An Unbiased Population-Based Search for the Geometry Optimization of Lennard–Jones Clusters: 2 ≤ N ≤ 372

WAYNE PULLAN

School of Information and Communication Technology, Griffith University, Gold Coast, Qld., 4215, Australia

Received 24 October 2004; Accepted 3 February 2005
DOI 10.1002/jcc.20226
Published online in Wiley InterScience (www.interscience.wiley.com).

Abstract: This article presents the results obtained using an unbiased Population Based Search (PBS) for optimizing Lennard–Jones clusters. PBS is able to repeatedly obtain all putative global minima, for Lennard–Jones clusters in the range 2 ≤ N ≤ 372, as reported in the Cambridge Cluster Database. The PBS algorithm incorporates and extends key techniques that have been developed in other Lennard–Jones optimization algorithms over the last decade. Of particular importance are the use of cut-and-paste operators, structure niching (using the cluster strain energy as a metric), two-phase local search, and a new operator, Directed Optimization, which extends the previous concept of directed mutation. In addition, PBS is able to operate in a parallel mode for optimizing larger clusters.


Key words: global geometry optimization; clusters; Lennard–Jones

Introduction

Atomic clusters are currently an active field of theoretical and experimental research. Atomic clusters are aggregates of atoms, sufficiently small so that a significant proportion of the atoms are present on the surface of the cluster. Theoretical investigations of atomic clusters address the following optimization problem:

Given N particles, interacting with two-body central forces, find the configuration in three-dimensional Euclidean space for which the total potential energy attains its global minimum.

The two-body central force used in this study is the Lennard–Jones pair-potential, which is defined as follows: Given $r$, the distance between two atoms, their interaction energy is

$$v(r) = 4\epsilon \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6$$

(1)

where $\epsilon = \sigma = 1$. The form of $v(r)$ is shown in Figure 1, where the minimum for $v(r)$ occurs at approximately $r = 1.123$. The total energy $E$ of a cluster of $N$ atoms is simply given by the sum of the pair interactions between all atom pairs, i.e.,

$$E(X) = E(X_1, \ldots, X_N) = \sum_{i<j} v(\|X_i - X_j\|),$$

where $X_i \in \mathbb{R}^3$ are the coordinates of the center of the $i$th atom and the norm is the Euclidean one. The minimization problem is the following:

$$\min_{X \in \mathbb{R}^{3N}} E(X).$$

It is important to remark here that we are interested in the global minimum for their problem. Indeed, local minima for this problem can be detected quite efficiently, but it is conjectured that their number increases exponentially with the number $N$ of atoms. This makes the global minimization problem an extremely difficult one that has been extensively studied and has many applications. For example, it is employed in the analysis of the three-dimensional conformation of clusters of inert atoms. Moreover, although the Lennard–Jones potential is not a good choice when modeling metals, in some cases (e.g., gold or nickel) there is a strong preference for the formation of 75-atom clusters with the same structure as the optimal Lennard–Jones cluster of that size. In addition, the Lennard–Jones pair-potential is an important component of the nonbonded pair interactions in complex molecular systems such as proteins. In studying the three-dimensional conformation of proteins it is recognized that this interaction, together with the electrostatic, or Coulomb, interaction, is the most relevant one. Therefore, developing efficient methods for the minimization

Correspondence to: W. Pullan; e-mail: w.pullan@griffith.edu.au
of the Lennard–Jones clusters could be an important step towards developing efficient methods for the protein folding problem.

This article is structured as follows: in Section 2, an overview of previous Lennard–Jones cluster optimization methods is presented, particularly those relevant to the Population Based Search (PBS) algorithm described in Section 3. The results achieved by PBS are shown in Section 4 while Section 5 contains a conclusion and possible future research directions.

**Historical Overview**

Prior to 1987, the most extensive study of atomic clusters was the work of Hoare et al. who developed a general growth algorithm and used it to generate large numbers of stable structures, mainly for \( N \leq 55 \). These were compared to find the lowest energy structures, which in turn, became candidates for the absolute minimal structures. Hoare and Pal observed that, while what they termed as the “icosahedral growth sequence” did not, in general, produce minimal structures, icosahedral subunits did appear regularly in relaxed configurations generated by other sequences. The icosahedral lattice can be described as 20 slightly flattened tetrahedrally shaped face-centered cubic units with 12 vertices on a sphere centered at the origin. For the icosahedral lattice, the total number of lattice points on each layer is 1, 12, 42, 92, 162, \ldots. Therefore, the number of lattice points in the sequence of closed shell icosahedral lattices is 1, 13, 55, 147, 309, 561, \ldots.

The observation by Hoare and Pal led to the so-called lattice search methods, where the search for optimal clusters was performed over an icosahedral lattice. Using these techniques, many putative global minima for \( N \leq 147 \) were discovered for the first time. Subsequently, lattice search methods have produced the putative global minima for \( 148 \leq N \leq 309, 310 \leq N \leq 561, \) and \( 562 \leq N \leq 1000 \). However, as lattice based search methods are biased (i.e., the search is restricted to only a portion of the feasible domain, namely that portion containing the predefined lattice structures), they do not explore other regions that have, in some cases, contained lower energy clusters with other types of structure.

Over the last decade a number of optimization methods have evolved for the unbiased, systematic optimization of Lennard–Jones clusters. These can be broadly grouped into Nonpopulation and Population-based categories and are now discussed in more detail.

**Unbiased Nonpopulation-Based Search**

With regard to unbiased, nonpopulation-based search methods, one of the most powerful is the Basin-Hopping (BH) method, which exploits the funnel structure of the energy landscape. A funnel can be defined as a set of local minima such that, for all of them, at least one decreasing sequence of “neighbor” local minima exists leading to a unique local minimum corresponding to the bottom of the funnel. The key observation was that, while the Lennard–Jones clusters have a large number of local energy minima (conjectured to be exponentially increasing with the number \( N \) of atoms), the number of funnels is usually very limited (in some easy cases, such as \( N = 13 \), there even exists a single funnel) and, for one of these funnels, the bottom is always the global minimum of the energy function. This observation led to the introduction of the BH algorithm, which is able to reach the bottom of a funnel and was the first unbiased algorithm to detect all putative global minima, as currently reported.

BH randomly explores local minima returned by local searches started within a neighborhood of the current local minimum until it decides, according to a Metropolis acceptance criterion, to move to one of those local minima or to stop because a prefixed number of iterations with no improvement has been reached. Because there is no guarantee that a single run of BH is able to detect the global minimum (at least within a reasonable amount of time), BH is usually employed in a multistart fashion, that is, it is run many times from different, randomly sampled, starting points. Despite its simplicity, the successes of BH were impressive and, along with its monotonic variant, Monotonic Basin Hopping (MBH), BH was able to detect a new optimal structure for the case of \( N = 98 \) atoms, the Leary tetrahedron.

Recently, the BH method has been extended by the incorporation of two-phase local searches. In these local searches, the local minimization of the original energy function \( E \) is preceded by the local minimization of a modified function \( F = E + g \) where function \( g \) is a parameterized geometric penalization term that allows different geometrical shapes to be favored. In principle, local searches could be substituted by two-phase local searches in all algorithms. When incorporated into MBH, two-phase local searches considerably improve the efficiency of detecting the most challenging Lennard–Jones global minima, namely those having a nonicosahedral structure (the FCC structure at \( N = 38 \), the decahedral structure at \( N = 75–77, 102–104 \), and the already mentioned Leary tetrahedron at \( N = 98 \)).

**Unbiased Population-Based Search**

In population-based approaches, for example, efficiency is increased by keeping, at each iteration, not a single local minimum.
as in BH but a population of local minima in such a way that all of them “sufficiently” differ from each other. Basically, diversification is forced between members of the population and many trajectories are followed at the same time, which often improves the performance with respect to methods, such as BH, which only follow a single trajectory at a time and can only force diversification through random restarting.

Over the last decade, a number of successful, unbiased, population-based, search methods have evolved for the systematic optimization of Lennard–Jones clusters. These include the genetic algorithm of Deaven et al., which introduced the concept of phenotype cut-and-paste operators for crossover and mutation. This algorithm was able to find most putative global minima up to $N = 110$ but failed for some cluster sizes where the optimal structure is nonicosahedral [the Face Centered Cubic (FCC) structure at $N = 38$, the decahedral structures at $N = 75–77$, the Leary tetrahedron at $N = 98$ and the decahedral structures at $N = 102–104$]. Subsequently, Hartke developed a phenotype population-based algorithm introducing the concepts of structure niching and directed mutation. This algorithm was able to find, with the exception of $N = 98$, all putative global minima in the range $2 \leq N \leq 150$. More recently, Lee et al. developed the Conformational Space Annealing (CSA) method, which is able to find all global minimum up to $N = 201$, but is only partially successful for $N = 184, 188–192, 198, 199$. While CSA has a different population management technique and uses a different structure metric than that of Hartke, it also utilizes the techniques of structure niching and directed mutation.

Within the context of unbiased population-based search, the techniques of cut-and-paste operators, structure niching, and directed mutation seem to be keys to the improved performance of these algorithms, and they are now discussed in more detail:

1. **Cut-and-Paste (Phenotype) Operators**: In BH, the generation of a new local minimum is simply obtained by starting a local search within the neighborhood of the current local minimum. In the framework of population-based approaches, a number of new operators have been defined that generate a new cluster by modifying a cluster (unary operators) or by combining two clusters (binary operators). As these operators function directly on the geometric model of the cluster, they have a greater probability of retaining good “building blocks” within the cluster when compared to operators that function on the genotype representation of the cluster.

2. **Structure Niching** is a population diversification technique implemented through dissimilarity metrics, which measure either the relative difference between clusters, or produce an absolute measure of cluster structure. These structure metrics allow structure niche groups to be maintained within the population. During population updating, the energy value of a cluster to be added to the population is only compared with the energy values of the clusters within the same structure group. This ensures that nonicosahedral structures are not eliminated from the population during the search by the more prevalent and, initially lower energy, icosahedral structures.

3. **Directed Mutation**: It has been observed that the performance of BH degrades as the number $N$ of atoms increases (private communication). It can be reasonably claimed that one of the main reasons for this lies in the mechanism that generates new candidate local minima in the neighborhood of the current local minimum. In BH, this is simply obtained by a random perturbation of all the coordinates of the current local minimum. This mechanism often leads quickly to a point close to the global minimum, and is also quite general (i.e., it can be extended to global optimization problems that are different from molecular conformation ones). However, it often happens that a better local minimum than the current one exists but is only slightly different from the current one. Such an improvement is difficult to detect by perturbing all the atoms in a cluster (thus disrupting the whole structure of the cluster), and this difficulty increases with the number $N$ of atoms. Therefore, the key to improving the performance is to find other, more structured, perturbations of atoms in addition to using random perturbations. Hartke implemented such a directed mutation technique by moving the “worst” atom to the “best” vacant position and observed that if directed mutation is employed “the resulting overall speedup can be so large that it makes all the difference between an efficient solution and impractically long computation times.” In addition, similar angular moves of the most weakly bound surface atoms were implemented in ref. 10 and also within the GMIN computer program.

A further inherent benefit of population-based algorithms is that they are often straightforward to parallelize with the obvious, but important consequence, of a considerable reduction in elapsed times for optimizing clusters.

**Population-Based Search Algorithm**

The Population-Based Search (PBS) presented in this article is tailored to cluster optimization problems and utilizes several of the key techniques described earlier. PBS is able to efficiently utilize any number of computer processors to optimize clusters. A Master task controls some number of subtasks using a simple MPI send–receive message/command interface. The Master task manages the population and allocates work to subtasks that either generate a new member for the population, mutate an existing member, or perform a crossover of two existing members. There is no concept of “generations,” and the only task synchronization point occurs at the end of the population creation phase.

PBS is a two-phase algorithm with an initial population creation phase (3.2), which parallels the Basin–Hopping method in that it randomly generates starting trial clusters and incorporates mechanisms for following the energy funnel within which the cluster lies. The primary goal of the population creation phase is to create an initial population that contains a range of structures, all with energies reasonably close to the global minimum. This population creation phase is followed by a search phase that starts with the initial population and, using crossover and mutation operators followed by local searches, iteratively updates the population.

The basic techniques utilized by both the population creation and search phases of PBS are detailed below, the population creation phase is presented later, and the search phase described after that.
be added to the population or discarded. Basically, a new cluster will be unconditionally added to the population if its structure niche group is below the maximum allowed group size and the energy of the new cluster is not within 0.1 above of an existing member of the structure niche group. If this is not the case, then the new cluster will replace the highest energy member of the structure niche group, provided its energy is less than that of the member.

Two-Phase Local Search

If the strain factor $S$ of the cluster to be optimized is less than $S_{\text{cut}}$ then the cluster will first be optimized using the potential transformation method described in ref. 15. For the potential transformation, the $D_{\text{15}}^\dagger$ parameter is proportional to the number of atoms in the cluster while the weights $15$ are mapped to a grid, dependent on the lengths of the mid and minor axes of the cluster. If an axis length is $>$0.85, then the corresponding weight becomes one; otherwise, the weight is equal to the rounded down (to one decimal place) axis length. All clusters are then locally optimized using the Limited Memory BFGS (LBFGS)$^{19}$ algorithm.

It is important to note that PBS has no inbuilt preferences with regard to the parameters to be used for the Two-phase Local Search. The parameters used are totally dependent on the current structure of the cluster.

Directed Optimization

The goal of the Directed Optimization operator is to iteratively identify and repair surface and interior “defects” in clusters. Directed Optimization functions in one of the following three modes:

1. **Surface Repair**—moves the atom with the lowest number of Nearest Neighbor (NN) atoms to the best adjacent vacant position near a target atom with the maximum (but less than 12) number of NN atoms. In this context, NN atoms are those whose squared distance from the target atom is in the range $0.81 \ldots 1.54$. The best adjacent vacant position is obtained by constructing all possible tetrahedron apexes for the triangles formed by all possible combinations of the target atom and its NN atoms. Basically, this mode attempts to move the “worst” surface atom to the “best” available position on the surface and is used during the generation of new clusters and whenever an existing population member is mutated.

2. **Interior Repair 1**—randomly selects an atom from all atom pairs that are closest to each other (provided the separation is less than 1.123) and removes this from the cluster, locally optimizes the remaining $N - 1$ cluster using LBFGS and then adds the removed atom back onto the cluster using the Surface Repair technique described above. This mode of Directed Optimization is used during the generation of all new clusters.

3. **Interior Repair 2**—randomly selects an atom from all atom pairs whose separation is less than 1.123, locally optimizes the remaining $N - 1$ cluster using LBFGS and then adds the removed atom back onto the cluster using the Surface Repair technique described above. This mode is used as a local mutation operator for current population members.

Basic PBS Techniques

The population management and search techniques that are common to both the population creation and search phases of PBS are described in the following subsections.

Population Structure Niche Groups

The PBS population contains a minimum of 8 and a maximum of 16 members. Within this population, an individual structure “niche group” will have a maximum of eight members. PBS uses an absolute (rather than comparative) structure metric to create and maintain structure niche groups within the population. The PBS structure metric $S$, the “strain factor” of the cluster is defined as the percentage of atom pairs whose interaction energy is in the range $-0.8 \ldots -0.95$. The $S$ values for the optimal configurations of all clusters in the range $2 \leq N \leq 372$ are shown in Figure 2. As can be seen, the cluster sizes that have nonicosahedral optima all have $S < S_{\text{cut}}(= 0.3)$, whereas, for $N > 70$, all icosahedral optima have $S > S_{\text{cut}}$. Thus, $S$ is an effective metric for identifying structure within PBS with basically zero computational cost as it is produced as a side effect of calculating the cluster energy. Within PBS, $S$ is used to classify a cluster into one of two niche groups within the population, those where $S < S_{\text{cut}}$ (the nonicosahedral group) and those where $S \geq S_{\text{cut}}$ (the icosahedral group). In addition, when $S < S_{\text{cut}}$ for a cluster, the potential transformation phase of the two-phase local search will be used for the cluster.

Population Updating

As each subtask returns the results of a cluster generation, mutation or crossover the master task decides if the new cluster should

Figure 2. The strain factor of the optimal configurations of Lennard–Jones clusters in the range $2 \leq N \leq 372$. Note that, to show important details, the $y$-axis is truncated at $S = 2$. All the nonicosahedral clusters ($N = 38$, 75–77, 98, 102–104, 188–192, 236–238) have an $S$ value of less than $S_{\text{cut}} = 0.3$. Although some of the smaller clusters are incorrectly classified by $S$, this is not an issue as these size clusters are relatively straight forward to optimize.
The primary motivations for the Directed Optimization operator are that, from Figure 1, it is clear that: (a) every atom should have the maximum possible number of nearest neighbors. The surface repair mode contributes to this by ensuring that all atoms in the outer “shell” are placed adjacent on the surface of the cluster; (b) if two atoms are nearest neighbors, then the distance between them should be as close to 1.123 as possible. The interior repair modes contribute to this by identifying atom pairs that are closer than the optimal distance and creating a situation that rectifies this and also causes a reorganization of the neighboring atoms.

Figure 3 shows the performance of the Directed Optimization operator during the optimization of the \( N/H_{11}005 \) \( 10^4 \) cluster. Given that the Directed Optimization operator is only applied to a cluster that has already been locally optimized using two-phase local search, Directed Optimization is clearly very effective in following the energy funnel and is, probably, the key factor for the performance of PBS.

**PBS Population Creation Phase**

PBS uses a technique for initially generating the population that is effective in following energy funnels. In fact, PBS is generally able to successfully optimize all clusters in the range 2...65 plus others outside this range during the population creation phase. Initially, PBS generates a maximum of 150 trial solutions, randomly generated within a cube whose volume is dependent on the number of atoms in the cluster. All trial solutions are subsequently locally optimized and then subjected to the Directed Optimization operator. From Figure 4, which shows the resulting energies obtained during the population creation phase for the \( N = 104 \) cluster, it can be seen that this phase of PBS is effective in generating a range of cluster energies, all relatively close to the putative global minimum. With regard to other cluster sizes, Figure 5 shows the absolute difference between the putative global minima and the best obtained by the PBS population creation phase for clusters in the range \( 2 \leq N \leq 372 \). Generally, the PBS population creation phase is able to generate initial populations that either contain the putative global minimum or contain clusters whose energies are reasonably close to the putative global minima.
run-time, median/average number of local optimizations has been achieved. Clearly, this is of considerable benefit to the subsequent PBS search phase.

**PBS Search Phase**

The PBS search phase uses mutation and crossover operators to generate new starting points for the local optimization methods described below. The mutation and crossover operators operate only on population members and function either globally or locally on a cluster to implement coarse and fine-grained searches. It should be noted that, at some point during the optimization of clusters in the range $2 \leq N \leq 372$, all mutation and crossover operators described here were the last operator applied immediately prior to finding at least one global optimum.

**Fine-Grained Search**

The primary role of the mutation and global mutation operators is to move around the search space in large steps. For crossover, all possible combinations within the population are used and two crossover operators exist: the random crossover (used with probability 0.8), which randomly rotates the clusters about the three axes, selects some number of atoms and, using the ones most distant from the $x$–$y$ plane, swaps these by translating atoms using the most distant atom from each cluster as the basis for the translation. The second crossover operator is the selective crossover, which attempts to combine “good” hemispheres from each cluster when it generates the child clusters.

The global mutations operate on a single cluster and affect all atoms in the cluster. These are applied to each member of the population and perform twist (atoms are rotated about a random axis, where the angle of rotation increases with distance from the lowest atom on that axis) and perturb (a small, random perturbation of all atoms) operations. In addition, a small number of new clusters are randomly generated during the search phase of PBS.

**Experimental Results**

The performance metrics used to classify algorithms should be complete in that they measure all of a particular aspect of the algorithm. For example, as a measure of computational effort or run-time, median/average number of local optimizations has been used. However, this measure does not effectively compare algorithms that have different uses for local optimization (e.g., only optimizing near-optimal clusters as against optimizing random clusters) or where the local optimizations have differing computational overheads. It is perhaps the appropriate metric when a relatively large number of repeated global optimizations are performed on a particular cluster size and when the algorithms being compared use local optimization in the same context.

The view taken in this study is that the use of processor time as a measure of computational effort provides a more encompassing metric as it reflects the total amount of work performed by the search. In addition, it was the metric chosen by the two most directly comparable prior studies to this article. However, the processor time metric does have the disadvantage that it is clearly dependent on the computer processor used for the test, and this makes comparison between algorithms tested on different computers difficult. To overcome this, the method proposed in the COCONUT project has been used and the computer processor time, in terms of the processor time taken to evaluate the shekel5 function at $1.0E+8$ points (AC++ program is available at http://www.mat.univie.ac.at/neum/glopt/coconut/shkel5.cpp) is documented. Although this provides only a basic measure of a computer processor, we feel it is adequate for comparing processor time requirements for optimization algorithms. All single-processor experiments in this article were performed on a dedicated computer that required 269.33 processor seconds to execute a nonoptimized version of shekel5, compiled under Linux using the g++ compiler.

Table 1 gives processor time statistics, from 10 runs of the single-processor version of PBS, to successfully optimize specific...
clusters in the range \(2 \leq N \leq 192\) while Figure 7 shows the actual single-processor time, from a single run of PBS, which successfully optimized all clusters in the range \(2 \leq N \leq 200\).

With regard to the multiprocessor environment used in this article, the situation was less than ideal. There was available up to a total of 30 2-Ghz Linux processors. However, it should be noted that these processors were available to other users and also the number on-line varied from day to day. The elapsed times (including all task startup and stopping times) for the parallel version of PBS are shown in Figure 8 for clusters in the range \(2 \leq N \leq 372\). With regard to repeatability in finding the putative global minima, PBS was run through the range \(2 \leq N \leq 372\) three times and, without exception, found all global minima as listed in ref. 12 with elapsed times similar to those shown in Figure 8. In addition, PBS was successfully applied numerous times to the “difficult” clusters within the \(2 \leq N \leq 372\) range.

### Conclusion

This article presented the results obtained using a population-based search, PBS, for optimizing Lennard–Jones clusters. PBS was able to repeatedly obtain all optimal configurations in the range \(2 \leq N \leq 372\) as reported in ref. 12. The PBS algorithm incorporates and extends key techniques that have been developed in other Lennard–Jones optimization algorithms over the last decade. Of particular importance are the use of cut-and-paste operators, structure niching (using the cluster strain energy as a structure metric), two-phase local search, and a new operator, Directed Optimization, which extends the previous concept of directed mutation. In addition, PBS is able to operate in a parallel mode for optimizing larger clusters.

Future plans for PBS include improving the population creation phase for larger clusters, implementation of additional mutation

---

### Table 1.

The Mean, Median, and Standard Deviation of Processor Times, in Seconds, for 10 Successful Runs (with No Discards) of the Single-Processor Version of PBS for a Selection of Icosahedral and Nonicosahedral Clusters in the Range \(2 \leq N \leq 192\).

<table>
<thead>
<tr>
<th>(N)</th>
<th>Mean</th>
<th>Median</th>
<th>SD</th>
<th>Mean</th>
<th>Median</th>
<th>SD</th>
<th>(N)</th>
<th>Mean</th>
<th>Median</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>17</td>
<td>12</td>
<td>13</td>
<td>40</td>
<td>4</td>
<td>4</td>
<td>50</td>
<td>8</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>60</td>
<td>17</td>
<td>11</td>
<td>19</td>
<td>70</td>
<td>84</td>
<td>84</td>
<td>99</td>
<td>30</td>
<td>75</td>
<td>150</td>
</tr>
<tr>
<td>76</td>
<td>631</td>
<td>353</td>
<td>807</td>
<td>77</td>
<td>642</td>
<td>626</td>
<td>429</td>
<td>80</td>
<td>153</td>
<td>132</td>
</tr>
<tr>
<td>90</td>
<td>247</td>
<td>251</td>
<td>15</td>
<td>100</td>
<td>374</td>
<td>371</td>
<td>14</td>
<td>102</td>
<td>315</td>
<td>378</td>
</tr>
<tr>
<td>103</td>
<td>541</td>
<td>404</td>
<td>361</td>
<td>104</td>
<td>725</td>
<td>648</td>
<td>262</td>
<td>110</td>
<td>413</td>
<td>413</td>
</tr>
<tr>
<td>120</td>
<td>469</td>
<td>596</td>
<td>216</td>
<td>130</td>
<td>220</td>
<td>114</td>
<td>258</td>
<td>140</td>
<td>245</td>
<td>165</td>
</tr>
<tr>
<td>150</td>
<td>936</td>
<td>938</td>
<td>19</td>
<td>160</td>
<td>1271</td>
<td>1214</td>
<td>143</td>
<td>170</td>
<td>479</td>
<td>266</td>
</tr>
<tr>
<td>180</td>
<td>1486</td>
<td>1481</td>
<td>33</td>
<td>188</td>
<td>3772</td>
<td>2768</td>
<td>2873</td>
<td>189</td>
<td>5052</td>
<td>4789</td>
</tr>
<tr>
<td>190</td>
<td>4388</td>
<td>3407</td>
<td>2800</td>
<td>191</td>
<td>2459</td>
<td>1660</td>
<td>3845</td>
<td>192</td>
<td>2878</td>
<td>3219</td>
</tr>
</tbody>
</table>

**Figure 7.** The computer processor times, for the single-processor version of PBS, for clusters in the range \(2 \leq N \leq 200\).

**Figure 8.** The elapsed times (including task startup and stopping times) for the multiprocessor version of PBS for clusters in the range \(2 \leq N \leq 372\). All multiprocessor experiments were performed using a maximum of 30, 2-Ghz nondedicated processors.
operators such as stretch and compress, and an enhanced version of the Directed Optimization operator. In addition, with a dedicated 128-node cluster shortly to become available, PBS will be applied to larger Lennard–Jones clusters and also extended to other, related problems such as Morse Clusters, Mixed Clusters, Benzene Clusters, and Water Clusters.

Acknowledgment

The author wishes to thank Marco Locatelli for fruitful discussions and for providing the code for two-phase local search.

References