Computational and Theoretical Chemistry 1107 (2017) 7–13

Contents lists available at ScienceDirect

Computational and Theoretical Chemistry

journal homepage: www.elsevier.com/locate/comptc

Conquering the hard cases of Lennard-Jones clusters with simple recipes

Mark Dittner, Bernd Hartke *

Institute for Physical Chemistry, University of Kiel, Olshausenstr. 40, 24098 Kiel, Germany

ARTICLE INFO

Article history:
Received 1 September 2016
Received in revised form 24 September 2016
Accepted 26 September 2016
Available online 28 September 2016

Keywords:
Non-deterministic global optimization
Evolutionary algorithms
Cluster structures
Order parameters
Deceptive energy landscapes

ABSTRACT

Lennard-Jones clusters are the best-known benchmark for global cluster structure optimization. For a few cluster sizes, the landscape is deceptive, featuring several funnels, with the global minimum not being in the widest one. More than a decade ago, several non-deterministic global search algorithms were presented that could solve these cases, mostly using additional tools to ensure structural diversity. Recently, however, many publications have advertised new search algorithms, claiming efficiency but being unable to solve these harder benchmark cases. Here, we demonstrate that evolutionary algorithms can solve these hard cases efficiently, if enhanced with one of several very different diversity measures (niching) which were set up in an ad-hoc way, without extensive deliberation, testing or tuning. Hence, these hard benchmark cases should definitely be considered solvable. Additionally, these niching concepts offer insights into the different Lennard-Jones structural types, and into the way niching works in evolutionary algorithms.

1. Introduction

Clusters of n atoms bound exclusively by pairwise Lennard-Jones (LJ) model potentials of the form $\widetilde{E} = 4\sum_{i<j}\left(\tilde{r}_{ij}^{-12} - \tilde{r}_{ij}^{-6}\right)$ (with the energy E and the pair distance r in reduced units of the pair well depth and distance) have been a standard benchmark for global cluster structure optimization algorithms for a long time [1–17]. As frequently noted in these and other studies, the global minima for most of the cluster sizes n are Mackay isosahedra and easy to find, despite the exponential increase of search space size with n. However, for a few isolated cluster sizes, the structure of the true global minima is different: decahedral for $n = 75$, 76, 77, 102, 103, 104, face-centered cubic (fcc) for $n = 38$, and tetrahedral for $n = 98$. Astonishingly, the latter case was discovered only in 1999 by Leary and Doye [18], i.e., it was missed by several of the first studies cited above, which documents that it is hard to find.

These isolated occurrences of different global minimum structures are linked to partially filled structural shells and the different ways structural strain (deviations from ideal pair distances) can be accommodated in different structural types, as clearly illustrated by Doye et al. [19]. These authors also demonstrated that locating the true global minima in these cases is hard because most of search space is still dominated by the standard icosahedral pattern and its associated funnel-like landscape, while the different structural patterns (also containing the global minimum) reside only in a small region of search space, isolated from the remainder by high energy barriers.

This also explains why the case $n = 38$ was considered very hard in the early days of LJ cluster studies, despite its small size, and why it is apparently acceptable to admit problems with the larger hard cases in publications up to the present day, despite explorations towards sizes up to $n = 1000$ and beyond quite some time ago [20]. For example, Lv et al. [10] reported good results for $n = 75$ but failed to find the T_d minimum for $n = 98$ in 7 out of 10 cases. Laykhov et al. [11] called $n = 75$ "exceptionally complex". Rogan et al. [12], Zhang et al. [16] and Avendaño-Franco et al. [17] even failed to find the decahedral global minimum for $n = 75$, the latter two in publications of the present year 2016. This is astonishing, given that 10–15 years earlier, several publications, e.g., Refs. [3,21,22], had already presented recipes that successfully reduced the search effort for several or all of these hard cases. Therefore, the present contribution serves to reconfirm those earlier works: Present-day publications aspiring to conform to the state of the art should be able to deal with these hard cases, because this does not require specialized, fine-tuned recipes but merely a somewhat more judicious and robust design of the search algorithm.

In fact, what is needed has been known since the early days of non-deterministic global search and has been re-analyzed many times, also in recent years [23]: The practical strength of these algorithms lies in their deliberate refusal to cover all search space; instead the search is narrowed down on "promising regions". This

* Corresponding author.
E-mail address: hartke@pctc.uni-kiel.de (B. Hartke).

http://dx.doi.org/10.1016/j.comptc.2016.09.032
2210-271X/

can lead to very large performance enhancements, compared to deterministic search, which always has to cover all search space, at least indirectly. In many cases in practice, this makes the difference between being able to solve a global optimization problem and having to give up. The price to pay for this advantage is that so-called "deceptive" search landscapes can trap non-deterministic search in regions that do not contain the global minimum. Hence, mechanisms are needed to avoid such a trapping.

Obviously, for LJ clusters, the trick is to avoid that all search power is spent within the broad icosahedral basin, which acts as a strong attractor for non-deterministic search. This has been done already, for example with niching in Evolutionary Algorithms (EAs) [3]. As a side note, we are very much in favor of EA nomenclature to become "less inspired" [24,25]; in this sense "niching" should be called "introduction of an order parameter" instead. Nevertheless, to make contact to previous EA literature, we continue to use the biologically inspired term "niching" here. In Ref. [3], structures similar to the icosahedral and decahedral type were differentiated by rotating each cluster into an orientation in which a two-dimensional plane projection of its atom positions was least dense, and then calculating this density as the fraction of occupied squares in a discretization of this plane. Icosahedral structures have a significantly higher projected density than decahedral ones. The actual niching then allows only a small number of individuals (much smaller than the whole population) to have similar projected densities. This projection niching in Ref. [3] was very much ad-hoc, tainted with a priori knowledge, and computationally expensive, since the desired differentiation can only be made very close to the ideal cluster orientation, requiring a long sequence of small incremental test rotations, at each of which the 2D projection has to be evaluated.

Within their adaptive immune optimization algorithm (AIOA), Cheng et al. [21] have based their niching-like diversity concept on differences in nearest-neighbor connectivity table entries, between two structures. This depends on the proper choice of a cutoff criterion, to discern small differences in nearest-neighbor distances. Otherwise, with a looser cutoff criterion, all inner atoms always have 12 nearest neighbors, as shown below and as to be expected for closest packings between particles with non-directional interactions. However, with proper choice of this cutoff, these authors achieved impressive efficiency for the LJ hard cases, including $n = 98$.

Rossi and Ferrando [22] implemented a similar niching-like concept in Monte Carlo with Minimization (MCM) [26], also known as basin-hopping (BH) [27,28]. In their implementation, several simultaneous MCM walkers repel each other in an order-parameter space. With suitable choice of these order parameters, exploration can be diverted into different funnels. For LJ clusters, they found significant search efficiency enhancement for $n = 38$ and 75. To differentiate between icosahedral, decahedral and fcc structures, they chose the common neighbor analysis (CNA) [29–32].

CNA is one of several ways [33] to categorize nearest neighbor arrangements of atoms. It is used frequently to detect structural faults, domain boundaries and phase transitions in bulk MD simulations [34,35], but also for structural characterization of clusters [36,37]. In CNA, to each atom pair, an integer triple (m,n,k) is assigned, with m nearest neighbors common to both atoms in the pair, between which there are n bonds, and k bonds of these form the longest connected chain. As pointed out by Ferrando et al. [22,38,39], it is sufficient to monitor the CNA signatures (5,5,5), (4,2,2) and (4,2,1) to distinguish icosahedral-, decahedral- and fcc-structured clusters.

While Rossi and Ferrando have shown [22] that CNA-based differentiation does help for the LJ hard cases $n = 38$ and 75, it is unclear if it also works for $n = 98$ with the different T_d structure. Further possible downsides of CNA are that it is pair-based instead of atom-centered, and that intuitive correspondences between the (m,n,k) designation and actual local neighborhood structures are unclear (except for (5,5,5) which is normally linked to local 5-fold symmetry axes).

To emphasize with the present contribution that special characteristics of niching or diversity concepts are not important and that LJ hard cases can be solved by essentially any reasonable concept of this kind, we present two nichings that do have some aspects of similarity with the earlier ones but also several differences, and, most importantly, strongly differ from each other. Nevertheless, they achieve similar degrees of efficiency, when compared with each other and with earlier results, as mentioned above.

The first niching concept is based on a different local neighborhood categorization, which is atom-centered, can also differentiate T_d from icosahedral, decahedral and fcc, and is intuitively understandable. Hence, it also contributes insights into how these four basic LJ structural types differ, at the level of local nearest-neighbor arrangements. When this categorization is used to define niches in an evolutionary algorithm, this enables the EA to solve all these hard LJ cases with one and the same setting.

The other niching concept is based on the so-called Coulomb matrix (CM in the following), $\mathbf{M}$, which is used, e.g., also in Machine Learning studies as common measure of similarity throughout the chemical compound space (see Refs. [40,41] and references therein for a small overview as well as restrictions of the measure used here and Ref. [42] as a general investigation on similarity measures). Any cluster thus is represented by

$$M_{IJ} = \begin{cases} 0.5Z_I^{2.4} & \text{for } I = J, \\ \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} & \text{for } I \neq J, \end{cases} \tag{1}$$

with the atomic nuclear charges Z and distances $\mathbf{R}$ between the atoms I and J. The CM represents the Coulomb repulsion on the non-diagonal elements and a polynomial fit of the nuclear charges to the total energies of free atoms on the diagonal ones [43,44]. Note that in the case of non-mixed (atomic) LJ-clusters of this paper, all nuclear charges are the same, such that this reduces effectively to a matrix storing all N^2 (redundant) distances between the atoms. To construct a (dis-)similarity measure between two clusters, we use the Euclidean norm of the diagonalized CMs: $d(\mathbf{M},\mathbf{M}') = d(\boldsymbol{\epsilon},\boldsymbol{\epsilon}') = \sqrt{\sum_I |\epsilon_I - \epsilon'_I|^2}$ with $\boldsymbol{\epsilon}$ as ordered eigenvalues of $\mathbf{M}$. This now represents a translation-, rotation- and (atomic) permutation-invariant measure. Note, though, that this descriptor is not unique but coarsened, as in the eigenvalue vectors of the clusters effectively only N items of information are included (the additional information carried by the eigenvectors is completely discarded). However, as we want to form coarsened similarity niches over the energy landscape, this is in our case no disadvantage. Additionally, with a very small threshold on d, this can even be used in most cases as identity check for LJ clusters of the sizes studied here, despite the non-uniqueness.

2. Generic niching implementations

Global cluster structure optimization has been done here with the universal, object-oriented OGOLEM package [45,46], which has already been applied to a wide variety [9,47–53] of global optimization tasks. To avoid serious serial bottlenecks, OGOLEM implements the generation-free pool concept [54]. A generic niching implementation has been integrated with the pool concept allowing for arbitrary niching criteria to be employed.