ISSN 1 746-7233, England, UK
World Journal of Modelling and Simulation
Vol. 7 (2011) No. 4, pp. 312-320

Minimizing Lennard-Jones potential using a real coded Genetic Algorithm and Particle Swarm Optimization*

Kusum Deep†, V. K Katiyar, Shashi

Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee 247667, India

(*Received March 1 2010, Accepted September 5 2011*)

Abstract. In this paper a recently developed real coded genetic algorithm called MI-LXPM of Deep et al. [8] and Particle Swarm Optimization (PSO) are applied to the challenging problem of finding the minimum energy configuration of a cluster of identical atoms interacting through the Lennard-Jones potential. Finding the global minimum of this function is very difficult because it has a large number of local minima, which grows exponentially with molecule size. Computational results for a cluster containing up to 15 atoms are obtained and presented. The obtained results show the remarkable performance of MI-LXPM as compared to PSO and to the earlier published results.

Keywords: Lennard-Jones potential, atomic cluster, Genetic Algorithm, Particle Swarm Optimization, global optimization

1 Introduction

Finding the most stable configuration of atomic clusters is a challenging problem as it is highly complex and its complexity increases with the increase in number of atoms. Minimum energy usually corresponds to maximum stability for atomic clusters[11]. Also it can be formulated as a global optimization problem. It is an eminently challenging global optimization problem as the number of local minima increases exponentially with the size of the molecule[16]. One of the first, most efficient and successful approaches to Lennard-Jones (L-J) cluster optimization was introduced in [22] and further refined in [24]. This approach is based on the idea of starting local optimization from initial configurations built by randomly placing atoms in predefined points in space, according to lattice structures which is most commonly found in nature.

Successful methods for solving the L-J problem are the basin hopping method[23], the big-bang method[19], Leary's [20] descent method, genetic algorithms[7, 14] and differential evolution[21]. A geometric crossover technique, known as cut and splice, has been proposed in [5, 6]. PSO has also been applied to L-J problem[17].

There are also hybrid algorithms which combine Monte Carlo (MC) or simulated annealing (SA) type optimization with genetic algorithm[15]. In [2], different optimization methods are considered, multi level single link, topographical differential evolution and a genetic algorithm but it is shown that genetic algorithms combined with an efficient local optimization method is especially speedy and reliable for the task of finding the structure of atomic and molecular clusters. A recently developed technique is hybridization of GA with Monte Carlo approach[12] for geometry optimization of atomic clusters.

It is observed that in most of the results in literature[2, 21] etc. that the computational effort is substantial in terms of function evaluations, where [21] used an iterative local search based differential evolution algorithm and [2] a binary coded genetic algorithm for solving L-J potential energy function. With a view to use an

* The authors gratefully acknowledge financial assistance from Ministry of Human Resource Development, India.

† Corresponding author. Tel.: +91-1332-285339. *E-mail address*: kusumfma@iitr.ernet.in, kusumdeep@gmail.com.

optimization algorithm which employs less number of function evaluations, in this paper the above problem is solved by a recently developed real coded genetic algorithm called MI-LXPM[8] and PSO[18]. The results show that MI-LXPM uses nearly one third of the function evaluations as reported in literature[2, 21]. However PSO's performance is not appreciable for this problem.

The paper is organized as follows. Section 2 describes the Lennard-Jones potential energy function mathematically. Section 3 describes the real coded genetic algorithm MI-LXPM applied to the problem described in section 2. Basic PSO is described in Section 4. Computational results are presented in Section 5 and discussion of results in Section 6. Conclusions are presented in Section 7.

2 Problem discussion

The L-J problem assumes that the potential energy of a cluster of atoms is given by the sum of pairwise interactions between atoms, with these interactions being Vander Waals forces given by the Lennard-Jones 6-12 potential. The problem consists of determining the positions $t_1, t_2, \cdots, t_n \in R^3$ of an n atom cluster in such a way that the L-J potential given by

$$V = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(e_{ij}^{-12} - 2e_{ij}^{-6} \right), \tag{1}$$

where e_{ij} is the Euclidean distance between the points t_i and t_j, generated by atomic interactions is minimized. Each atom of the cluster is characterised by a set of three real numbers corresponding to the (x, y, z) position of each atom. The problem is then to find the positions of each atom of the cluster that corresponds to the global minimum value of V, given in Eq. (1).

3 Real coded genetic algorithm MI-LXPM

Genetic algorithms are population based heuristics which are used to determine solution of non-linear optimization problems. GA's mimic the Darwin's principal of "survival of fittest". GA uses three basic operations selection, crossover and mutation in moving from one generation to another.

MI-LXPM[8] is a real coded genetic algorithm which uses the Laplace crossover[10] and Power mutation[9]. It also uses tournament selection. It is efficient to solve integer, real and mixed integer, non-linear constrained optimization problems. More details of these operators are defined in subsections.

3.1 Laplace crossover

Two offsprings, $y^1 = (y_1^1, y_2^1, \cdots, y_n^1)$ and $y^2 = (y_1^2, y_2^2, \cdots, y_n^2)$ are generated from two parents, $x^1 = (x_1^1, x_2^1, \cdots, x_n^1)$ and $x^2 = (x_1^2, x_2^2, \cdots, x_n^2)$ in the following way. First, uniform random numbers are $u_i, r_i \in [0, 1]$ generated. Then a random number β_i is generated satisfying the Laplace distribution, as under,

$$\beta_i = \begin{cases} a - blog(u_i), & r_i \leq 1/2, \\ a + blog(u_i), & r_i > 1/2, \end{cases} \tag{2}$$

where a is location parameter and $b > 0$ is scaling parameter. With smaller values of b, offsprings are likely to be produced nearer to parents and for larger values of b, offsprings are expected to be produced far from parents. Having computed β_i, the two offsprings are obtained as under:

$$y_i^1 = x_i^1 + \beta_i \left| x_i^1 - x_i^2 \right|, \; y_i^2 = x_i^2 + \beta_i \left| x_i^1 - x_i^2 \right|. \tag{3}$$

Also one important thing to notice is that Eq. (3) gives:

$$|y_i^1 - y_i^2| = \left| x_i^1 - x_i^2 \right|, \tag{4}$$

which shows that in laplace crossover the spread of offsprings is proportional to the spread of parents.

3.2 Power mutation

A solution x is created in the vicinity of a parent solution $\overline{x}$ in the following manner. First, a uniform random number t between 0 and 1 and a random number s which follows the power distribution, $s = (s_1)^p$, where s_1 is a uniform random number between 0 and 1, are created. p is called the index of mutation. It governs the strength of perturbation of power mutation. Having determined s a muted solution is created as:

$$x = \begin{cases} \overline{x} - s(\overline{x} - x^l), & t < r, \\ \overline{x} + s(x^u - \overline{x}), & t \geq r, \end{cases} \tag{5}$$

where

$$t = \frac{\overline{x} - x^l}{x^u - \overline{x}},$$

x^l and x^u being the lower and upper bounds on the value of the decision variable and r a uniformly distributed random number between 0 and 1.

3.3 Selection technique

A selection technique in a GA is simply a process that favors the selection of better individuals in the population for the mating pool. MI-LXPM uses tournament selection.

3.4 Computational steps of MI-LXPM

Computational steps of the MI-LXPM algorithm are as follows:

Step 1. Generate a suitably large initial set of random points within the domain prescribed only by the bounds on variable i.e. points satisfying $x_j^L \leq x_j \leq x_j^U, j = 1, 2, \cdots, n$, for variables which are to have real values and $y_j^L \leq y_j \leq y_j^U$, y_j integer for variables which are to have integer values.
Step 2. Check the stopping criteria, if satisfied stop; else goto *Step 3*.
Step 3. Apply tournament selection procedure on initial (old) population to make mating pool.
Step 4. Apply laplace crossover and power mutation to all individuals in mating pool, with probability of crossover p_c and probability of mutation p_m respectively, to make new population.
Step 5. Apply integer restrictions on variables where necessary and evaluate their fitness values.
Step 6. Increase generation; replace old population by new population; goto *Step 2*.

4 Particle swarm optimization (PSO)

The idea behind PSO is based on the simulation of the social behavior of bird flock and fish schools. PSO is a swarm intelligence method for global optimization problems. It differs from well-known evolutionary algorithms as in evolutionary algorithms a population of potential solutions is used to probe the search space, but no operators, inspired by evolution procedures, are applied on the population to generate new promising solutions. Instead in PSO, each individual, namely particle, of the population, called swarm, adjusts its trajectory towards its own previous best position (pbest), and towards the previous best position of any member of its topological neighborhood (gbest). Two variants of the PSO have been developed, one with a global neighborhood and the other with a local neighborhood. According to the global variant, each particle moves towards its best previous position and towards the best particle in the whole swarm. On the other hand, in the local variant, each particle moves towards its best previous position and towards the best particle in its restricted neighborhood.

Working of PSO may be briefly described as under:
Suppose the search space is D dimensional, then the i_{th} particle of the swarm can be represented by a D-dimensional vector, $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})^T$. The velocity (position change) of this particle can be represented by another D- dimensional vector $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})^T$. The best previously visited position of the i_{th} particle is denoted as $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})^T$. Defining g as the index of the best particle in the swarm,

the swarm is manipulated according to the following two equations:
velocity update equation:

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}), \tag{6}$$

position update equation:

$$x_{id} = x_{id} + v_{id}, \tag{7}$$

where $d = 1, 2, \cdots, D; i = 1, 2, \cdots, S$, where S is the size of the swarm; c_1 and c_2 are constants, called cognitive and social scaling parameters respectively (usually, $c_1 = c_2$; r_1, r_2 are random numbers, uniformly distributed in $[0, 1]$). Eq. (6) and (7) define the initial version of PSO algorithm. A constant, $V_{\max}$, was used to arbitrarily limit the velocities of the particles and improve the resolution of the search. The pseudo code of PSO is shown below:

```
For $t = 1$ to the max. bound of the number on iterations,
  For $i = 1$ to the swarm size,
    For $d = 1$ to the problem dimensionality,
      Apply the velocity update equation:
      Update Position
    End- for-d;
    Compute fitness of updated position;
    If needed, update historical information for $P_i$ and $P_g$;
  End-for-i;
  Terminate if $P_g$ meets problem requirements;
End-for-t;
End algorithm.
```

The maximum velocity $V_{\max}$, serve as a constraint to control the global exploration ability of particle swarm. A larger $V_{\max}$ facilitates global exploration, while a smaller $V_{\max}$ encourages local exploitation. Clerc [4] has introduced a constriction factor, χ which improves PSO's ability to constrain and control velocities. χ is computed as:

$$\chi = \frac{2}{|2 - \phi - \sqrt{\phi(\phi - 4)}|}, \tag{8}$$

where $\phi = c_1 + c_2$, $\phi > 4$ and the velocity update equation is then

$$v_{id} = \chi^{*}(v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})). \tag{9}$$

Eberhart and Shi [13] found that, χ combined with constraints on $V_{\max}$, significantly improved the PSO performance.

5 Computational results

In this section we present numerical results using MI-LXPM and PSO algorithm to the energy function V. Both the programs are executed on a Pentium IV with 512 MB of RAM.

The problem is to find the most stable conformation of the cluster with atoms ranging from 3 to 15 having known global minimum energy values[1]. Cartesian coordinates of each atom are considered as decision variables.

5.1 Parameter setting for MI-LXPM

100 independent runs are performed, each time using a different seed for the generation of random numbers. The algorithm stops whenever either the maximum number of generations have been attained or

the known global minimum has been reached. A run is considered successful if the global minimum function value is reached.

Population size for each of the case is kept fixed and is equal to ten times the number of variables however maximum number of generations allowed is varied in each of the cases as the complexity of the problem increases with the increase in the number of atoms. The probability of crossover is fine-tuned keeping the probability of mutation fixed at 0.001. The mutation probability is fixed as there is negligible effect on the success rate of its variation from 0.001 to 0.005 which is fine tuned earlier[10], whereas there is significant variation in the success rate with the change of crossover probability. Tab. 1 shows the variation of success rate of MI-LXPM as we change the probability of crossover (p_c) between 0.5 to 1.0 keeping probability of mutation fixed ($p_m = 0.001$). Also Fig. 1 shows the effect of crossover probability on success rate and it is clearly seen that p_c =0.9 corresponds to the highest success rate (100%), so p_c is taken as 0.9. Fig. 2 represents generation wise variation of the objective function value corresponding to two different crossover probabilities i.e. p_c=0.8 and p_c=0.9 and it is quiet clear that choice of appropriate crossover probability is the major contributing factor in the efficiency and efficacy of a genetic algorithm.

Table 1. Fine tuning of crossover probability (p_c) between 0.5 to 1.0 keeping probability of mutation, p_m=0.001 fixed for n=10

Probability of cross over	0.5	0.6	0.7	0.8	0.9	1.0
% of success	37	43	61	94	100	73

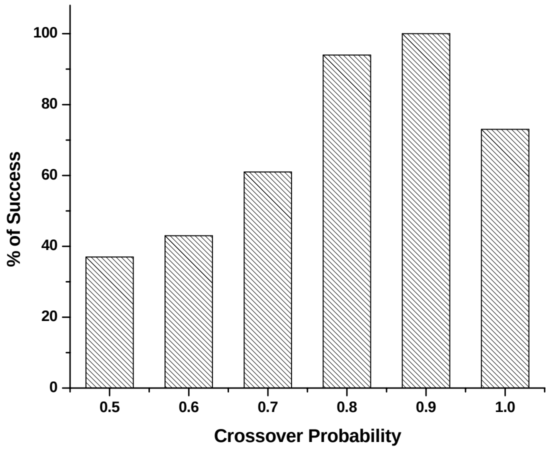

Fig. 1. Variation of success rate of MI-LXPM with crossover probability for n=10

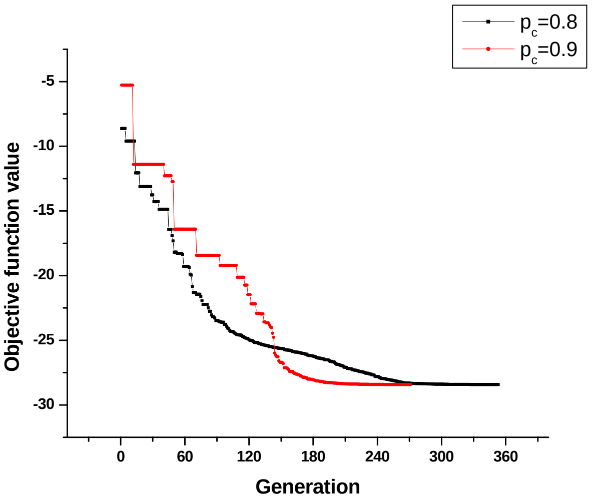

Fig. 2. Variation of objective function values corresponding to different crossover probabilities

5.2 Parameter setting for PSO

In literature, different values of these parameters are used. In this paper we set swarm size $S = 100$ and the cognitive and social parameters c_1 and c_2 of PSO are set to be 2.8 and 1.3 as taken in literature[3]. Constriction coefficient χ is calculated from Eq. (8). The inertia weight is set equal to 0.5. The total number of simulations considered is 100. The criterion to terminate the algorithm is reaching maximum number of iterations which is set to be 100,000.

6 Discussion of results

All the observations using MI-LXPM and PSO are recorded in Tab. 2 along with the earlier published results given in [21] and [2] using two algorithms namely DELP (The local search based differential evolution algorithm for potential minimization) and binary coded GA. Tab. 2 shows the number of function evaluations (average, minimum, maximum) using MI-LXPM and average function evaluations using GA (binary coded), DELP for the given potential energy function. For PSO average of minimum number of function evaluations required to reach the best energy value achieved by PSO in each run.

From Tab. 2, it is clear that MI-LXPM reached the global minimum function value with much less effort, in terms of function evaluation than other algorithms.

Table 2. Comparison of functional evaluations using $GA^{*[2]}$, $DE^{**[21]}$, MI-LXPM and PSO for a cluster containing atoms from 3 to 15

	GA^*	DE^{**}	MI-LXPM					PSO		
n	f_{avg}	f_{avg}	Cluster Energy	Succ %	f_{min}	f_{max}	f_{avg}	Minimum Obtained Energy	Succ %	f_{avg}
3	1550	1973	-3.0000	100	201	507	341	-3.0000	100	13970
4	3673	10359	-6.0000	100	732	1857	1144	-6.0000	100	68400
5	8032	31443	-9.1038	100	4778	6937	5929	-9.1038	100	220800
6	31395	67953	-12.7120	100	7935	10997	9271	-12.2991	74	306854
7	48900	103449	-16.5053	100	14025	36220	24253	-15.0429	60	389641
8	121247	207776	-19.8214	100	23844	40982	31107	-19.1671	0	421032
9	346397	1463183	-24.1133	100	29379	75035	57208	-21.8320	0	516890
10	721370	4653805	-28.4225	100	99513	197009	136846	-24.2940	0	623768
11	–	–	-32.7659	70	88483	241797	197514	-26.4812	0	687213
12	–	–	-37.9676	100	139733	451858	279910	-32.1131	0	740265
13	–	–	-44.3268	84	171149	521437	327712	-33.2150	0	789542
14	–	–	-47.8451	100	301615	614189	456716	-34.4274	0	847646
15	–	–	-52.3226	64	321611	649368	485462	-37.3204	0	996421

To further analyze the performance of MI-LXPM a graphical representation in the form of box plot is shown in Fig. 3 and the best performer is marked with star.

The important fact to notice is the presence of outliers in box plot representation in each of the algorithm considered which indicate an extreme of behavior of the problem under study thus showing the complexity of the problem.

Further the relative error for PSO using known minimum function value is defined by

$$\text{Relative Error (RE)} = \left| \frac{\text{Best Optima Obtained(Minimum function value)-Global Optima}}{\text{Global Optima}} \right|, \qquad (10)$$

which is shown in Fig. 4 and it is clear from the figure that as the number of atoms of the cluster goes on increasing the relative error also increases which indicates the complexity of the problem. Finally from computational results we can observe that MI-LXPM outperformed all other algorithms for solving the L-J potential problem. The following are the optimized structures of the L-J cluster obtained using MI-LXPM.

7 Conclusions

A Lennard-Jones potential of an atomic cluster up to 15 atoms is minimized which is very useful in predicting the stable structure of the molecule which in turn dictates majority of its properties. Computa-

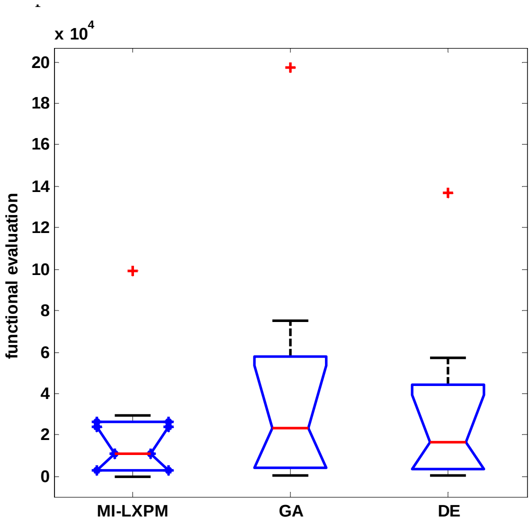

Fig. 3. Box Plot showing function evaluations for $n = 3$ to $n = 10$ using MI-LXPM, GA, DE

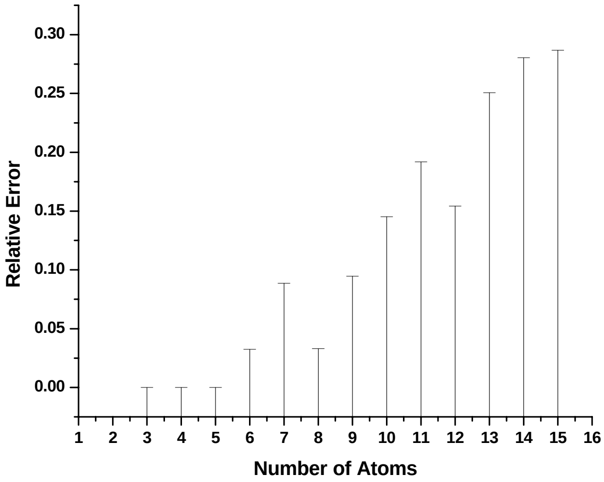

Fig. 4. Relative error using PSO

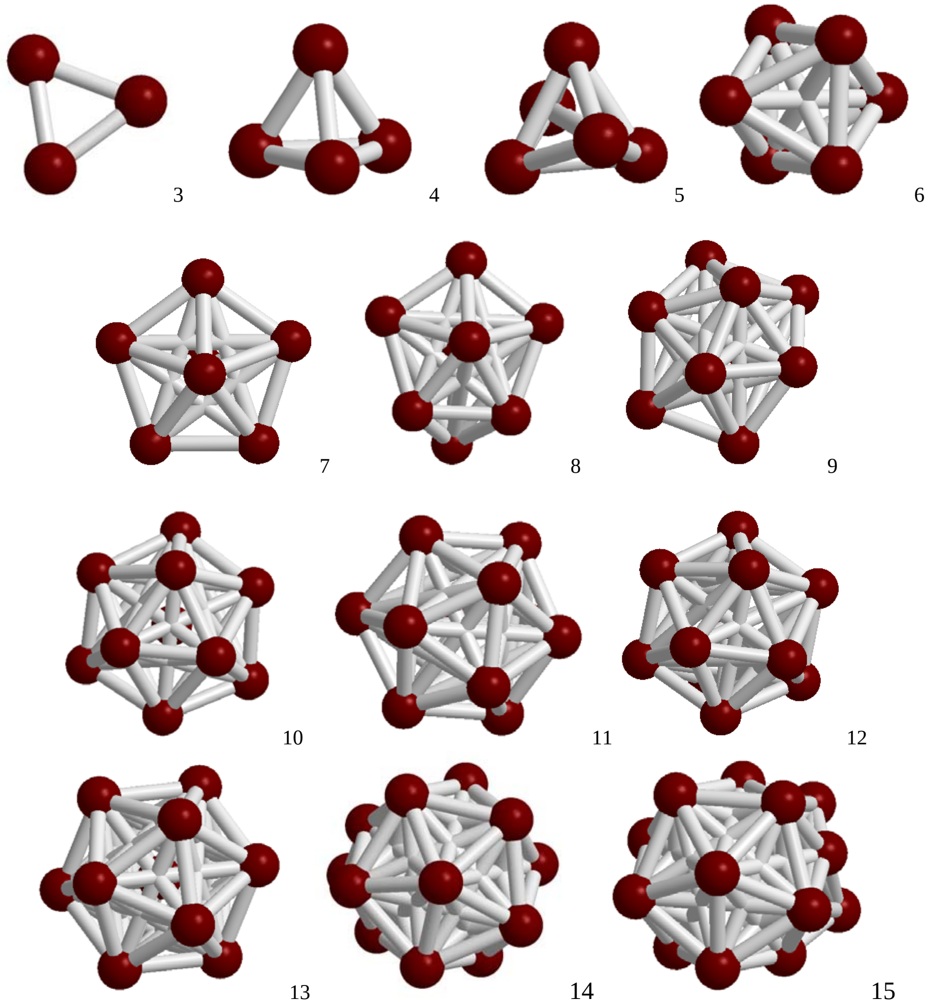

Fig. 5. Optimized structures obtained using MI-LXPM of L-J cluster of atoms from $n = 3$ to $n = 15$

tional results are obtained and presented using MI-LXPM and PSO and are compared with earlier published results[2, 21] and it is clearly seen (Tab. 2) that MI-LXPM performs far better than the other algorithms used. It is important to notice that the comparison of performance was based on the number of function evaluations, necessary to achieve the global minima rather than on the overall time of the search.

Although PSO is also an efficient search technique for solving complex optimization problems but it is not successful for L-J problem. It might be because of the fact that L-J problem is highly multi-modal in nature and the cooperation among the particles of swarm might limit the search space exploration as the position update rule creates a bias to move towards the points which have already been explored. And because of this it may stuck in the local optima. Whereas MI-LXPM uses Laplace Crossover and Power Mutation; Laplace Crossover is basically a parent centric approach i.e. the offsprings produced by this approach are proportional to the spread of parents. Therefore, it preserves the maximum properties of parents, so convergence is fast, and its combination with power mutation also maintains the diversity in the population.

Hence, MI-LXPM algorithm performs better than the other algorithms. However, MI-LXPM becomes slow as we take more atoms in the cluster. This is because of the complexity of the problem which increases with the increase in number of atoms in cluster. Therefore, improving MI-LXPM for larger problems will form the basis of our future research.

References

[1] Http://physchem.ox.ac.uk/ doye/jon/structures/LJ.html.

[2] M. Ali, R. Smith, S. Hobday. The structure of atomic and molecular clusters, optimised using classical potentials. *Computer Physics Communications*, 2006, **175**: 451–464.

[3] A. Carlisle, G. Dozier. An off-the-shelf pso. **in:** *Proceeding of Workshop on Particle Swarm Optimization, Indianapolis*, 2001.

[4] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. **In:** *Proceeding of ICEC, Washington, DC*, 1999, 1951–1957.

[5] S. Darby, T. Mortimer-Jones, et al. Theoretical study of cu-au nanoalloy clusters using a genetic algorithm. *Journal of Chemical Physics*, 2002, **116**: 1536–1550.

[6] D. Deaven, K. Ho. Molecular-geometry optimization with a genetic algorithm. *Physics Revised Letters*, 1995, **75**: 288–291.

[7] D. Deaven, N. Tit, et al. Structural optimization of lennard-jones clusters by a genetic algorithm. *Chemical Physics Letters*, 1996, **256**: 195–200.

[8] K. Deep, K. Singh, et al. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathmatics Computer*, 2009, **212**: 505–518.

[9] K. Deep, M. Thakur. A new crossover operator for real coded genetic algorithms. *Applied Mathmatics Computer*, 2007, **188**: 895–912.

[10] K. Deep, M. Thakur. A new mutation operator for real coded genetic algorithms. *Applied Mathmatics Computer*, 2007, **193**: 211–230.

[11] P. Dove. *The Structure: Thermodynamics and Dynamics of Atomic Clusters*. Department of Chemistry, University of Cambridge, 1996.

[12] N. Dugan, S. Erkoc. Genetic algorithm monte carlo hybrid geometry optimization method for atomic clusters. *Computer Mathmatics Science*, 2009, **45**: 127–132.

[13] R. Eberhart, Y. Shi. Comparing inertia weights and constriction factors in particle swarm optimization. **in:** *Proceeding of Congress on Evolutionary Computing*, vol. 1, 2000, 84–88.

[14] B. Hartke. Efficient global geometry optimization of atomic and molecular clusters. *Global Optimization*, 2006, **85**: 141–168.

[15] A. Herzog, V. Spravedlyvyy, et al. Adaptation of shape of dendritic spines by genetic algorithm. *Lecture Notes Computer Science*, 2004, **3215**: 476–482.

[16] M. Hoare. Structure and dynamics of simple microclusters. *Advanced Chemistry Physics*, 1979, **40**: 49–135.

[17] R. Hodgson. Particle swarm optimization applied to the atomic cluster optimization problem. **In:** *Proceeding of the Genetic and Evolutionary Computation Conference*, 2002, 68–73.

[18] J. Kennedy, R. Eberhart. Particle swarm optimization. **In:** *Proceeding of IEEE International Conference Neural Network*, 1995, 1942 – 1948.

[19] R. Leary. Global optima of lennard-jones clusters. *Journal of Global Optima*, 1997, **11**(1): 35–53.

[20] R. Leary. Global optimization on funneling landscapes. *Journal of Global Optima*, 2000, **18**(4): 367–383.

[21] N. Moloi, M. Ali. An iterative global optimization algorithm for potential energy minimization. *Computer Optimization Applied*, 2005, **30**(2): 119–132.

[22] J. Northby. Structure and binding of lennard-jones clusters: $13 \leq n \leq 147$. *Journal of Chemical Physics*, 1987, **87**: 6166–6178.

[23] D. Wales, J. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *Journal of Chemical Physics*, 1997, **101**: 5111–5116.

[24] G. Xue. Improvements on the northby algorithm for molecular conformation: Better solutions. *Journal of Global Optimization*, 1994, **4**(4): 425–440.