Improvement on the Northby Algorithm for Molecular Conformation: Better Solutions

GUOLIANG XUE
Department of Computer Science and Electrical Engineering, The University of Vermont, Burlington, VT 05405, U.S.A. (E-mail: xue@cs.uvm.edu)

(Received: 20 October 1992; accepted: 4 July 1993)

Abstract. In 1987, Northby presented an efficient lattice based search and optimization procedure to compute ground states of n-atom Lennard–Jones clusters and reported putative global minima for $13 \leqslant n \leqslant 150$. In this paper, we introduce simple data structures which reduce the time complexity of the Northby algorithm for lattice search from $O(n^{\frac{5}{3}})$ per move to $O(n^{\frac{2}{3}})$ per move for an n-atom cluster involving full Lennard–Jones potential function. If nearest neighbor potential function is used, the time complexity can be further reduced to $O(\log n)$ per move for an n-atom cluster. The lattice local minimizers with lowest potential function values are relaxed by a powerful Truncated Newton algorithm. We are able to reproduce the minima reported by Northby. The improved algorithm is so efficient that less than 3 minutes of CPU time on the Cray-XMP is required for each cluster size in the above range. We then further improve the Northby algorithm by relaxing *every* lattice local minimizer found in the process. This certainly requires more time. However, *lower energy configurations* were found with this improved algorithm for $n = 65, 66, 75, 76, 77$ and 134. These findings also show that *in some cases, the relaxation of a lattice local minimizer with a worse potential function value may lead to a local minimizer with a better potential function value.*

Key words. Molecular conformation, data structures, improved algorithms, better solutions.

1. Introduction

The minimization of potential energy functions of clusters of atoms is known as the *molecular conformation* problem. Global minimizers (or *ground states*) of potential energy functions are of great interest to researchers in chemistry, biology, physics, and optimization. One of the fundamental problems in molecular conformation is the minimization of the pure Lennard–Jones potential function [5]. This problem is very hard to solve because it has many local minimizers. Hoare [7] has claimed that the number of local minimizers of an n-atom cluster grows as fast as the the function $O(e^{n^2})$. Nonetheless, many papers have been published on computational methods [1,2,4,6,7,8,10,12,13,16–23] and putative global minima for cluster sizes as large as $n = 150$ have been reported [7,16,19].

The most successful algorithm for minimizing Lennard–Jones clusters has been Northby's algorithm. Northby makes use of the IC and FC lattices (which are described in detail in Appendix A) in finding good starting points for continuous minimization. Suppose that we want to find the global minimizer for an n-atom cluster. We may first find an IC or FC lattice with $m \geqslant n$ points. We may then assign the n atoms to n of the m lattice points. With each such assignment, there

Journal of Global Optimization **4**: 425–440, 1994.

is a corresponding configuration of the cluster and an energy function value. We will call such a configuration a *lattice configuration*. Two lattice configurations are said to be neighbors if one can be obtained from another by reassigning an atom to an unassigned lattice point. A *lattice local minimizer* is a lattice configuration whose corresponding energy value is not greater than all of its neighbor's. The Northby algorithm first finds a set of lattice local minimizers and then *relaxes* those lattice minimizers by performing continuous minimization starting with those lattice minimizers. With this algorithm, Northby is able to publish putative global minima for cluster sizes ranging from 13 to 150 [16]. These values are believed to be global minima and serve as a standard to test algorithms for molecular conformation.

In Northby's algorithm, the lattice search part solves a discrete optimization problem by moving from one lattice configuration to another (neighboring) lattice configuration with a better function value. We call each of these move a *pivot* (or just a *move*). In Northby's implementation [16], each move takes $O(n^{\frac{5}{3}})$ for a cluster with n atoms. In [24], we have reduced the time complexity of each move to $O(n^{\frac{2}{3}})$ for a cluster with n atoms. Therefore, with the supercomputer CM-5 and a two-level simulated annealing algorithm, we have been able to solve the lattice search problem for cluster sizes as large as 100,000.

In this paper, we make two additional improvements on the Northby algorithm for Molecular Conformation problem. First, we note that by employing AVL trees [9], the time complexity per move can be reduced to $O(\log n)$ for an n-atom cluster if nearest neighbor potential is used. This improvement, combined with the improvement which we have made in [24], makes the Northby algorithm very fast. Less than 3 minutes of CPU time on the Cray-XMP is required for each cluster size in the studied range. Second, we relax *every lattice local minimizer* found instead of relaxing only those lattice local minimizers with best known potential function value. In this case, the algorithm requires more time. However, *lower energy configurations* were found with this improved algorithm for $n = 65$, 66, 75, 76, 77 and 134. These findings also show that *in some cases, the relaxation of a lattice local minimizer with a worse potential function value may lead to a local minimizer with a better potential function value.*

The rest of this paper is organized as follows. In Section 2, we describe the Molecular Conformation problem and the Northby algorithm. In Section 3, we discuss improvements on the Northby algorithm. In Section 4, computational results on the Cray-XMP are presented. Concluding remarks are given in Section 5.

2. Molecular Conformation and the Northby Algorithm

Given n atoms (points), $p_1, p_2, \dots, p_n$, in 3-dimensional Euclidean space, the total 2-body potential energy function is defined as

$$V_n(p) = \sum_{j=2}^{n} \sum_{i=1}^{j-1} v_{LJ}(\|p_j - p_i\|_2) , \tag{1}$$

where $v_{LJ}(r)$ is the Lennard–Jones potential function ([5]) defined as

$$v_{LJ}(r) = \frac{1}{r^{12}} - \frac{2}{r^{6}} \,. \tag{2}$$

The problem is to find a configuration (positions for the n points) such that the total potential energy function $V_n(p)$ is minimized.

Finding a global minimizer of $V_n(p)$ is extremely difficult except for very small cluster sizes. The difficulty is due to the fact that while it is always possible with a supercomputer and a local minimization algorithm (e.g. quasi-Newton method) to relax any reasonable initial configuration to some local minimizer, unless the starting configuration is in the catchment basin of the global minimizer, the minimizer found may not be the global minimizer. Hoare has claimed that the number of local minimizers in the potential energy surface of an n-atom Lennard-Jones cluster is about $O(e^{n^2})$. Thus, it is impractical to perform an undirected search for all local minimizers of the potential function in order to find the global minimizer, except for very small clusters.

Chemical physicists have learned from previous research that the "ground states" of Lennard–Jones clusters exhibit certain kind of lattice structures. So far, the most successful algorithms for computing ground states of Lennard–Jones clusters are based on lattice search followed by local minimization from the lattice minimizers, represented by the Northby algorithm [16]. As stated in [11], a critical assumption for lattice search based algorithms is that *a well-defined set of lattice structures contains at least one initial cluster configuration which relaxes to the ground state.* As described and supported by computational results in [16], the *IC* and *FC* lattices (see Appendix A for definitions and descriptions of these lattices) are well-defined lattice structures for the pure Lennard–Jones clusters. It is believed that in *most of the cases, the relaxation of a global lattice minimizer will results in a configuration with a lower energy than the relaxation of a non-global lattice local minimizer.*

Northby's algorithm for Molecular Conformation can be summarized as follows:

```
Algorithm 1: Northby's Algorithm for Molecular Conformation
  1. Fix the base lattice to either the IC lattice or the FC lattice. Choose the lattice search potential
     function as either the full Lennard–Jones potential function or the nearest-neighbor potential
     function.
  2. Repeat
       2.1. randomly generate a lattice configuration;
       2.2. pivot to a lattice local minimizer;
       2.3. store all lattice local minimizers with the best known potential function value and
            discard those with worse potential function values;
     Until the lowest potential function value has been repeated for 250 times.
  3. Relax each of the lattice local minimizers stored in step 2 with the full Lennard–Jones
     potential function.
```

Fig. 1. Northby's algorithm for Molecular Conformation.

Suppose that we want to find a lattice local minimizer of an n-atom cluster. Let us assume that we have chosen one of the two types of lattices for the lattice minimization. First, find the largest IC lattice which contains fewer than n points (if one of the IC lattices has exactly n points, we simply put all n atoms on the lattice points of that lattice and quit). Call this IC lattice the *core* and let N_{core} and I_{core} be the number of points in this IC lattice and the index set of this IC lattice, respectively. Next, find the next layer of IC (or FC) lattice which contains N_{surf} (surface) points. Let I_{surf} be its index set. If $N_{core} + N_{surf} = n$, we simply put all n atoms on the lattice points of the lattice and quit.

An initial configuration can be constructed by filling N_{core} atoms into the core lattice and randomly put the remaining $(n - N_{core})$ atoms onto the N_{surf} surface lattice sites. This is equivalent to partitioning the index set I_{surf} to two subsets I_{surf}^{filled} and I_{surf}^{vacant} such that $|I_{surf}^{filled}| = n - N_{core}$ and that site $i \in I_{surf}$ is filled with an atom if and only if $i \in I_{surf}^{filled}$.

Northby [16] computes the interaction matrix $VP(i, j)$, the pair interaction between an atom on site i and one on site j at the very beginning of the algorithm and stores it as a lookup table. After this is done, Northby's pivot algorithm for finding a lattice local minimizer can then be summarized as follows.

Algorithm 2: Northby's Pivot Algorithm for Lattice Search

1. {*Find the most loosely bound atom*}
   Find $i_{loose} \in I_{surf}^{filled}$ such that

$$i_{loose} = arg \max_{i \in I_{surf}^{filled}} \left\{ \sum_{j \in I_{core}} VP(i, j) + \sum_{j \in I_{surf}^{filled}, j \neq i} VP(i, j) \right\}. \tag{3}$$

   Site i_{loose} is called the most loosely bound filled site and the atom at that site is called the most loosely bound atom. Let $gain_{loose}$ be the maximum function value that the maximization problem in (3) achieves at i_{loose}. Apparently, this is the total contribution that the atom at site i_{loose} has towards the total potential energy.

2. {*Find the most tightly binding vacant site*}
   Find $i_{tight} \in I_{surf}^{vacant}$ such that

$$i_{tight} = arg \min_{i \in I_{surf}^{vacant}} \left\{ \sum_{j \in I_{core}} VP(i, j) + \sum_{j \in I_{surf}^{filled}, j \neq i_{loose}} VP(i, j) \right\}. \tag{4}$$

   Site i_{tight} is called the most tightly binding vacant site. Let $gain_{tight}$ be the minimum function value that the minimization problem in (4) achieves at i_{tight}. This is the new contribution that the atom at site i_{loose} has towards the total potential energy when moved to site i_{tight}.

3. {*Pivot on the Lattice*}
   If $gain_{tight} - gain_{loose} < 0$ then move the atom at site i_{loose} to site i_{tight} and goto step 2. Otherwise, the current configuration is a lattice local minimizer.

Fig. 2. Northby's pivot algorithm.

Moving an atom from one lattice site to another is called a *move* (or just a *pivot*) in the Northby algorithm. Each time an atom is added or removed from a site the program recalculates from VP the total potential V, and the energy change

$DV(i)$ associated with adding or removing another atom at each site. If the full Lennard–Jones potential function is used in the lattice search, we can easily deduce from the above description that the time complexity for choosing the most loosely bound atom is $O(n(n - N_{core}))$ and the time complexity for choosing the most tightly binding vacant site is $O(n(N_{surf} - (n - N_{core})))$. Therefore the time complexity for each move is $O(nN_{surf})$. Since $N_{surf} = O(n^{\frac{2}{3}})$ and that $N_{core} < n < N_{core} + N_{surf}$, the time complexity of each move in Northby's implementation is $O(n^{\frac{5}{3}})$.

Since it is likely that the true minimal configurations will have in some sense the largest possible number of nearest neighbors, Northby also used *nearest-neighbor* (NN) potential function defined below for lattice search.

$$\begin{aligned} v_{NN}(r) &= +100\,; \quad && r < 0.8\,, \\ &= -1\,; && 0.8 \leqslant r \leqslant 1.3,. \\ &= 0\,; && r > 1.3\,, \end{aligned} \tag{5}$$

If the NN potential function is used in the lattice search, the time complexity of each move in the Northby algorithm becomes $O(n^{\frac{2}{3}})$ for an n-atom cluster because the number of nearest neighbors of each atom is bounded by a constant.

3. Improvements on the Northby Algorithm

In [11], we have carefully studied Northby's algorithm. We note that the interaction matrix VP only speeds up the computation by a constant factor (of about 4) at the cost of $O(n^2)$ storage. Therefore, we have dropped the interaction matrix. Instead, we have introduced a double precision array $CORE(0{:}N_{surf})$ which stores the value of $\Sigma_{i,j\in I_{core}, i\neq j} VP(i, j)$ in $CORE(0)$ and the values of $\Sigma_{j\in I_{core}} VP(i, j)$ in $CORE(i)$ for each $i \in I_{surf}$. With the aid of this simple data structure, we have reduced the time complexity of each move to $O(n^{\frac{4}{3}})$.

In [24], we have further reduced the time complexity of each move to $O(n^{\frac{2}{3}})$ by introducing a new $O(n^{\frac{2}{3}})$ storage data structure. This new data structure is a double precision array $SURF(N_{surf})$. Given an initial configuration represented by I_{surf}^{filled}, the array $SURF$ is initialized in $O(n^{\frac{4}{3}})$ time so that $SURF(i) = \Sigma_{j\in I_{surf}^{filled}, j\neq i} VP(i, j)$ for each $i \in I_{surf}$. The value of $CORE(i) + SURF(i)$ is the contribution of the atom which is placed at the ith surface lattice point (if $i \in I_{surf}^{filled}$) or the amount that will be added to the total potential energy if a new atom is to be placed at site i of the surface lattice (if $i \in I_{surf}^{vacant}$). After this initialization is done, the most loosely bound atom can be found in $O(n^{\frac{2}{3}})$ time; if we delete i_{loose} from I_{surf}^{filled} and put it in I_{surf}^{vacant}, it requires $O(n^{\frac{2}{3}})$ time to update the array $SURF$; then the most tightly binding vacant site can be computed in $O(n^{\frac{2}{3}})$ time; to insert i_{tight} into I_{surf}^{filled} and update the array $SURF$ again takes $O(n^{\frac{2}{3}})$ time. Therefore the time complexity per move is reduced to $O(n^{\frac{2}{3}})$. The improved pivot algorithm is described as follows.

```
Algorithm 3: Modified Northby Pivot Algorithm
1. {Find the most loosely bound atom}
   Find $i_{loose} \in I_{surf}^{filled}$ such that
       $i_{loose} = arg \max_{i \in I_{surf}^{filled}} \{CORE(i) + SURF(i)\}$ .   (6)
   Let $gain_{loose}$ be the maximum function value that the maximization problem in (6) achieves at $i_{loose}$.
2. {pick up the most loosely bound atom}
   Drop $i_{loose}$ from $I_{surf}^{filled}$ and insert it into $I_{surf}^{vacant}$. Update the array $SURF$ in the following way: For
   each $i \in I_{surf}$ and $i \neq i_{loose}$, decrease $SURF(i)$ by $v(r_{i_{loose},i})$, where $v(\bullet)$ is the Lennard–Jones pair
   potential and $r_{i_{loose},i}$ is the Euclidean distance between site $i_{loose}$ and site $i$ on the surface lattice.
3. {Find the most tightly binding vacant site}
   Find $i_{tight} \in I_{surf}^{vacant}$ such that
       $i_{tight} = arg \min_{i \in I_{surf}^{vacant}} \{CORE(i) + SURF(i)\}$ .   (7)
   Let $gain_{tight}$ be the minimum function value that the minimization problem in (7) achieves at $i_{tight}$.
4. {put the atom at the most tightly binding vacant site}
   Drop $i_{tight}$ from $I_{surf}^{vacant}$ and insert it into $I_{surf}^{filled}$. Update the array $SURF$ in the following way: For
   each $i \in I_{surf}$ and $i \neq i_{tight}$, increase $SURF(i)$ by $v(r_{i_{tight},i})$.
5. {Check for stopping rule}
   If $gain_{tight} - gain_{loose} \geqslant 0$ then stop, this is a lattice local minimizer; otherwise, goto step 1.
```

Fig. 3. Modified Northby's pivot algorithm.

Now consider the case where the NN potential function is used in the lattice search. We may set up an AVL tree [9] for the values of $CORE(i) + SURF(i)$ for all the filled surface lattice points i and an AVL tree for the values of $CORE(i) + SURF(i)$ for all the vacant surface lattice points i. After the AVL trees are set up, the most loosely bound atom can be found in $O(\log n)$ time. If we delete i_{loose} from I_{surf}^{filled} and put it in I_{surf}^{vacant}, it requires constant time to update the array $SURF$ and only $O(\log n)$ time to update the AVL trees since the number of atoms which interact with this *picked up* atom is bounded by a constant. Then the most tightly binding vacant site can be found in $O(\log n)$ time. To insert i_{tight} into I_{surf}^{filled}, it takes constant time to update the array $SURF$ and $O(\log n)$ time to update the AVL trees. Therefore the time complexity per move is reduced to $O(\log n)$ for an n-atom cluster if the NN potential function is used in the lattice search. It should be noted that each time a new configuration is generated randomly, $O(n^{\frac{2}{3}})$ time is required to initialize the array $SURF$ and $O(n^{\frac{2}{3}} \log n)$ time to initialize the AVL trees.

The reduction of time complexity of each move discussed above makes the Northby algorithm for lattice search more efficient. In the following, we are going to modify Northby's strategy for relaxation of lattice local minimizers.

Out of curiosity, we have made a small change in the Northby algorithm by making a relaxation whenever a new lattice local minimizer with a better potential function value is found. With this modification, a lower energy configuration was found for $n = 134$. Northby reported −781.989 as the putative global minimum for $n = 134$, which was found by the relaxation of an IC lattice local minimizer (under the NN potential function). The new putative global minimum that we have found is −782.206 which was found by the relaxation of an IC lattice local minimizer. The configuration was found by relaxation of an IC lattice local minimizer with a LJ potential function value of −732.376. However, the best known lattice local minimizers have a value of −732.616, all of them relax to local minimizers with a potential function value of −781.476.

This fact indicates that *in some cases, the relaxation of a lattice local minimizer with a worse potential function value may lead to a local minimizer with a better potential function value.* Therefore we propose the following modified Northby algorithm for Molecular Conformation.

```
Algorithm 4: Modified Northby Algorithm for Molecular Conformation
1. Fix the base lattice to either the IC lattice or the FC lattice. Choose the lattice search potential
   function as either the full Lennard-Jones potential function or the nearest-neighbor potential
   function.
2. Repeat
     2.1. randomly generate a lattice configuration;
     2.2. pivot to a lattice local minimizer;
     2.3. relax this lattice local minimizer with the full LJ potential function;
   Until the lowest potential function value has been repeated for 250 times.
```

Fig. 4. Modified Northby algorithm for Molecular Conformation.

Algorithm 4 surely requires more time than Algorithm 1 because it performs more work. Therefore we have used Algorithm 4 *in addition to* Algorithm 1 and changed the stopping criterion in step 2 to 60 times instead of 250 times to save computer time. With Algorithm 4, *lower energy configurations* were found for $n = 65$, 66, 75, 76, 77 and 134. Computational results will be presented in the next section.

4. Computational Results

The algorithm was implemented on the Cray-XMP at the Minnesota Supercomputer Center (MSC) in Fortran 77 and compiled with the *cft*77 version 4.0 compiler in single-precision (64-bit). Relaxations are performed using the Truncated Newton code of S.G. Nash. For information on the Truncated Newton algorithm, readers are referred to [14,15].

For each problem instance (fix a cluster size, lattice type, and potential function), the algorithm never requires more than 3 minutes of CPU time on the Cray-xmp. Therefore the problems are solvable in interactive mode (3 minutes of CPU time). For this reason, timing issues are not reported in the tables.

Putative global minima are reported in Table I and Table II for $13 \leqslant n \leqslant 55$ and $56 \leqslant n \leqslant 147$, respectively. Lattice searches are divided into four categories: FC lattice NN potential, FC lattice LJ potential, IC lattice NN potential, and IC lattice LJ potential. In these tables, an x means that a putative global minimizer was produced by performing a lattice search in the corresponding category followed by relaxation; an o means that the lattice search in the corresponding category does not lead to a putative global minimizer; an. means that this category of lattice search does not exist (for FC lattices).

The differences in results obtained by Northby Algorithm and by Modified Northby Algorithm are presented in Table III. Note that for $n = 65$, 66, 75, 76, 77, and 134, lower energy configurations are found. For other values of n in Table III; the putative global minima are found in more categories.

The optimal configuration for the 76-atom cluster is plotted in Figure 5 in the appendix. It has a potential energy function value of -402.385. The sub-optimal configuration for 76-atom cluster with potential function value -402.177 is plotted in Figure 6 in the appendix. The difference in the two configurations is hard to tell. Therefore, we present their xyz coordinates in Tables IV and V in the appendix.

size	energy	FC lattice		IC lattice		size	energy	FC lattice		IC lattice	
n	V_n	NN	LJ	NN	LJ	n	V_n	NN	LJ	NN	LJ
13	-44.327	x	x	x	x	35	-155.757	o	o	x	x
14	-47.845	x	x	x	x	36	-161.825	o	o	x	x
15	-52.323	x	x	x	x	37	-167.034	o	o	x	o
16	-56.816	x	x	o	o	38	-173.134	o	o	x	x
17	-61.318	x	o	o	o	39	-180.033	o	o	x	x
18	-66.531	x	x	o	o	40	-185.250	o	o	x	x
19	-72.660	x	o	x	x	41	-190.536	o	o	x	x
20	-77.177	x	o	o	o	42	-196.278	o	o	x	x
21	-81.685	x	o	o	o	43	-202.365	o	o	x	x
22	-86.810	x	o	o	o	44	-207.689	o	o	x	x
23	-92.844	x	o	o	o	45	-213.785	o	o	x	x
24	-97.349	x	o	o	o	46	-220.680	.	.	x	x
25	-102.373	x	o	o	o	47	-226.012	.	.	x	x
26	-108.316	x	o	o	o	48	-232.200	.	.	x	x
27	-112.874	x	o	o	o	49	-239.092	.	.	x	x
28	-117.822	x	o	o	o	50	-244.550	.	.	x	x
29	-123.587	x	o	o	o	51	-251.254	.	.	x	x
30	-128.287	x	o	o	o	52	-258.230	.	.	x	x
31	-133.586	o	o	x	x	53	-265.203	.	.	x	x
32	-139.636	o	o	x	x	54	-272.209	.	.	x	x
33	-144.843	o	o	x	x	55	-279.248	.	.	x	x
34	-150.045	o	o	x	x						

Table 1: Putative global minima for $13 \leq n \leq 55$.

size	energy	FC lattice		IC lattice		size	energy	FC lattice		IC lattice	
n	V_n	NN	LJ	NN	LJ	n	V_n	NN	LJ	NN	LJ
56	-283.643	o	o	x	x	102	-569.278	o	o	x	x
57	-288.343	x	x	o	o	103	-575.659	o	o	x	x
58	-294.378	x	x	o	o	104	-582.038	o	o	x	x
59	-299.738	x	o	o	o	105	-588.267	o	o	x	x
60	-305.875	x	x	o	o	106	-595.061	o	o	x	x
61	-312.009	x	x	o	o	107	-601.912	o	o	x	x
62	-317.354	x	o	o	o	108	-609.033	o	o	x	x
63	-323.490	x	x	o	o	109	-615.411	o	o	x	x
64	-329.620	x	x	o	o	110	-621.788	o	o	x	x
65	-334.972	x	o	o	o	111	-628.068	o	o	x	x
66	-341.111	x	x	o	o	112	-634.875	o	o	x	x
67	-347.252	x	x	o	o	113	-641.700	o	o	x	x
68	-353.395	x	o	o	o	114	-648.833	o	o	x	x
69	-359.726	x	o	o	o	115	-655.636	o	o	x	x
70	-366.892	x	x	o	o	116	-662.809	o	o	x	x
71	-373.350	x	o	o	o	117	-668.283	o	o	x	x
72	-378.524	x	o	o	o	118	-674.770	o	o	x	x
73	-384.789	x	o	o	o	119	-681.419	o	o	x	x
74	-390.908	x	o	o	o	120	-687.022	o	o	x	x
75	-396.239	x	o	o	o	121	-693.820	o	o	x	x
76	-402.385	x	o	o	o	122	-700.939	o	o	x	x
77	-408.518	x	o	o	o	123	-707.802	o	o	x	x
78	-414.681	o	o	x	x	124	-714.921	o	o	x	x
79	-421.811	o	x	o	o	125	-721.303	o	o	x	x
80	-428.084	x	o	o	o	126	-727.350	o	o	x	x
81	-434.344	x	o	o	o	127	-734.480	.	.	x	x
82	-440.550	o	o	x	x	128	-741.332	.	.	x	x
83	-446.924	o	o	x	x	129	-748.461	.	.	x	x
84	-452.657	o	o	x	x	130	-755.271	.	.	x	x
85	-459.056	o	x	o	o	131	-762.442	.	.	x	x
86	-465.384	o	o	x	x	132	-768.042	.	.	x	o
87	-472.098	o	o	x	x	133	-775.023	.	.	o	x
88	-478.935	o	o	x	x	134	-782.206	.	.	o	x
89	-486.054	o	o	x	x	135	-790.278	.	.	o	x
90	-492.434	o	o	x	x	136	-797.453	.	.	o	x
91	-498.811	o	o	x	x	137	-804.631	.	.	o	x
92	-505.185	o	o	x	x	138	-811.813	.	.	o	x
93	-510.878	o	o	x	x	139	-818.994	.	.	x	x
94	-517.264	o	o	x	x	140	-826.175	.	.	x	x
95	-523.640	o	o	x	x	141	-833.359	.	.	x	x
96	-529.879	o	o	x	x	142	-840.539	.	.	x	x
97	-536.681	o	o	x	x	143	-847.722	.	.	x	x
98	-543.547	o	o	x	x	144	-854.904	.	.	x	x
99	-550.667	o	o	x	x	145	-862.087	.	.	x	x
100	-557.040	o	o	x	x	146	-869.273	.	.	x	x
101	-563.411	o	o	x	x	147	-876.461	.	.	x	x

Table 2: **Putative global minima for** $56 \leq n \leq 147$**.**

Northby Algorithm						Modified Northby Algorithm					
size	energy	FC lattice		IC lattice		size	energy	FC lattice		IC lattice	
n	V_n	NN	LJ	NN	LJ	n	V_n	NN	LJ	NN	LJ
14	-47.845	x	.	.	o	14	-47.845	x	x	x	x
15	-52.323	x	.	.	o	15	-52.323	x	x	x	x
19	-72.660	x	.	.	o	19	-72.660	x	o	x	x
50	-244.550	.	.	x	o	50	-244.550	.	.	x	x
65	-334.915	o	x	o	o	65	-334.972	x	o	o	o
66	-341.043	o	x	o	o	66	-341.111	x	x	o	o
67	-347.252	o	x	o	o	67	-347.252	x	x	o	o
75	-396.037	o	x	o	o	75	-396.239	x	o	o	o
76	-402.177	o	x	o	o	76	-402.385	x	o	o	o
77	-408.463	o	o	x	x	77	-408.518	x	o	o	o
134	-781.989	.	.	x	o	134	-782.206	.	.	o	x
139	-818.994	.	.	o	x	139	-818.994	.	.	x	x

Table 3: **Difference in results from different algorithms.**

5. Conclusions

In this paper, we have introduced simple data structures into the Northby algorithm for lattice search which greatly reduce the time complexity of the original algorithm. With this improvement, we were able to compute the ground states of Lennard–Jones clusters much faster. We have also modified the Northby algorithm for Molecular Conformation by relaxing every lattice local minimizer found in the process. With this improved algorithm, *lower energy configurations* were found for $n = 65$, 66, 75, 76, 77 and 134. These findings also show that *in some cases, the relaxation of a lattice local minimizer with a worse potential function value may lead to a local minimizer with a better potential function value.* Since these cases are less than 5% of the total cases, we still believe that in *most of the cases, the relaxation of a global lattice minimizer will results in a configuration with a lower energy than the relaxation of a non-global lattice local minimizer.*

It should be noted that the search for a lattice global minimizer can be formulated as a discrete optimization problem. We need to assign n atoms to $m \geqslant n$ lattice points. Corresponding to each assignment, there is a potential energy function value. The lattice global minimizer corresponds to the assignment which has the lowest potential energy function value. This problem could be solved by existing techniques for quadratic assignment problems. It can also be viewed as a set partitioning problem and solved using techniques for set partitioning.

Acknowledgement

This research was supported in part by the Army Research Office contract number DAAL03-89-C-0038 with the University of Minnesota Army High Performance Computing Research Center. I would like to thank C. Floudas from Princeton University for pointing out several printing mistakes in an earlier version of this paper. Thanks are also due to Jorge Moré from Argonne National Laboratory and Panos Pardalos from University of Florida for their encouragement.

Appendix

A. THE CONSTRUCTION OF IC AND FC LATTICES

The icosahedral lattice [3,7,16] introduced by Mackay can be described as 20 slightly flattened tetrahedrally shaped fcc units with 12 vertices on a sphere centered at the origin. The ratio between the interatomic spacing in the 20 equilateral outer faces and the radial lines connecting the 12 vertices with the origin is $\sqrt{\dfrac{2}{1+\cos\left(\dfrac{\pi}{5}\right)}}$, which is approximately 1.05146.

For the *IC* lattice, the total number of the lattice on each layer is 1, 12, 42, 92, . . . , $10i^2 + 2$, ... Therefore the number of lattice points in the sequence of closed shell *IC* lattice is 1, 13, 55, 147, . . . , $1 + (10i^3 + 15i^2 + 11i)/3$, ...

The *FC* lattice consists of a smaller *IC* lattice enclosed by a layer of *stacking fault* icosahedral shell. This shell has 12 vertices and 20 facets as described above. However, it has fewer filling lattice points on each facet. These lattice points are located at the stacking fault positions of the *IC* lattice shell. The number of lattice points on the outer layer of an *FC* lattice is 1, 12, 32, 72, 132, . . . , $10i(i-1) + 12$, ... Therefore the number of lattice points in the sequence of closed shell *FC* lattice is 1, 13, 45, 127, . . . , $11 + (10i^3 + 15i^2 - 19i)/3$, ...

Figure 1 of [16] best describes how each of the facets are filled with other lattice points for both the *IC* shell and the *FC* shell. A short fortran subroutine for constructing the icosahedral lattice is attached in the following.

```
      subroutine icgen(n, x, y, z)
      integer n
      double precision x(n), y(n), z(n)
c**********
c     This subroutine generates the coordinates for an icosahedron
c     lattice with n points
c
c     n is an integer INPUT variable
c       on entry, n is the number of points in an icosahedron lattice
c       on exit, n is not changed
c
c     x, y, z are double precision OUTPUT arrays
c       on entry, they need not be specified
c       on exit, they contain the x-y-z coordinates of the lattice
c**********
      integer i, j, k, point, oldpoint, newpoint, ir
      double precision cccc, ssss, angle, cosalpha, sinalpha
      double precision cc, ss, r, x1, y1, z1, x2, y2, z2, x3, y3, z3

      angle = -2.0*3.1415926/5.0
      cccc = cos(angle)
      ssss = sin(angle)
      cc = cos(angle/2.0)
      ss = sin(angle/2.0)
      cosalpha =  cc/(1.0 + cc)
      sinalpha =  sqrt(1.0 - cosalpha*cosalpha)
      point = 1
      x(1) = 0.0
      y(1) = 0.0
      z(1) = 0.0
      oldpoint = 1

      r = 1.0
00010 ir = int(r)
      x1 = 0.0
      y1 = r
      z1 = 0.0
      x2 = r*sinalpha
      y2 = r*cosalpha
      z2 = 0.0
      x3 = cccc*x2 + ssss*z2
      y3 = y2
      z3 =-ssss*x2 + cccc*z2
      do 00030, i = 0, ir-1
        do 00020, j = 1, ir-i
          point = point + 1
          x(point) = real(i)/r*x1 + real(j)/r*x2 + real(r-i-j)/r*x3
          y(point) = real(i)/r*y1 + real(j)/r*y2 + real(r-i-j)/r*y3
          z(point) = real(i)/r*z1 + real(j)/r*z2 + real(r-i-j)/r*z3
00020   continue
00030 continue

      x1 = cc*x2 + ss*z2
      y1 =-y2
      z1 =-ss*x2 + cc*z2
      do 00050, i = 1, ir
        do 00040, j = 0, ir-i
          point = point + 1
          x(point) = real(i)/r*x1 + real(j)/r*x2 + real(r-i-j)/r*x3
          y(point) = real(i)/r*y1 + real(j)/r*y2 + real(r-i-j)/r*y3
          z(point) = real(i)/r*z1 + real(j)/r*z2 + real(r-i-j)/r*z3
00040   continue
00050 continue

      x2 = cccc*x1 + ssss*z1
      y2 = y1
      z2 =-ssss*x1 + cccc*z1
      do 00070, i = 1, ir-1
        do 00060, j = 1, ir-i
          point = point + 1
          x(point) = real(i)/r*x1 + real(j)/r*x2 + real(r-i-j)/r*x3
          y(point) = real(i)/r*y1 + real(j)/r*y2 + real(r-i-j)/r*y3
          z(point) = real(i)/r*z1 + real(j)/r*z2 + real(r-i-j)/r*z3
00060   continue
00070 continue

      x3 = 0.0
      y3 =-r
      z3 = 0.0
      do 00090, i = 1, ir-1
        do 00080, j = 0, ir-i-1
          point = point + 1
          x(point) = real(i)/r*x1 + real(j)/r*x2 + real(r-i-j)/r*x3
          y(point) = real(i)/r*y1 + real(j)/r*y2 + real(r-i-j)/r*y3
          z(point) = real(i)/r*z1 + real(j)/r*z2 + real(r-i-j)/r*z3
00080   continue
00090 continue
      newpoint = point

      do 00110, k = 1, 4
        do 00100, i = oldpoint + 1, newpoint
          point = point + 1
          x(point) = cos(k*angle)*x(i) + sin(k*angle)*z(i)
          y(point) = y(i)
          z(point) =-sin(k*angle)*x(i) + cos(k*angle)*z(i)
00100   continue
00110 continue
      point = point + 1
      x(point) = 0.0
      y(point) = r
      z(point) = 0.0
      point = point + 1
      x(point) = 0.0
      y(point) = -r
      z(point) = 0.0
      oldpoint = point
      if (oldpoint .lt. n) then
        r = r + 1.0
        goto 00010
      endif
99999 return
      end
```

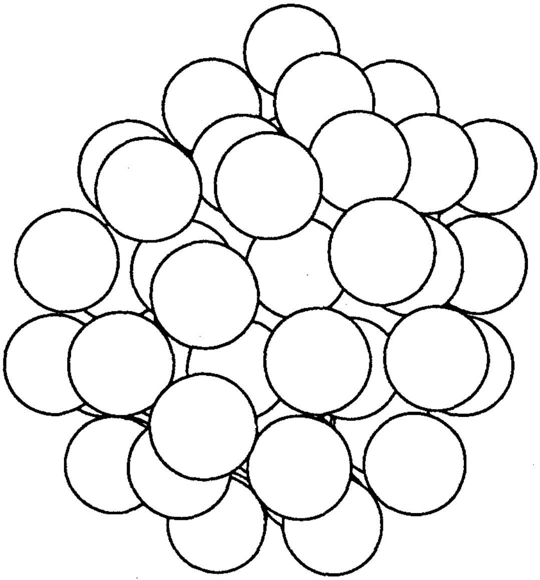

Fig. 5. New result for the 76-atom cluster, energy = −402.385.

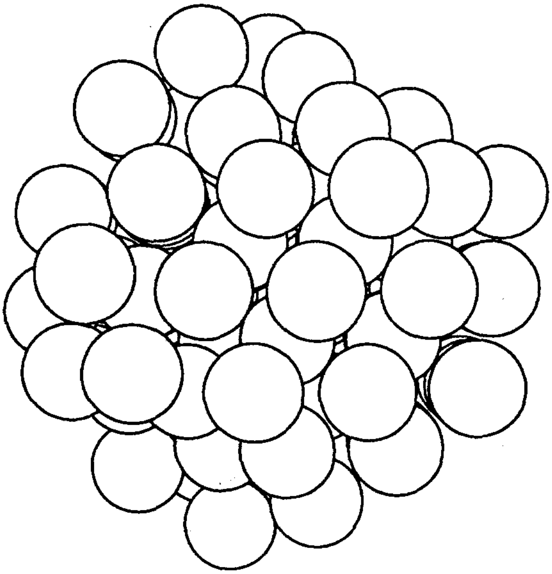

Fig. 6. Northby's result for the 76-atom cluster, energy = −402.177.

B. CONFIGURATIONS FOR THE 76-ATOM CLUSTER

In this section, we provide information for the configurations for the 76-atom cluster. Both the coordinates and the plots are provided here. Note that the two configurations are not the same and that one can not be obtained by performing reflection or rotation. They have different potential energy function values.

i	x_i	y_i	z_i	i	x_i	y_i	z_i
1	0.002	-0.001	0.002	39	-1.389	0.845	-0.981
2	0.838	0.408	0.013	40	-0.702	1.382	-0.482
3	0.680	-0.423	0.502	41	-0.013	-0.020	-1.621
4	0.256	0.409	0.807	42	-0.976	-0.014	-1.300
5	-0.255	-0.418	0.778	43	-0.557	-0.870	-1.591
6	-0.680	0.411	0.502	44	0.408	-0.889	-1.304
7	-0.839	-0.417	0.005	45	-0.275	-1.383	-0.766
8	-0.684	0.414	-0.479	46	1.121	0.844	-0.804
9	-0.268	-0.426	-0.783	47	0.519	0.838	-1.612
10	0.255	0.410	-0.789	48	0.261	1.379	-0.800
11	0.673	-0.434	-0.485	49	1.548	-0.020	-0.490
12	-0.003	0.930	0.016	50	0.949	-0.021	-1.314
13	0.017	-0.937	0.016	51	1.365	-0.876	-0.992
14	1.119	0.839	0.842	52	1.379	-0.879	0.017
15	1.702	0.836	0.019	53	0.688	-1.414	-0.494
16	0.856	1.377	0.023	54	-0.004	1.895	0.025
17	0.946	-0.020	1.346	55	0.021	-1.903	0.043
18	1.552	-0.016	0.527	56	-0.741	-2.211	-0.468
19	1.365	-0.866	1.030	57	-0.225	0.614	2.299
20	0.407	-0.857	1.311	58	-1.013	-1.651	1.400
21	0.697	-1.401	0.533	59	-1.889	0.071	1.463
22	-0.439	0.831	1.346	60	-1.600	-1.660	0.578
23	0.509	0.821	1.648	61	-1.008	-1.720	-1.292
24	0.257	1.373	0.845	62	-1.181	0.615	1.985
25	-0.947	-0.027	1.310	63	-0.053	-1.708	-1.656
26	-0.014	-0.027	1.616	64	0.127	-0.773	2.268
27	-0.518	-0.843	1.567	65	-1.451	-0.777	1.746
28	-1.106	-0.859	0.813	66	1.084	-0.777	1.987
29	-0.262	-1.377	0.820	67	-0.786	-1.276	2.359
30	-1.399	0.845	0.024	68	-1.593	-1.708	-0.464
31	-1.384	0.823	1.030	69	-2.051	-0.786	0.950
32	-0.701	1.375	0.533	70	-0.035	-1.644	1.754
33	-1.564	-0.009	-0.484	71	0.231	-2.187	0.965
34	-1.557	-0.018	0.527	72	0.921	-1.653	1.471
35	-1.706	-0.855	0.020	73	0.202	-2.225	-0.858
36	-1.137	-0.876	-0.789	74	-0.747	-0.237	2.267
37	-0.837	-1.374	0.030	75	0.653	0.074	2.299
38	-0.439	0.848	-1.308	76	-0.751	-2.183	0.579

Table IV. Optimal configuration for 76-atom cluster, energy = −402.385.

i	x_i	y_i	z_i	i	x_i	y_i	z_i
1	-0.003	-0.001	0.002	39	-1.383	0.816	-0.982
2	0.827	0.423	-0.004	40	-0.713	1.353	-0.507
3	0.662	-0.422	0.493	41	-0.018	0.011	-1.641
4	0.248	0.423	0.795	42	-0.961	-0.002	-1.300
5	-0.279	-0.414	0.795	43	-0.521	-0.835	-1.631
6	-0.676	0.428	0.491	44	0.424	-0.852	-1.333
7	-0.837	-0.419	0.002	45	-0.281	-1.376	-0.835
8	-0.679	0.402	-0.482	46	1.104	0.868	-0.826
9	-0.254	-0.414	-0.801	47	0.495	0.859	-1.627
10	0.249	0.421	-0.798	48	0.234	1.391	-0.829
11	0.669	-0.418	-0.495	49	1.540	0.005	-0.510
12	-0.010	0.937	-0.006	50	0.943	0.007	-1.328
13	-0.017	-0.937	-0.007	51	1.368	-0.849	-1.005
14	1.109	0.867	0.811	52	1.372	-0.862	-0.001
15	1.689	0.859	-0.007	53	0.673	-1.389	-0.514
16	0.839	1.398	-0.014	54	-0.029	1.906	-0.020
17	0.940	0.000	1.322	55	-0.039	-1.904	-0.022
18	1.536	0.000	0.503	56	-2.087	-0.713	-0.891
19	1.354	-0.858	1.004	57	-1.930	0.136	-1.392
20	0.398	-0.859	1.321	58	-1.669	-1.606	-0.611
21	0.666	-1.395	0.500	59	-0.284	0.753	-2.244
22	-0.442	0.875	1.316	60	-0.992	2.168	-0.074
23	0.512	0.860	1.616	61	-2.376	-0.137	-0.009
24	0.246	1.398	0.804	62	-2.218	0.711	-0.511
25	-0.982	0.022	1.309	63	-1.980	0.185	1.332
26	-0.026	0.008	1.629	64	-1.574	-0.667	1.666
27	-0.574	-0.838	1.614	65	-1.687	1.635	0.441
28	-1.150	-0.849	0.806	66	-1.085	-1.600	-1.406
29	-0.299	-1.384	0.807	67	-0.810	-0.116	-2.244
30	-1.383	0.841	-0.008	68	-2.212	0.722	0.489
31	-1.377	0.874	0.989	69	-0.122	1.633	-1.728
32	-0.713	1.398	0.487	70	-1.669	1.620	-0.549
33	-1.546	-0.008	-0.498	71	-1.221	0.726	-1.921
34	-1.537	0.011	0.486	72	-0.831	-2.151	-0.610
35	-1.704	-0.846	-0.003	73	-0.389	2.166	-0.912
36	-1.113	-0.835	-0.811	74	-2.144	-0.657	0.845
37	-0.884	-1.382	-0.011	75	-1.058	1.617	-1.398
38	-0.455	0.838	-1.299	76	-1.495	-0.709	-1.704

Table V. Sub-optimal configuration for 76-atom cluster, energy = −402.177

References

1. J. P. Brunet, A. Edelman, and J. P. Mesirov (1990), An Optimal Hypercube Direct N-body Solver on the Connection Machine, Proceedings of Superconducting '90, pp. 748–752, IEEE Computer Society Press.
2. R. H. Byrd, E. Eskow, R. B. Schnabel, and S. L. Smith (1991), Parallel Global Optimization: Numerical Methods, Dynamic Scheduling Methods, and Application to Molecular Configuration,

Technical Report CU-CS-553-91, University of Colorado at Boulder, Department of Computer Science, Boulder, CO.

3. J. H. Conway and N. J. A. Sloane (1988), *Sphere Packings, Lattices and Groups*, Springer-Verlag.
4. J. Farges, M. F. De Feraudy, B. Raoult, and G. Torchet (1985), Cluster Models Made of Double Icosahedron Units, *Surface Science* **156**, 370–378.
5. I. Z. Fisher, (1964), *Statistical Theory of Liquids*, Chicago and London.
6. D. G. Garrett, K. D. Kastella, and D. M. Ferguson, New Results on Protein Folding from Simulated Annealing, submitted to *Journal of the American Chemistry Society*, 1992.
7. M. R. Hoare (1979), Structure and Dynamics of Simple Microclusters, *Advances in Chemical Physics* **40**, 49–135.
8. J. Danna Honeycutt and Hans C. Andersen (1987), Molecular Dynamics Study of Melting and Freezing of Small Lennard-Jones Clusters, *Journal of Physical Chemistry* **91**, 4950–4963.
9. E. Horowitz and S. Sahni (1990), *Data Structures in Pascal*, Computer Science Press.
10. R. S. Judson, M. E. Colvin, J. C. Meza, A. Huffer, and D. Gutierrez (1991), Do Intelligent Configuration Search Techniques Outperform Random Search for Large Molecules?, Sandia Report SAND91-8740, Sandia National Laboratories, Center for Computational Engineering, Livermore, CA.
11. R. S. Maier, J. B. Rosen, and G. L. Xue (1992), A Discrete-Continuous Algorithm for Molecular Energy Minimization, in Proceedings of the 1992 IEEE/ACM Supercomputing Conference, Minneapolis, MN, November 16–20, 1992, pp. 778–786. Also available as Army High Performance Computing Research Center Preprint 92-31, University of Minnesota, Minneapolis.
12. C. D. Maranas and C. A. Floudas (1992), Global Optimization of Lennard–Jones Microclusters, *Journal of Chemical Physics* **97**(10), 7667.
13. C. D. Maranas and C. A. Floudas (1993), Global Optimization for Molecular Conformation Problems, *Annals of Operations Research*, in press (1993).
14. S. G. Nash (1984), User's Guide for TN/TNBC: Fortron Routines for Nonlinear Optimization, Report 397, Mathematical Sciences Department, The Johns Hopkins University.
15. S. G. Nash (1985), Preconditioning of Truncated-Newton Methods, *SIAM Journal on Scientific and Statistic Computing* **6**, 599–616.
16. J. A. Northby (1987), Structure and Binding of Lennard–Jones Clusters: $13 \leqslant n \leqslant 147$, *Journal of Chemical Physics* **87**, 6166–6178.
17. D. R. Ripoll and S. J. Thomas (1990), A Parallel Monte Carlo Search Algorithm for the Conformational Analysis of Proteins, Proceedings ACM/IEEE Superconducting '90, pp. 94–102.
18. T. Schlick and M. Overton (1987), A Powerful Truncated Newton Method for Potential Energy Minimization, *Journal of Computational Chemistry* **8**, 1025–1039.
19. David Shalloway (1991), *Packet Annealing: A Deterministic Method for Global Minimization*, Application to Molecular Conformation, Recent Advances in Global Optimization, C. Floudas and P. Pardalos, eds. Princeton University Press: Princeton, NJ.
20. D. G. Vlachlos, L. D. Schmidt, and R. Aris (1991), Structures of Small Metal Clusters: Phase Transitions and Isomerization, Army High Performance Computing Research Center Preprint 91-69, University of Minnesota, Minneapolis.
21. D. G. Vlachos, L. D. Schmidt, and R. Aris (1991), Structures of Small Metal Clusters: Low Temperature Behavior, Army High Performance Computing Research Center Preprint 91-70, University of Minnesota, Minneapolis.
22. L. T. Wille (1987), Minimum-Energy Configurations of Atomic Clusters: New Results Obtained by Simulated Annealing, *Chemical Physics Letters* **133**, 405–410.
23. G. L. Xue, R. S. Maier, and J. B. Rosen (1991), Minimizing the Lennard–Jones Potential Function on a Massively Parallel Computer, in Proceedings of the 1992 ACM International Conference on Supercomputing, Washington, DC., July 19–23, 1992, pp. 409–416. Also available as Preprint 91-115, Army High Performance Computing Research Center, University of Minnesota, Minneapolis, Decemter 1991.
24. G. L. Xue (1992), Parallel Two-Level Simulated Annealing: Applications to Molecular Conformation, Preprint 92-047, Army High Performance Computing Research Center, University of Minnesota, Minneapolis, April 1992.