A dynamic lattice searching method for fast optimization of Lennard-Jones clusters

ARTICLE *in* JOURNAL OF COMPUTATIONAL CHEMISTRY · NOVEMBER 2004

Impact Factor: 3.59 · DOI: 10.1002/jcc.20096 · Source: PubMed

3 AUTHORS:

Xueguang Shao
Nankai University
230 PUBLICATIONS **3,575** CITATIONS
SEE PROFILE

Longjiu Cheng
Anhui University
43 PUBLICATIONS **558** CITATIONS
SEE PROFILE

Wensheng Cai
Nankai University
201 PUBLICATIONS **2,606** CITATIONS
SEE PROFILE

A Dynamic Lattice Searching Method for Fast Optimization of Lennard–Jones Clusters

XUEGUANG SHAO, LONGJIU CHENG, WENSHENG CAI
Department of Chemistry, University of Science and Technology of China, Hefei, Anhui, 230026, People's Republic of China

Received 17 February 2004; Accepted 10 May 2004
DOI 10.1002/jcc.20096
Published online in Wiley InterScience (www.interscience.wiley.com).

Abstract: A highly efficient unbiased global optimization method called dynamic lattice searching (DLS) was proposed. The method starts with a randomly generated local minimum, and finds better solution by a circulation of construction and searching of the dynamic lattice (DL) until the better solution approaches the best solution. The DL is constructed adaptively based on the starting local minimum by searching the possible location sites for an added atom, and the DL searching is implemented by iteratively moving the atom located at the occupied lattice site with the highest energy to the vacant lattice site with the lowest energy. Because the DL can greatly reduce the searching space and the number of the time-consuming local minimization procedures, the proposed DLS method runs at a very high efficiency, especially for the clusters of larger size. The performance of the DLS is investigated in the optimization of Lennard–Jones (LJ) clusters up to 309 atoms, and the structure of the LJ_{500} is also predicted. Furthermore, the idea of dynamic lattice can be easily adopted in the optimization of other molecular or atomic clusters. It may be a promising approach to be universally used for structural optimizations in the chemistry field.

 J Comput Chem 25: 1693–1698, 2004

Key words: global optimization; local optimization; Lennard–Jones clusters; dynamic lattice

Introduction

Global optimization is a challenging problem in many sciences and engineering fields today.[1] In chemical fields, the applications of global optimization include finding the lowest energy configuration of a molecular system or finding the lowest energy conformations of molecular or atomic clusters.[2–7] These problems are often nondeterministic polynomial-time (NP)-hard due to the large quantity of parameters to be optimized and the fact that the number of local minima grows exponentially with the problem size. Therefore, establishing a highly efficient global optimization method is a great challenge of computational chemistry.

Finding the global minimum of the Lennard–Jones (LJ) cluster is one of these problems.[1,8,9] LJ clusters consist of identical atoms interacting by an LJ potential:

$$V(r) = 4\varepsilon \sum_{i=1}^{N-1} \sum_{j>i}^{N} \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] \quad (1)$$

where N is the number of atoms and r_{ij} represents the distance between atoms i and j, which can be determined by the position of each atom, and the generally used values of ε and σ are 1. It is not only interesting as a model for heavy inert gases but also serves as a popular benchmark system for global optimization algorithms. In fact, despite the simplicity of the potential function, finding the global minima of LJ clusters has been a challenging problem even for a small N. Lots of methods have been developed for the problem, such as basin-hopping and its variant,[1,10–14] fast annealing evolutionary algorithm (FAEA),[15,16] random tunneling algorithm (RTA),[17,18] genetic algorithm (GA),[19–21] simulated annealing (SA),[22–24] etc. Furthermore, modeling methods are also proposed for making the problem easier.[25,26]

The modeling method is the most powerful one for the LJ clusters. For example, many global minima for $N \leq 150$ were found for the first time by using lattices derived from icosahedron,[25] and a similar strategy was used for $N \leq 309$.[26] These methods are powerful for a general case, but they cannot find the global minimum when the real configuration is different from the model.

Correspondence to: X. Shao; e-mail: xshao@ustc.edu.cn

Contract/grant sponsor: Outstanding Youth Fund (from the National Natural Scientific Foundation of China); contract/grant number: 20325517.

Contract/grant sponsor: Teaching and Research Award Program for Outstanding Young Teachers (TRA POYT).

Recently, there were a number of successful applications of unbiased search methods for LJ clusters. One of the most powerful methods is the basin-hoping method.[1,10,11] Almost all global minima listed in the Cambridge Cluster Database (CCD)[27] were reproduced for $N \leq 110$. In ref. 12, a variant of the basin-hopping method, called monotonic sequence basin-hopping (MSBH), was applied for $N \leq 110$, and had a better performance than the original version for some magic numbers ($N = 75$–77, 98, 102–104). More recently, a powerful evolutionary global optimization method, named conformational space annealing (CSA),[28] was applied for $N \leq 201$, and obtained a very high successful rate for most of N. It is a hybrid of Monte Carlo with minimization,[29] GA, and SA. Also, there are some other powerful unbiased methods for LJ clusters, such as the hierarchical greedy algorithm (HGA),[30] FAEA,[15,16] and RTA.[17,18]

Due to the large number of parameters to be optimized and the continuity of the searching space, an efficient local minimization (LM) method is necessary. The limited-memory quasi-Newton method (L-BFGS)[31] is widely adopted for LM of LJ clusters. However, LMs are very time-consuming when the atom number N is large. For example, mean CPU time needed by one LM is about 0.5 s for $N = 300$ using an Itanium2 processor (1.5 GHz). Most unbiased global optimization methods are based on random mutations or random moving atoms of the out layer, and LMs are necessary after those operations. Therefore, the number of LMs is always very large before convergence. For the fastest case, a number of LMs needed for convergence is 1256 at $N = 110$ even if a simple greedy strategy is adopted.[12] Of course, the number will be much larger for population-based evolutionary strategies. Therefore, the key for reducing computational time is to reduce the number of LMs.

In our experiences, it was found that just specific positions could be located at after LM when one atom is added to a fixed cluster. Therefore, all the possible locations surrounding an unknown local minimum can be found by LM. In this study, all these locations are called the dynamic lattice (DL). With the established DL, if we iteratively move the atom with high energy (called an active atom) to the lattice site with lower energy, a new local minimum with lower energy can be obtained, and by repetition of the procedure, it will be possible to find out the globally minimized configuration. Based on this strategy, a novel global optimization method, called dynamic lattice searching (DLS), is proposed. Because consumed time for establishing the DL is very short compared to that for LMs, and the speed of DL searching is very high, the proposed method has proven to be very fast in convergence speed and very efficient in optimization ability. Furthermore, it should be noted that the DLS method is unbiased due to the fact that it starts from a randomly generated local minimum and the DL is constructed adaptively. Therefore, the DLS method can be utilized as a universal method to solve the problems in optimization of other molecular or atomic clusters.

Method

Dynamic Lattice Searching (DLS) Method

The basic idea of the DLS method consists of the fact that just specific positions will be located after LM when one atom is added to a fixed cluster. Therefore, the first step of the method is to randomly generate a starting local minimum. Then, perform a circulation of DL construction, DL searching, and LM. The DL construction procedure will adaptively generate the DL around the starting local minimum, the DL searching procedure will get some low-energy candidates, and the LM procedure will obtain the corresponding local minima of these candidates. If the best one of the obtained local minima has lower potential energy than the starting local minimum, it will be taken as the starting local minimum of the next generation and repeat the above circulation. Otherwise, the current starting local minimum is taken as the result of this calculation. Detailed procedures of DL construction and DL searching will be introduced in following sections.

Because only the low-energy candidates are minimized in each generation and DLS utilizes the monotonic descent sequence strategy, which is a kind of simple greedy strategy, DLS has a very high convergence speed, and the number of LMs needed for convergence is very small. However, DLS may converge at various funnels of the potential surface due to the adoption of the greedy strategy. We should not expect to find the global minima at one run of DLS. Therefore, DLS will be restarted for N_{runs} times from fresh randomly generated local minima in practical calculations. It is interesting that, when N_{runs} is reasonably big enough, for example, $N_{\text{runs}} \geq 1000$, the best solutions in different runs are always distributed in the different funnels of the potential surface, even for the steep funnels with vary narrow width. This may be the reason to account for the efficiency of the DLS method.

DL Construction

To a given local minimum of an LJ cluster, at first, the coordinates of each atom are rescaled to make all the atoms around the center of the cluster, and obtain the maximum radius of the cluster (R_{max}). Then, N_P primary lattice sites are generated with a uniform distribution on the surface of a sphere with radius $R_0 = R_{\text{max}} + \tau$, where τ is a positive constant ($\tau = 0.2$ in this study). All the N_P sites compose the primary lattice (PL). Actually, the nth shell of the Mackay icosahedral cluster was used to generate the PL in this study, and the radius of the PL is rescaled to R_0. For $n = 3$, 4, 5, 6, 7, and 8, the value of N_P is 92, 162, 252, 362, 492, and 642, respectively. It should be noted that the use of the Mackay icosahedron is just to simulate a uniform distribution on the surface of a sphere, and the topological information of the Mackay icosahedron is not utilized. Therefore, this procedure is unbiased. Finally, keeping the N atoms in the local minimum structure fixed, put an atom into each site of the PL, and perform sub-LM on the single atom to get a new position for the site. All the new sites form a set of DL candidates. If the distance between a pair of DL candidates is smaller than δ, where δ is a positive constant ($\delta = 0.05$ in this study), the two sites are thought to be the same one. By such an approach, the average size (N_{DL}) of the constructed DL over 1000 runs is shown in Figure 1. It can be seen that the average size increases lower than linearly with the number of atoms in the cluster. This indicates that the searching space in the DL searching procedure is greatly reduced.

Obviously, the number of parameters to be optimized in the sub-LM is only three, and the evaluation and gradient function of the sub-LM is about $2/(N - 1)$ complexity to that of the LM on

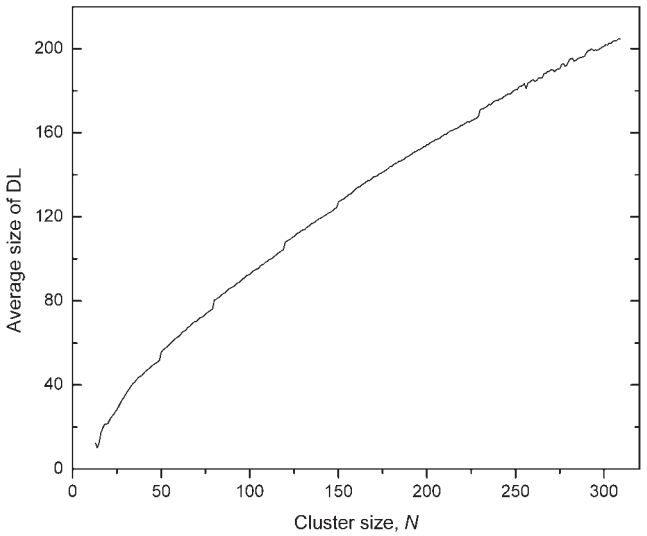

Figure 1. Average size of DL as a function of cluster size.

an N-atom cluster. Therefore, sub-LM is much faster than the generally used LM on all the atoms of a cluster. For example, CPU time needed by one sub-LM is about 1/500 of that needed by one LM at $N = 100$.

DL Searching

The aim of the DL searching procedure is to find some low-energy solutions (low-energy configuration of an LJ cluster) by searching the current DL. At first, the starting local minimum is searched to find N_{mov} atoms with the highest subenergy, and merge them into the DL constructed above. The total size of the DL becomes $N'_{\mathrm{DL}} = N_{\mathrm{DL}} + N_{\mathrm{mov}}$, and the atom number of the remaining cluster becomes $N' = N - N_{\mathrm{mov}}$. Then calculate the following subenergy items:

Potential energy of the remaining cluster:

$$E_{\mathrm{C}} = 4\varepsilon \sum_{i=1}^{N'-1} \sum_{j>i}^{N'} \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right] \tag{2}$$

Potential energy between the site i ($i = 1, 2, \ldots, N'_{\mathrm{DL}}$) of the DL and the remaining cluster:

$$E_{\mathrm{DLC}}(i) = 4\varepsilon \sum_{j=1}^{N'} \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right] \tag{3}$$

Potential energy between sites i and j ($i, j = 1, 2, \ldots, N'_{\mathrm{DL}}$) of the DL:

$$E_{\mathrm{DL}}(i,j) = 4\varepsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right] \tag{4}$$

In this way, DL searching becomes a combinational optimization problem with the searching space $C_{N'_{\mathrm{DL}}}^{N_{\mathrm{mov}}}$. A possible solution is to select N_{mov} sites from the DL, and the total potential energy of the solution can be given by the evaluation function:

$$E = E_{\mathrm{C}} + \sum_{i=1}^{N_{\mathrm{mov}}} E_{\mathrm{DLC}}(i) + \sum_{i=1}^{N_{\mathrm{mov}}-1} \sum_{j>i}^{N_{\mathrm{mov}}} E_{\mathrm{DL}}(i,j) \tag{5}$$

When N_{DL} and N_{mov} are small, it is not difficult to search the DL even using an exhaust searching method. According to Figure 1, exhaust searching is available at about $N \leq 110$ and $N_{\mathrm{mov}} \leq 6$. But the consumed time will be unacceptable for larger N or N_{mov}. On the other hand, if a small value is used for the number of active atoms (N_{mov}), the number of movable atoms will also be small, and thus the optimization ability will be very limited.

Therefore, in this study, the principle of the simple greedy method (SGM) is utilized to search the DL. Steps in a single SGM search procedure can be summarized as follows:

1. Select N_{mov} sites randomly from the DL to generate a starting solution S_0. The energy of the solution can be obtained with eq. (5).
2. With the current solution S_k ($k = 0, 1, 2, \cdots$), calculate the energy of each site in S_k. Then, move the atom located at the occupied site with highest energy to the vacant site with lowest energy in the DL to generate a new solution S'_k.
3. If $E(S'_k) < E(S_k)$, take S'_k as the starting solution of the next generation (S_{k+1}) and return to step 2. Otherwise, terminate the iteration with the current solution S_k as the best solution of this SGM search.

The SGM search procedure is very fast due to the greedy strategy. However, it should be noted that the best solution of an SGM search is obtained only by iteratively moving an atom at a worse site to a better site. It may converge at solutions in various funnels of the potential surface. Therefore, to find the funnel containing the better solution, the SGM procedure will be repeated for N_{try} times from different random starting solution S_0. On the other hand, because the best solution of an SGM search is obtained without LM, there is generally a small gap from the solution to the bottom of the potential funnel after LM, and the gap may be different for the best solutions of the N_{try} times the SGM search. This will result in a problem that the best solution obtained in an SGM search with lower potential energy is not always lower after LM. Therefore, during the N_{try} times calculation, N_{best} solutions with lower potential energy, instead of only one best solution with lowest potential energy, will be recorded for further LM. The solution with the lowest potential energy after LM will be selected as the final solution of this DL searching procedure.

Results and Discussion

Searching Procedure of DLS

In the methods based on the greedy strategy, the search procedure from a starting local minimum to a final result will produce a

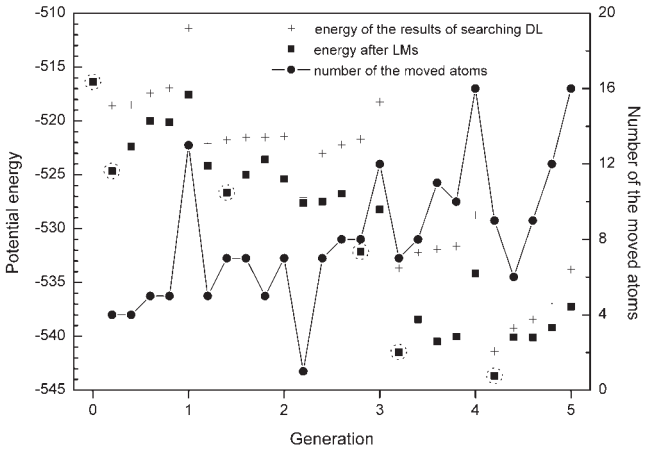

Figure 2. The searching procedure of DLS in the optimization of LJ_{98} with parameters $N_{best} = 5$ and $N_{mov} = 20$. The squares enclosed by dot circles denote the local minima within the monotonic descent sequence.

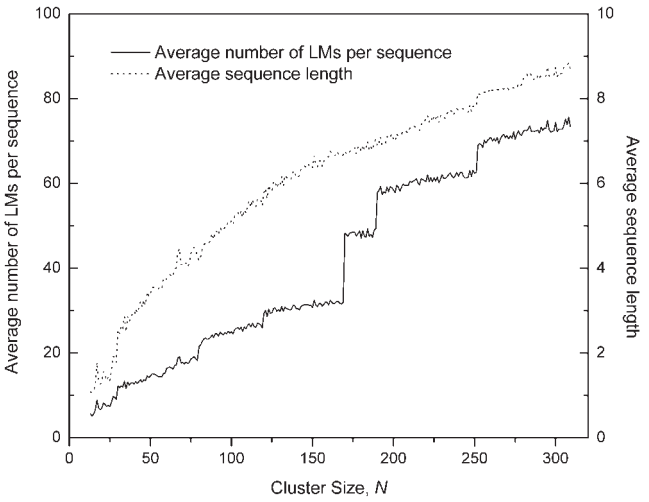

Figure 3. Average sequence lengths as a function of cluster size.

sequence of local minima with lower and lower potential energy. This sequence is generally called the monotonic descent sequence, and the number of the local minima in the sequence is called sequence length.

To illustrate the searching procedure and the performance of DLS, the monotonic descent sequence in finding the global minimum of LJ_{98} with parameter $N_{mov} = 20$ and $N_{best} = 5$ was shown in Figure 2. In the figure, the first square enclosed by a dot circle above 0 generation is the starting local minimum. With the constructed DL from the local minimum, 5 (N_{best}) best solutions are found within 300 (N_{try}) trials of DL searching, which are labeled with "+" above 0–1 generation in the figure, and the number of the moved atoms is recorded simultaneously as shown by the solid circles connected with the solid line. With LMs, the solutions will be minimized to their local minima labeled with the squares, and the best one (enclosed by a dot circle) will be chosen as the starting point for the next generation of DL construction and DL searching. The last square enclosed by a dot circle is the final solution of this run of DLS. From Figure 3, it can be seen that, in this run, the global minimum was found within only five generations, and the number of LMs is 22. Because the speed of the DL construction and the DL searching is very fast, the total consumed time of this run is only around 1 s. On the other hand, from the number of the actually moved atoms in each DL searching process, it can be found that, generally, simultaneously movement of more than four atoms are needed for finding a better solution. This operation is too difficult for the conventional optimization methods such as GA and SA.

Computational Results and Performance of DLS

At first, with the parameters listed in Table 1, DLS was run for the optimization of the LJ clusters with $13 \leq N \leq 309$. The aim of setting N_{runs} more than 1000 is to guarantee more than 10 hits of the global minima in most cases, especially for the cases of $N \leq 250$. It was found that all the known putative global minima for these clusters given in ref. 27 were reproduced with a very high efficiency.

The average sequence length and the average number of LMs per sequence in DLS calculations are given in Figure 3 as a function of cluster size. Clearly, both the average length and the average number of LMs (including the initial one) increase in a trend lower than linearly with N. However, from Figure 3, there are several obvious jump points in the curve of the average number of LMs. With Table 1, it is easy to find that these points are caused by the change of the parameters N_{mov} and N_{best}, because the searching space increases with N_{mov} and the number of LMs equals to (or, sometimes, less than) the product of N_{best} and the sequence length. This result indicates that DLS can greatly reduce the searching space and the number of the time-consuming LMs, and furthermore, the reduction can be controlled by the parameters N_{mov} and N_{best}. For example, the number of LMs in MSBH[12] is 1256 in the optimization of the LJ_{110} cluster, but the average number of LMs per sequence in DLS for the cluster is only 26.5. Therefore, DLS is a method with a very fast convergence speed, even compared with MSBH.

Table 1. Parameters Used in DLS.

N	N_{mov}	N_p	N_{try}	N_{best}	N_{runs}[a]
13–49	10	92	100	4	1000
50–79	15	162	200	4	1000
80–119	15	252	300	5	1000
120–149	20	362	350	5	1000
150–169	20	492	400	5	1000
170–189	20	492	450	8	2000
190–199	20	492	500	10	2000
200–229	20	492	550	10	2000
230–251	20	642	550	10	2000
252–309	25	642	700	10	2000
500	40	812	1000	10	10,000

[a]For the magic numbers (38, 75–77, 98, 102–104) $N_{runs} = 10{,}000$.

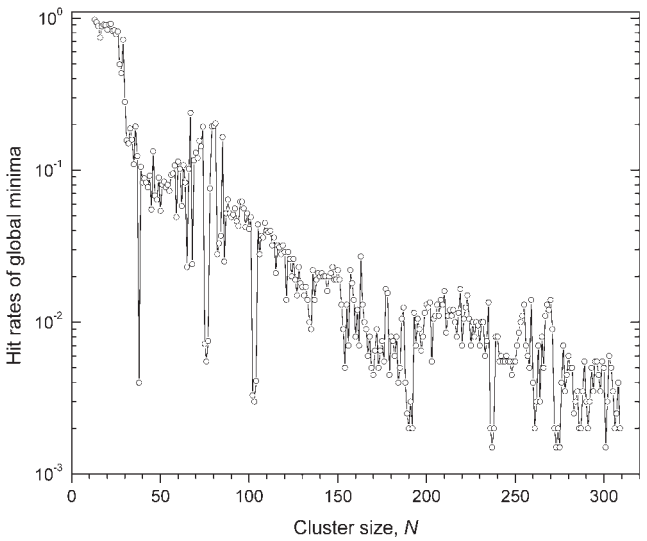

Figure 4. Plot of the hit rates in the optimizations of the LJ clusters with $13 \leq N \leq 309$.

The successful rate or hit rate, that is, the number of runs hitting the global minimum over the total number of runs, is another key factor to evaluate an unbiased optimization method. Figure 4 shows the observed fractional hit rate as a function of cluster size N in the calculations with the parameters listed in Table 1. Clearly, the fractional hit rates for the clusters with $N \leq 29$ are very high due to the simplicity of their potential surface, and then it has a quick drop. After the drop, the hit rate goes down slowly with the increase of the cluster size N, but the descending curve is full of jumps in different magnitudes. The big jumps are found corresponding to those clusters with different configurations from their neighbors. Furthermore, there are four obvious downwards peaks in the curve within $N \leq 250$. The first one is LJ_{38}, with face-centered cubic (fcc) packing and the others are LJ_{75-77}, $LJ_{102-104}$, $LJ_{188-192}$, and $LJ_{236-238}$ with decahedral packing, which are difficult cases for unbiased optimization methods. However, it is very surprising that the hit rate for LJ_{98} with the Leary tetrahedral[11] configuration, which is generally known as the most difficult case for optimization, is just a similar value with its neighbors. Regardless of the exceptions, from the curve after $N = 150$, especially after $N = 280$, it can be found that the descending trend of the hit rates with the cluster size becomes very slow. This indicates that the configuration of a cluster is the key factor affecting the hit rate, that is, the difficulty of the optimization. On the other hand, this also indicates that DLS method may be a promising tool for clusters with large size.

From the results above, the hit rates of DLS are not higher than the conventional methods. The lowest hit rate is only 0.15%. However, the high convergence speed can compensate for its hit rate, and makes it a very fast unbiased optimization method. At first, the average number of LMs needed by one hit in the optimization of some selected clusters is compared with the MCM basin-hoping algorithm,[10] MSBH,[12] and HGA[30] in Table 2. It can be seen that, for the clusters with $N \leq 60$, the number is in the same level. However, for the clusters with $N \geq 70$, the numbers

Table 2. Mean LMs per Hit of Global Minima for Selected N by DLS and Other Unbiased Global Optimization Method.

N	MCM	MSBH	HGA	DLS
20	35	34		8.3
30	1140	739		43.5
38[a]	2674	2875		3240
40	208	279		152.6
50	251	460		270
60	384	388		143
70	1527	1526		137
75[a]	$\sim 10^7$	152,000	9259	2586
77[a]			13,158	2551
80	2540	2009		110
90	3024	4699		501
98[a]	$>10^6$	180,000	5660	597
100	7610	9128		610
102[a]	$>10^6$	36,028		7733
110	11,362	40,420		663
150				1398
188[a]				12,234
200				4494
236[a]				30,875
250				11,347
300				14,875

[a]Nonicosahedral global minima.
Results of MCM and MSBH are from ref. 12, and results of HGA are from ref. 30.

are significantly improved, especially for the clusters with nonicosahedral configurations.

Figure 5 shows the average number of LMs per hit of global minima for the clusters with $13 \leq N \leq 309$, the average CPU time per hit is also shown in the figure. The computations were carried out on an HP cluster with Intel Itanium2 Madsion proces-

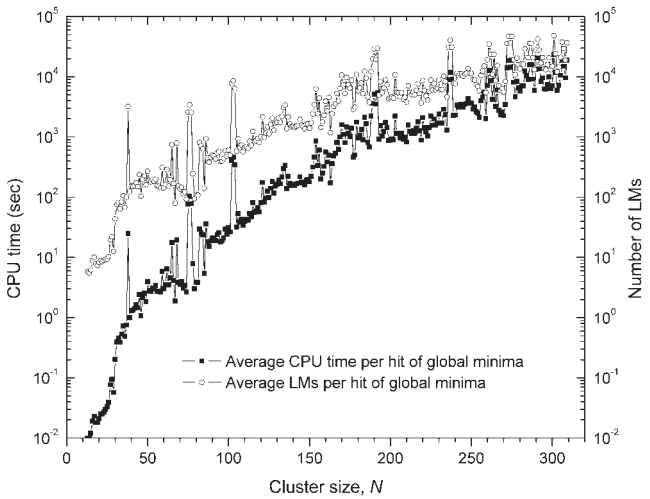

Figure 5. Average CPU time (seconds) and average number of LMs per hit of global minima for clusters with $13 \leq N \leq 309$.

sors (1.5 GHz). The program is written in C language, and the compiler is Intel(R) C++ Compiler for Linux IA-64 7.1. The same LM method was used as in CSA,[28] but, compared with $|G|/\sqrt{3N} \leq 1.0 \times 10^{-3}$ used in CSA, a more precise stop criterion $|G|/\sqrt{3N} \leq 1.0 \times 10^{-4}$ was used in this work, where G is the gradient of the LJ potential function. By comparison of the average CPU time in Figure 5 with that in Figure 2 of ref. 28 (obtained with a processor Athlon 1.667 GHz), it can be found that DLS is a much faster method than CSA. For example, the CPU time for one hit of the global minima of LJ_{75}, LJ_{98}, and LJ_{201} in ref. 28 is nearly 10^4, 10^4, and 10^5 s, respectively. However, in this work, the corresponding time is only 75.6, 29.3, and 935 s, respectively.

On the other hand, the principle and the methodology of the DLS are very simple. The CPU time consumed by DL construction and DL searching is no more than 30% of the total time for most cases, for example, 25% for $N = 100$, 18% for $N = 200$ and, 15% for $N = 300$, when parameters given in Table 1 are used in the calculations.

To further investigate the performance of DLS with a larger N, it is applied to the global optimization of LJ_{500} using the parameters given in Table 1. The obtained structure with lowest energy is found to be icosahedral packing with potential energy $E = -3382.693487$, which is the same structure with the result reported in ref. 30. The hitting rate of the lowest energy structure is 17/10,000, and the number of LM per hit is around 70,000, which is slightly bigger than that of the HGA. This indicates that DLS may be a promising tool for finding the global minima of LJ clusters with larger N.

Conclusion

A highly efficient unbiased method, called dynamic lattice searching (DLS), is proposed applied to the optimization of LJ clusters. In DLS, the global minimum of a cluster is found by iteratively finding a better solution from the randomly generated starting local minimum with a simple greedy strategy (SGM) operation on the adaptively constructed dynamic lattices (DLs). By using the DL, both the searching space and the number of the time-consuming local minimization procedures can be greatly reduced. With the successful optimization of the LJ clusters with $13 \leq N \leq 309$, it was proven that the DLS method runs in a very high efficiency, especially for the large-size clusters. On the other hand, the idea of dynamic lattice construction can be adopted in optimization of other molecular or atomic clusters. It may be a useful approach to reduce the searching space for global optimization problems and universally used for structural optimizations in chemistry field.

Acknowledgments

Thanks to the USTC-HP High Performance Computing Joint Lab (HPCJL) for providing the HP-cluster.

References

1. Wales, D. J.; Scheraga, H. A. Science 1999, 285, 1368.
2. Doye, J. P. K.; Wales, D. J. Phys Rev B 2001, 64, 235409.
3. Yoo, S.; Zeng, X. C.; Zhu, X. L.; Bai, J. J Am Chem Soc 2003, 125, 13318.
4. Baletto, F.; Mottet, C.; Ferrando, R. Phys Rev Lett 2000, 84, 5544.
5. Cleveland, C. L.; Landman, U. J Chem Phys 1991, 94, 7376.
6. Besley, N. A.; Johnston, R. L.; Stace, A. J.; Uppenbrink, J. J Mol Struct (Theochem) 1995, 341, 75.
7. Hearn, J. E.; Johnston, R. L. J Chem Phys 1997, 107, 4674.
8. Doye, J. P. K.; Wales, D. J.; Berry, R. S. J Chem Phys 1995, 103, 4234.
9. Doye, J. P. K.; Wales, D. J. Chem Phys Lett 1995, 247, 339.
10. Wales, D. J.; Doye, J. P. K. J Phys Chem A 1997, 101, 5111.
11. Leary, R. H.; Doye, J. P. K. Phys Rev E 1999, 60, R6320.
12. Leary, R. H. J Global Optim 2000, 18, 367.
13. White, R. P.; Mayne, H. R. Chem Phys Lett 1998, 287, 463.
14. Pillardy, J.; Liwo, A.; Scheraga, H. A. J Phys Chem A 1999, 103, 9370.
15. Cai, W. S.; Shao, X. G. J Comput Chem 2002, 23, 427.
16. Cai, W. S.; Feng, Y.; Shao, X. G.; Pan, Z. X. J Mol Struct (Theochem) 2002, 579, 229.
17. Jiang, H. Y.; Cai, W. S.; Shao, X. G. Phys Chem Chem Phys 2002, 4, 4782.
18. Shao, X. G.; Jiang, H. Y.; Cai, W. S. J Chem Inf Comput Sci 2004, 44, 193.
19. Hartke, B. J Comput Chem 1999, 20, 1752.
20. Wolf, M. D.; Landman, U. J Phys Chem A 1998, 102, 6129.
21. Gregurick, S. K.; Alexander, M. H.; Hartke, B. J Chem Phys 1996, 104, 2684.
22. Xue, G. L. J Global Optim 1994, 4, 425.
23. Ma, J. P.; Straub, J. E. J Chem Phys 1994, 101, 533.
24. Schelstraete, S.; Verschelde, H. J Phys Chem A 1997, 101, 310.
25. Northby, J. A. J Chem Phys 1987, 87, 6166.
26. Romero, D.; Barrón, C.; Gómez, S. Comput Phys Commun 1999, 123, 87.
27. The complete up-to-date list of global minima of the LJ clusters can be found at http://brian.ch.cam.ac.uk/ for $N \leq 150$ and at http://www.vcl.uh.edu/~cbarron/LJ_cluster/researchpot.html for $147 \leq N \leq 309$.
28. Lee, J.; Lee, I. H.; Lee, J. Phys Rev Lett 2003, 91, 080201.
29. Li, Z.; Scheraga, H. A. Proc Natl Acad Sci USA 1987, 84, 6611.
30. Krivov, S. V. Phys Rev E 2002, 66, 025701.
31. Liu, D. C.; Nocedal, J. Math Prog B 1989, 45, 503.