of atomic clusters, initially, we have used the
same algorithm parameter as used in benchmark functions. The problem size is rather small as the conformation optimization problem because we are interested mostly in the performance of the EP and the PSO as a general-purpose optimizer. We are not aiming at proving the superiority of these algorithms over the special purpose program. \(^{18,19}\)

![Fig. 5. The evolution of the total potential energy for the 7-atom Lennard-Jones cluster. The PSO show the best performance.](image)

Figure 5 shows the results for 7 atom Lennard-Jones cluster \(\text{LJ}_7\) where the number of variables is \(n = 3 \times 7 - 6 = 15\). Among three algorithms tested, the PSO shows the best performance on average and it can find out the global minimum. Actually the final result of the best conformation energy agrees within 5 digits with the best value \(f_{\text{LJ}} = -16.505384\) reported so far. The best result among the best 50 results obtained at the last step of each run are summarized in table II.

Similarly, the results for 11-atom cluster and 13-atom cluster are shown in Figure 6 and Figure 7. The results for the 7-, 11- and 13-atom clusters are summarized in table II. Among the three optimization algorithms examined, the PSO shows the best performance again. The PSO can successfully find out the most-probable global minimum for 7- and 11-atom clusters. Unfortunately, it cannot find out the global minimum for 13-atom cluster.

In the PSO, each particle follows the equation motion (2) and (3). Therefore, the best particle with index \(i = g\) will lose the velocity steadily because \(p_{id} = x_{id}\), and \(p_{gd} = x_{id}\) unless new best particle appear and will takes over the old one. Therefore, the best particle would stagnate at the local minimum, and the whole population might be attracted toward this local minimum. Similarly, the population in the EP
Table 2: The best, average and standard deviation results for the Lennard-Jones clusters at the last step of the run. The CPU times necessary to conduct 50 trials by the Pentium IV 2GHz operating under Red Hat Linux 7.3 are also shown.

<table>
<thead>
<tr>
<th>Cluster (Global min.)</th>
<th>Steps</th>
<th>Algorithms</th>
<th>Best value</th>
<th>Average value</th>
<th>Standard Dev.</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LJ_7 (-16.505384)</td>
<td>5000</td>
<td>CEP</td>
<td>-15.9350</td>
<td>-15.1070</td>
<td>0.521976</td>
<td>22m06.360s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FEP</td>
<td>-15.9350</td>
<td>-15.3938</td>
<td>0.519583</td>
<td>24m39.090s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>-16.5054</td>
<td>-15.6011</td>
<td>0.689472</td>
<td>11m52.490s</td>
</tr>
<tr>
<td>LJ_11 (-32.765970)</td>
<td>5000</td>
<td>CEP</td>
<td>-32.3652</td>
<td>-30.1710</td>
<td>0.907479</td>
<td>38m52.680s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FEP</td>
<td>-32.6750</td>
<td>-26.7931</td>
<td>6.11191</td>
<td>52m37.750s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>-32.7659</td>
<td>-30.6488</td>
<td>1.27071</td>
<td>29m24.780s</td>
</tr>
<tr>
<td>LJ_13 (-44.326801)</td>
<td>5000</td>
<td>CEP</td>
<td>-40.8842</td>
<td>-37.6523</td>
<td>2.78377</td>
<td>47m09.460s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FEP</td>
<td>-40.3179</td>
<td>-23.9940</td>
<td>9.19536</td>
<td>74m06.210s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>-41.3935</td>
<td>-39.0490</td>
<td>1.31955</td>
<td>40m50.400s</td>
</tr>
</tbody>
</table>

Fig. 6. The evolution of the total potential energy for the 11-atom Lennard-Jones cluster. The PSO show the best performance.
may lose the diversity due to the tournament selection. Then the majority of the population would degenerate into the local minimum.

Seeing that the problem size for 13-atom cluster and Griewank benchmark function examined in the previous section is the same, the PSO may not be efficient for the realistic problem. Another conclusion drawn from this simple exercise is that the comparison of performance using the benchmark function only serves as a rough indication of the ability of the algorithm since no single algorithm is know to be best for all optimization problem from the “No free lunch theorem”.  

The algorithm may not be so efficient to realistic problem even if it is effective to benchmark problems of the same size.

The reason is apparent. Figure 9 compares the energy landscape of the potential energy of $\text{LJ}_{13}$ and the Griewank function in a two-dimensional plane where other variables are fixed to the values at the global minimum. Because of the divergence of the interatomic potential (11) at the origin, the search space is partitioned by singular points which satisfies, $x_i = x_j$, $y_i = y_j$ and $z_i = z_j$, and $x_i = y_i = z_i = 0$ because one atom is fixed at the origin $(0,0,0)$ to eliminate the 3 degree of freedom of the cluster. On the other hand the energy landscape of Griewank function is very rugged but is still quite regular. Therefore, even though the size of the problem is the same, the problem of finding global minimum is more difficult for $\text{LJ}_{13}$ than other benchmark functions such as Griewank functions. Most of the benchmark functions that are popular among computer scientists do not possess the singularity. Therefore even if the given algorithm is efficient for benchmark functions,
Particle Swarm and Evolutionary Programming

Fig. 8. The evolution of the total potential energy for the 13-atom Lennard-Jones cluster. The PSO with final inertia weight $w_{\text{final}} = 0.1$ was used. Now, the global minimum was found before 3500 steps. The initial inertial weights $w_{\text{initial}}$ used are given in the figure. The constant inertia weight $w = 0.1$ seems to show the best performance.

it may not be so for the cluster optimization problem.

According to Shi, the inertia weight $w$ in the PSO algorithm was decreased linearly from $w_{\text{init}}$ to $w_{\text{final}}$ as in eq. (1). In order to check the effect of inertia weight on the performance of the PSO, we change the final weight $w_{\text{final}}$ from 0.4 to 0.1, which hopefully realizes fine search. Figure 8 shows the evolution of the best value of 50 samples when $w_{\text{initial}} = 0.9, 0.7, 0.5, 0.3, 0.1$. As can be seen clearly, the PSO can successfully find out the global minimum $f_{\text{LJ}} = -44.326801$ of 13-atom cluster now. Furthermore the algorithm with constant inertia weight $w_{\text{initial}} = 0.1$ seems to show the fastest convergence, which seems to throw a doubt on the effectiveness of simulated-annealing-like weight scheduling (1). Although our weight scheduling is not optimum, weights smaller than 0.1 could not give the global minimum. The issue of the optimal weight scheduling is beyond the scope of our study. But it is apparent from this exercise that the PSO shares the same deficit of cumbersome temperature scheduling with as the classical SA.

One of the attractive feature of so-called intelligent optimization methods such as the EP and the PSO is the self-adaptation of search strategies, while classical optimization methods such as the simulated annealing (SA) need cumbersome hand-code scheduling of search strategies by trial and error. Unfortunately, our exercise for the Lennard-Jones clusters shows that the self-adaptation of the EP and the PSO is not effective enough to handle the realistic cluster optimization problem. Furthermore, even the PSO shares the same problem of hand-code scheduling with
Fig. 9. The potential energy landscapes (a) of the 13-atom Lennard-Jones cluster and (b) of the Griewank function. The darker the lower is the energy. Global minimum is at (0.483770,0.483602) for LJ\textsubscript{13} and at (0,0) for the Griewank function. The variables except two variables corresponding to the vertical and the horizontal axes are all fixed to the values at the global minimum. The potential energy of the Lennard-Jones potential diverges at \(x = y = 0\) because one atom is fixed at the origin (0,0,0) in order to eliminate the 3 degree of freedom, and because the interatomic potential diverges at the origin. (11). Two lines along \(x = 0\) and \(y = 0\) are high ridges of potential walls.

the SA.

In order to design a good global optimizer, an efficient combination of global search (exploration) and local search (exploitation) is necessary. It is well recognized that the global optimizers such as the SA\textsuperscript{4} and the GA\textsuperscript{5} have to be combined with a gradient-based local optimization method such as the conjugate-gradient method to optimize the structure of the Lennard-Jones cluster of realistic size. This means that the ability of local search is relatively weak compared to that of global search in SA and GA. Since the performance of our EP and PSO seems almost comparable to the previous data of SA in the literature,\textsuperscript{4} both the EP and the PSO should be further reinforced by including a similar gradient-based local search method. From such a point of view, the Monte-Carlo plus local minimization called Basin-Hopping,\textsuperscript{18} which implements both the global search by Monte-Carlo move and the local search by the gradient-based method seems still the powerful strategy to tackle the problem of conformation optimization.

4. Conclusion

In this article, we have tested the two general-purpose optimizers, the particle swarm optimization (PSO) and the evolutionary programming algorithms (EP), for the standard benchmark function optimization as well as the global conformation
optimization of atomic clusters. We have chosen the problem of conformation optimization of atomic clusters, because it is realistic yet very standard benchmark problem in physical chemistry. The numerical experiments clearly indicate the superiority of the PSO over the EP. However, the results are not satisfactory, because not only the EP but also the PSO cannot locate the global minimum of very small cluster composed of only 13 atoms without tuning the algorithm parameters. Only after tuning the inertia weight parameter, we can successfully locate the global minimum by the PSO.

The conclusions which can be drawn from our exercise are:

1. A test by the standard benchmark functions can only serve as a rough measure of the performance of the algorithm, because those benchmark functions are still not very singular.

2. The so-called intelligent strategies by self-adaptation of search strategies in the EP and by the information sharing among individuals in the PSO are not very effective for the realistic cluster optimization problem.

3. The PSO and the EP show comparable performance to the SA, which indicates that both the PSO and the EP need the ability of fine local search as the SA does.

4. The PSO shares the same deficit of cumbersome scheduling of search strategy with the SA.

Our simple exercise suggests that the EP and the PSO should be hybridized with standard gradient-based local optimizer to furnish the ability of the fine local search to make them a truly general-purpose global optimizer.

Acknowledgements
The author is grateful to the Department of Physics, Tokyo Metropolitan University for providing him a visiting fellowship during the course of this work. He is particularly indebted to Professor Y. Okabe for his continuous interest in his work.

References