Dissimilarity Measures for Population-Based Global Optimization Algorithms

paper presented at the Erice Workshop on "New Problems and Innovative Methods in Non Linear Optimization", 2007

Andrea Cassioli*- Marco Locatelli† - Fabio Schoen *

Abstract

Very hard optimization problems, i.e., problems with a large number of variables and local minima, have been effectively attacked with algorithms which mix local searches with heuristic procedures in order to widely explore the search space. A *Population Based Approach* based on a Monotonic Basin Hopping optimization algorithm has turned out to be very effective for this kind of problems. In the resulting algorithm, called Population Basin Hopping , a key role is played by a dissimilarity measure. The basic idea is to maintain a sufficient dissimilarity gap among the individuals in the population in order to explore a wide part of the solution space.

The aim of this paper is to study and computationally compare different dissimilarity measures to be used in the field of Molecular Cluster Optimization, exploring different possibilities fitting with the problem characteristics. Several dissimilarities, mainly based on pairwise distances between cluster elements, are introduced and tested. Each dissimilarity measure is defined as a distance between cluster descriptors, which are suitable representations of cluster information which can be extracted during the optimization process.

It will be shown that, although there is no single dissimilarity measure which dominates the others, from one side it is extremely beneficial to introduce dissimilarities and from another side it is possible to identify a group of dissimilarity criteria which guarantees the best performance.

KEYWORDS: Global Optimization, Cluster Optimization, Population-Based Approaches, Dissimilarity Measures.

*DSI - Università degli Studi di Firenze, Italy
†DI - Università di Torino, Italy

1 Introduction

A very effective approach for tackling highly multimodal optimization problems has been proved to be the so called *Population Basin Hopping* (*PBH*) algorithm (see (Grosso et al., 2007b)), a population based implementation of the well known *Monotonic Basin Hopping* (*MBH*) approach (Leary, 2000; Wales and Doye, 1997). *MBH* iterates through a sequence of perturbations followed by local optimizations. This turned out to be an effective strategy for functions with a funnel landscape (see, e.g., (Locatelli, 2005; Wales and Doye, 1997) for a description of such landscapes), so that *MBH* is often referred as a *funnel-descent* method.

In order to increase the search capability of *MBH*, a population framework has been proposed in (Grosso et al., 2007b). There, a collection of individuals is maintained and at each iteration a suitable perturbation/mutation operator is applied to each individual.

Then, an appropriate selection mechanism allows to define the population for the next iteration. This approach derives from the Genetic Algorithm (GA) family (see for example (Russel and Norvig, 1995)) often used for hard global optimization problems. The performance of the algorithm is strictly connected with the information used for the selection. Our strategy involves both objective function evaluations and a dissimilarity measure as criteria to decide upon the survival of each individual in the population.

The dissimilarity measure has a key role in the evolution of the population, being responsible to ensure that at each iteration a certain amount of dissimilarity between individuals is preserved. This should increase the capability of widely exploring the solution space by hopefully keeping inside the population individuals belonging to different funnels.

The paper is organized as follows. In Section 2 a general description of *PBH* is given. In Section 3 we introduce a general framework for dissimilarity measures. In Section 4 we present a special class of global optimization problems, the minimization of the Morse potential energy, which turns out to be particularly well suited to test different dissimilarity measures. The dissimilarity measures for such problem are presented in Section 5. Finally, in Section 6 we present and discuss the results of the computational experiments.

2 Population Basin-Hopping Algorithm

PBH is a Population-Based algorithm which tries to explore in parallel distinct regions of the solution space. The basic idea is to keep a set of solutions stored in what is usually called a *population* of *individuals*, from which a new set of

candidates is generated. The algorithm is briefly sketched in Algorithm 1 where Φ performs the *mutation/perturbation* operation on each member of the current population X^i, while $U(\cdot,\cdot)$ is the *update function*, which performs what in the field of genetic algorithms is called a *selection* process, choosing which new elements are allowed to replace some of the older ones.

while *stopping criterion is false* **do**
  $Y = \Phi(X^i)$
  $X^{i+1} = U(X^i, Y)$
  $i := i + 1$
end

Algorithm 1: a short sketch of *PBH*.

Note that, although other choices are possible, as a mutation/perturbation operator Φ we will always employ throughout the paper the one employed in the original *MBH* approach (see (Leary, 2000; Wales and Doye, 1997)), i.e., a random perturbation of the current individual followed by a local search started from the perturbed point. In particular, this means that individuals within the population are always local minima.

A very simple choice, which ensures the monotonicity for *PBH*, is to let a new individual substitute an old one if it has a better function value. In fact, a new candidate is compared with the worst element within the population and replaces it if it has a better function value. This simple update rule, which can be viewed as a greedy rule, is described in Algorithm 2 (f denotes the objective function of the GO problem at hand).

Input: X, Y
Output: X
foreach $y \in Y$ **do**
  let $c \in \operatorname{argmax}_j f(X_j)$
  if $f(X_c) > f(y)$ **then**
    $X_c = y$
  end
end

Algorithm 2: The greedy update rule.

If Algorithm 1 is implemented with the greedy update rule, the following behavior can be observed in practice:

- there are some (but, unfortunately, not always all) good optima towards which *PBH* quickly converges.
- The population X is quickly filled with these optima.
- *PBH* hardly escapes these optima.

The advantages and disadvantages of a greedy approach are well known: if the global optimum is within the set of easily reachable optima, then the greedy approach is able to reach it quite quickly; on the other hand, if the global optimum is outside this set, then the greedy approach very often will miss it. Therefore, being interested in building a method which is both efficient and robust, it is worthwhile to look for a way to reduce (in a sense to be better precised) the greedy effect, possibly loosing some efficiency on easy instances in order to gain effectiveness on the hard ones.

The key idea is to avoid new individuals to enter the population if someone similar (in a sense to be defined) is already inside; for instance, we would like, at least, to prevent the presence of more than one copy of the same individual in the population. This leads to the introduction of a *Dissimilarity Measure* (*DM* in what follows) $d(\cdot, \cdot)$ between members of the population which is used to maintain a sufficient diversity among its elements.

Using a dissimilarity measure d, the detailed sketch for *PBH* becomes that of Algorithm 3 (where m denotes the size of the population). The update

let X be randomly generated
while *stopping criteria is false* **do**
  $Y = \Phi(X)$
  for *k in 1..m* **do**
    $c \in \text{argmin}_j \, \text{d}(X_j, Y_k)$
    if $\text{d}(Y_k, X_c) \geq DCut$ **then**
      $c \in \text{argmax}_j \, f(X_j)$
    end
    if $f(X_c) > f(Y_k)$ **then**
      $X_c = Y_k$
    end
  end
end

Algorithm 3: *PBH* detailed sketch.

process has now two branches depending on a parameter $DCut$, which, in turn, depends on d. In any case an update step is performed in which a new individual Y_k replaces an old and worse one, X_c. The choice of element X_c to be possibly replaced by Y_k depends on the dissimilarity measure and on the $DCut$ parameter: X_c is chosen as the element which is the "least dissimilar" from Y_k, if the dissimilarity is smaller than $DCut$; otherwise, it is chosen in a greedy way as the element in the population with the worst function value.

The *PBH* update process involves different aspects like *hesitation* and *backtracking* which are ruled by the type of *DM* employed. For an experimental analysis about the *PBH* behavior we refer to (Grosso et al., 2007a).

The definition of $DCut$ allows in some sense to control the amount of greediness we want to keep in the algorithm, spanning between two opposite limit

cases:

$DCut \to 0$ - only the greedy branch tends to be (and, in fact, is when $DCut = 0$) active;

$DCut \to +\infty$ - only the non greedy branch tends to be active.

Although other choices are possible and, in some cases, as observed in (Grosso et al., 2007b) might also enhance the performance, in this paper we consider a standard definition for $DCut$, following (Lee et al., 2003), as the average value of $d(\cdot, \cdot)$ among all pairs of elements in the initial population. Although it is possible to update $DCut$ during the iterations of the algorithm, in this paper we decided not to explore this possibility.

For later reference we also introduce here a very simple population-based approach, Algorithm 4, indicated with `nodist` in what follows, where no col-

X is randomly generated
while *stopping criterion is false* **do**
  $Y = \Phi(X)$
  for k *in* $1..m$ **do**
    if $f(X_k) > f(Y_k)$ **then**
      $X_k = Y_k$
    end
  end
end

Algorithm 4: the `nodist` approach.

laboration between members of the population takes place. Each child Y_k is only compared with its father and the whole algorithm can be viewed as a set of m parallel and independent MBH runs. While trivial, we introduce also this algorithm as a reference for the other ones: of course, we expect that `nodist` is outperformed by all the other PBH approaches where collaboration takes place.

3 A framework for Dissimilarity Measures

From Section 2 it is clear that, although there is a stochastic component, the evolution of the population is mainly driven by the choice of the DM d and the induced value of $DCut$. Our aim is basically to define a *common framework* for DMs. Semi-metric properties should be at least fulfilled (see e.g. (Veltkamp and Hagedoorn, 1999) or (Veltkamp, 2001)), i.e., given two individuals A and B:

$$\begin{cases} d(A, B) \geq 0 \\ A = B \Rightarrow d(A, B) = 0 \\ d(A, B) = d(B, A) \end{cases}$$

These properties are sufficient to fit with *PBH* behavior requirements: the first and the last are obviously fundamental, while the second ensures that *PBH* recognizes pairs of *identical* individuals, which is important, as *PBH* needs to prevent similar configurations to be stored in the same population, preserving diversity within the population. Note that by identical we do not simply mean two individuals with the same coordinate values because in some problems we can consider as identical also individuals which can be obtained from each other by symmetry operations or, as in the case of the molecular conformation problems discussed in Section 4, by translation and/or rotation operations or even atom permutations.

The drawback is that different solutions might be marked as identical while indeed they aren't. This could be avoided with the stronger metric properties (usually with more computational effort), but such event hardly occurs in practice, so that semi-metric properties are usually sufficient.

In order to implement *PBH*, we focused on developing an easy to use yet effective framework common to the *DM*s we planned to use.

We have followed an approach which compute *DM* in two steps: the first creates a synthetic object/individual descriptor, usually focusing on problem dependent features; the second step returns the actual numerical value for the dissimilarity measure.
Such idea is well known and common in literature (see, e.g., (Belongie et al., 2002; Peura and Iivarinen, 1997; Osada et al., 2002)), although some authors (e.g, (Gunsel and Tekalp, 1998)) have proposed single step solutions.

Our basic idea is to define simple and easily adaptable measures to fit problems sharing common features, instead of creating each time a brand new one. In other words, we limit ourselves to *DM*s which can be defined as a suitable distance between two descriptors which can be computed separately for each individual. This is by no means the most general dissimilarity measure, as it does not include all measures which depend intrinsically on joint characteristics of pairs of individuals; an example of such a *DM* is the `RMSD` distance between two molecules, which is computed only after a suitable superimposition of one molecule over the other.

Let us introduce the following notation:

1. **descriptor operator** - $T : C_n \rightarrow S$ mapping elements from the object/individual space C_n to the descriptor space S;

2. **descriptor distance** - a suitable distance F, defined over the descriptor space S.

Then, every *DM* d we propose can be decomposed as follows:

$$d(A, B) = F(T(A), T(B)) : \mathcal{C}_n \times \mathcal{C}_n \rightarrow \mathbb{R}$$

Due to the non-unique mapping given by T, the dissimilarity measure will be just a semi-metric (see (Gunsel and Tekalp, 1998; Kolmogorov and Fomin, 1968)), since different individuals may have the same descriptor.

There are some other implementation issues worth to be considered:

- at each step of *PBH* we need to compute $m(m-1)/2$ dissimilarity measures (recall that m is the size of the population); since m usually grows with problem dimension, as larger problems need a larger population, we restrict to measures with low complexity. In any case, the time to compute all the dissimilarities at some iteration should be negligible with respect to the computational effort required by the local searches at the same iteration.

- Usually, children can neither inherit descriptors from their father nor use them to compute their own descriptors faster. This is basically due to the perturbation plus local optimization process: the former is usually randomly performed, the latter transforms the input in an unpredictable output even when the random perturbation only involves a small portion of the individual. Hence, descriptors have to be computed for every new child, i.e., m times for each *PBH* step.

From what stated before, it's clear how low complexity, for the computation of the descriptor operator T and the descriptor distance F, is a crucial point (see (Peura and Iivarinen, 1997; Veltkamp and Hagedoorn, 1999; Veltkamp, 2001)).

4 The Morse Potential Energy minimization problem

Although the *PBH* approach can be in principle applied to any global optimization problem, here we will restrict our attention to a special class of GO problems, the molecular conformation ones based on the Morse potential energy, which turn out to be particularly challenging and whose structure allows the definition of a wide variety of dissimilarity measures. Such problems are defined as follows. Given a cluster A of N *identical* atoms whose centers are in

$X_i^A \in \mathbb{R}^3$, $i = 1, \ldots, N$, the Morse energy of such cluster is

$$E(A) = E(X_1^A, \ldots, X_N^A) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} v(r_{ij})$$

where

$$r_{ij} = \|X_i^A - X_j^A\|_2$$

is the Euclidean distance between atom i and atom j in the cluster, and

$$v(r_{ij}) = (e^{\rho(1-r_{ij})} - 1)^2 - 1$$

is the Morse pair potential energy. Since the most stable configuration of a cluster is the one with minimum energy, in order to predict such configuration we are lead to solve the following GO problem

$$\min_{X_1^A, \ldots, X_N^A} E(A).$$

The parameter ρ is used to tune the shape of the contribution of a single atom pair, as can be seen in Figure 1. Increasing ρ gives a harder problem with a larger number of local minima (exponentially increasing with N) and, even worse, with a rougher funnel landscape. Here we will focus our attention on the case $\rho = 14$, the most challenging one among those reported in (Wales et al., 2007).

5 Proposed Dissimilarity Measures

In this section we introduce the different descriptors T and distances F we have identified to define dissimilarity measures for the problem of minimizing the Morse energy. We start with the introduction of some descriptors of an atomic cluster.

Pairwise distances cluster descriptor

Due to the explicit dependence of the objective function on pairwise distances between atoms in the cluster, these are natural candidates to be used in the definition of cluster descriptors (in particular, note that such distances are already available for free for each cluster because they are computed in order to evaluate the energy value of the cluster).

Given a cluster A with N atoms, the $N(N-1)/2$ distances between all its atoms give rise to matrix $D_A \in EDM^N$, where EDM^N represents the space of Euclidean Distance Matrices of N points in a suitable Euclidean space (which,

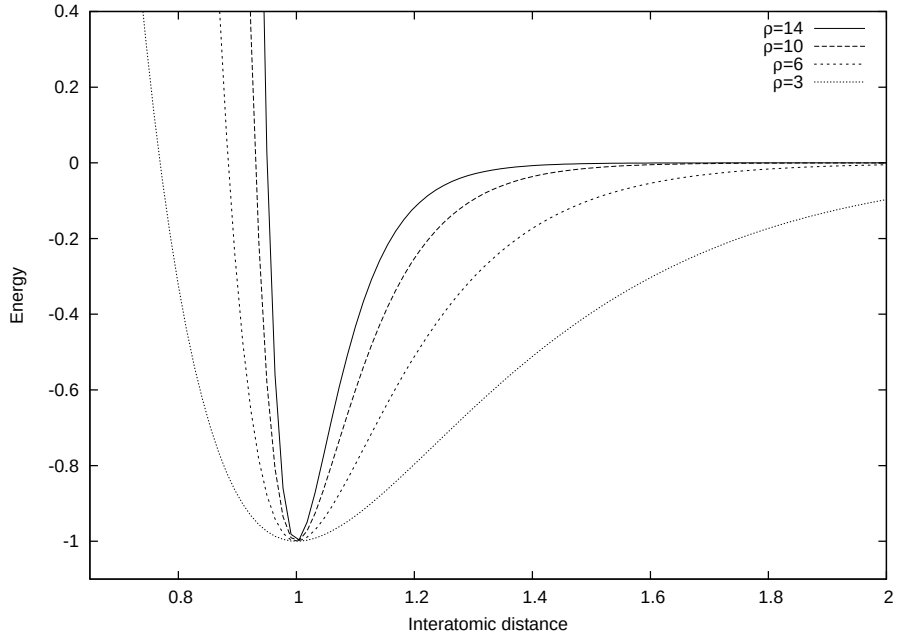

Figure 1: Morse Potential for different values of the range parameter ρ

in this case, is $\mathbb{R}^3$). Then, we might use matrix D_A as a descriptor for cluster A, i.e., in this case we have

$$T: \ A \in \mathbb{R}^{3N} \ \rightarrow \ D_A \in EDM^N.$$

Once we have defined a descriptor T, we need to define a distance F. A natural candidate distance between matrices could be the Frobenius norm, ending up with

$$d(A,B) = F(T(A),T(B)) = \|D_A - D_B\|_{Frob}.$$

A drawback of this dissimilarity measure is that it is sensitive to atom permutations, thus not fulfilling the required semi-metric properties. In other words, if we permute the labels of the N atoms, the cluster we obtain is the same, as all atoms are identical, but the descriptor changes. However, we can easily overcome this difficulty by using sorted distances. More precisely, we first redefine the descriptor T as follows

$$T(A) = sort(vect(D_A)),$$

i.e., we first convert, through the *vect* operator, the distance matrix D_A into a

vector, whose $N(N-1)/2$ components are then sorted in a nonincreasing way. After that, we can define F as the p-norm distance between the descriptors, i.e.,

$$F(T(A), T(B)) = \|sort(vect(D_A)) - sort(vect(D_B))\|_p.$$

It can be easily checked that this definition fulfils the required semi-metric properties.

Centroid Distances

An alternative to using all the $N(N-1)/2$ interatomic distances is to use only the N distances between each atom and the centroid of the cluster. Given a cluster A with atom positions $X_i^A \in \mathbb{R}^3$, $i = 1, \ldots, N$, its centroid is defined as follows

$$c_A = \frac{1}{N} \sum_{i=1}^{N} X_i^A$$

In general the centroid doesn't match the cluster's center of mass, but in our instances the two concepts coincide in view of the fact that we are assuming all atoms in a cluster are equal. The descriptor will be the N-dimensional vector of distances between each atom and the centroid of the cluster, again sorted in nonincreasing order, i.e.,

$$\begin{aligned} T(A) &: \mathbb{R}^{3N} \to \mathbb{R}^N \\ v_i^A &= \|X_i^A - c_A\|_2 \\ T(A) &= sort(v^A) \end{aligned}$$

Note that the local optimization strategy often includes a centering step, in which the cluster is translated in such a way its centroid is placed in the coordinates origin. In our experience, this operation, which also removes the translation degree of freedom, reduces numerical problems which sometimes arise since double precision numerical representations have their better resolution close to zero.

For what concerns the distance F, this can again be chosen as the p-norm distance between the descriptors.

Statistical Moments

A different way to take distances into account is to consider statistical moments. As usual, let us consider cluster A with atom positions $X_i^A \in \mathbb{R}^3$, $i = 1, \ldots, N$. Then, we can define the first ℓ moments μ_i, $i = 1, \ldots, \ell$, for the distribution of

all the interatomic distances as follows:

$$\begin{aligned}\mu_1 &= \tfrac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \|X_i^A - X_j^A\|_2 \\ \mu_r &= \sqrt[r]{\tfrac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} (\|X_i^A - X_j^A\|_2 - \mu_1)^r} \quad \forall\ r = 2...\ell\end{aligned} \tag{1}$$

Note that we use the r-th square root in order to reduce difficulties due to different factor scale. Then, the descriptor T is defined as follows

$$T : A \in \mathbb{R}^{3N} \rightarrow (\mu_1, \ldots, \mu_l) \in \mathbb{R}^{\ell}.$$

Distribution-based Descriptors

Following (Osada et al., 2002), shape descriptors can be constructed using sampling distributions for geometric properties, so called *shape distributions*. In our case, sampling can be substituted with the whole finite atom set and using the interatomic distance as geometric feature considered.
This set of data can be easily represented by an *empirical distribution function* (see, e.g., (Ross, 1987)), *edf* in what follows. Given a cluster A with atom positions X_i^A, its descriptor is the following

$$T(A) = Edf_A(y) = \frac{2}{N(N-1)} \sum_{i<j} \mathbf{1}\{\|X_i^A - X_j^A\|_2 \leq y\} \tag{2}$$

The *edf* is identically zero from $-\infty$ until the smallest distance in the cluster, non decreasing, piecewise constant and becomes identically equal to one as soon as the diameter of the cluster is reached. Figure 5 shows the *edf* for five clusters with $N = 30$.

Although the edf's are not perfectly readable from the figure, it is quite evident that they all share some characteristics, although every pair of them displays significant differences. We notice that jumps occur at some preferred distance values. As largely expected, the first high jump occurs at distance close to 1 (the minimum of the Morse pair potential). The next one occurs at distance approximately equal to $\sqrt{2}$, corresponding to the diagonals of squares formed by four atoms at distance one, and so on. Note that in this case the major differences are located among the larger distance values. This fact reflects the widely accepted (although never proven) remark that larger distances differ more than shorter, suggesting that outer atoms may be crucial in differentiating between clusters.

In the case of distribution-based descriptors, a common choice for the distance between descriptors is the Minkowski p-norm, defined as follows

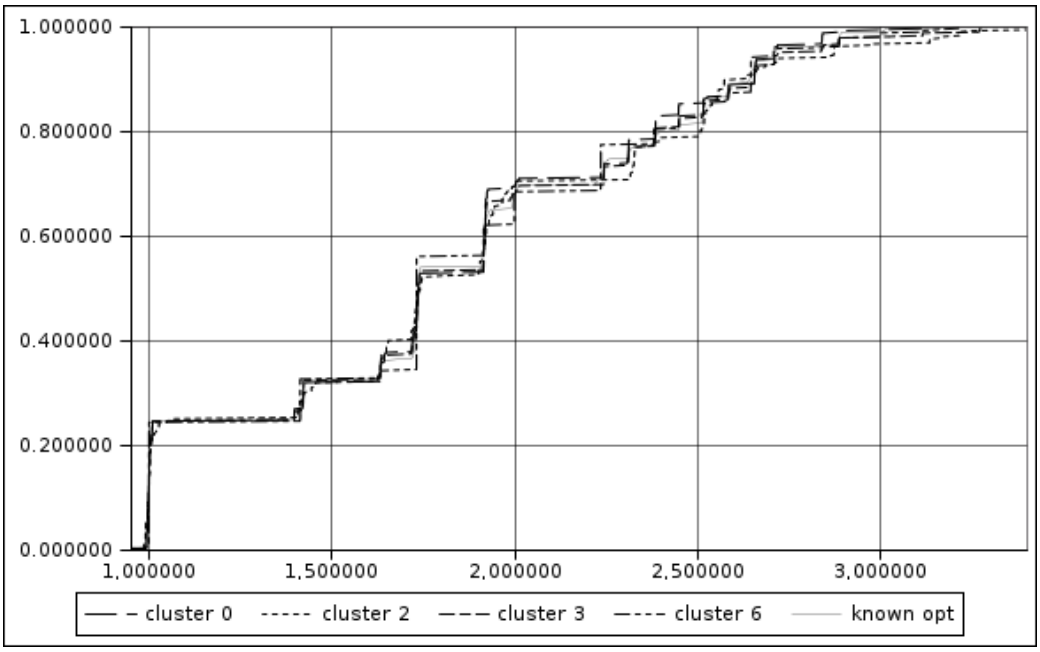

Figure 2: M_{30} *edf* example: the plot shows the curve obtained in the case of five local optima (in gray the putative global optimum).

$$F(T(A), T(B)) = L_p(Edf_A, Edf_B) = \sqrt[p]{\int_{-\infty}^{+\infty} |Edf_A(x) - Edf_B(x)|^p dx},$$

Given the fact that the edf's originate from a discrete distribution, the integral in the above formula reduces to a finite sum over intervals.

Histogram-based Descriptors

A different approach with respect to the *edf* one can be obtained considering a descriptor based on histograms. This approach is already known in literature (see (Lee et al., 2003)) and it turns out to be easy to compute and effective. The basic idea is to collect information about the local neighbor structure of each atom. Given a cluster A and a threshold distance $\tau > 0$, we define a function as follows

$$h_\tau^A(k) = \sum_{i=1}^{N} \mathbf{1} \left\{ \sum_{j=1,\ j\neq i}^{N} \mathbf{1} \left\{ \|X_i^A - X_j^A\|_2 \leq \tau \right\} = k \right\} \quad k = 0, \ldots, N-1$$

The above formula counts how many atoms in cluster A have exactly k neighbors at distance not larger than τ.

Several threshold values τ can be used, so that a kind of neighbors' hierarchy is generated.

Given the ℓ threshold values $\tau_1 < \cdots < \tau_\ell$, the descriptor of cluster A is defined as follows

$$T(A) = (h_{\tau_1}^A, \ldots, h_{\tau_\ell}^A)$$

Now we need to define a distance F. We employed the following one

$$F(T(A), T(B)) = \sum_{k=0}^{N-1} (k+1) \left\{ \sum_{r=0}^{\ell-1} (\ell - r) \cdot |h_{\tau_r}^A(k) - h_{\tau_r}^B(k)| \right\} \tag{3}$$

where, in the experiments, $\ell = 2$ was our preferred choice. Note that in (3) there is a different weight $(l - r)$ for the different neighbor levels τ_r. In particular, the weight decreases as the level increases.

Shell Decomposition

A simple iterative procedure can be used to decompose a given cluster in an outer-to-inner convex hull sequence of so called shells. We recall that in all the implementations of the algorithms presented in this paper the elements of each population are local optima of the Morse potential field. As it has been proven (see (Schachinger et al., 2007)) that locally optimal clusters do not degenerate in a plane (i.e., they are 3-dimensional objects), trivial computations show that the number of shells is at most

$$ns_{max} = \left\lceil \frac{N-4}{2} \right\rceil + 1.$$

The iterative procedure is sketched in Algorithm 5. Notice the innermost shell is obtained without actually evaluating its convex hull but relying again on a not planar structure. Let $Conv(A)$ denote the convex hull of a cluster A and let $\partial Conv(A)$ be its frontier.

An example of shell decomposition is reported in Figure 5

In our computational test we have used the freely available code from (Barber and Huhdanpaa, 2003), which implements the *Quickhull* algorithm for convex

let $i = 1$
let $\mathcal{P}_1 = A$
while $\mathcal{P}_i \neq \emptyset$ **do**
  if $|\mathcal{P}_i| \leq 2$ **then**
    $sh^A(i) = \mathcal{P}_i$
    $\mathcal{P}_i = \emptyset$
  else
    $sh^A(i) = \mathcal{P}_i \cap \partial Conv(\mathcal{P}_i)$
    $\mathcal{P}_{i+1} = \mathcal{P}_i \setminus sh^A(i)$
    $i = i + 1$;
  end
end
$n_A = i$

Algorithm 5: Shell Decomposition procedure

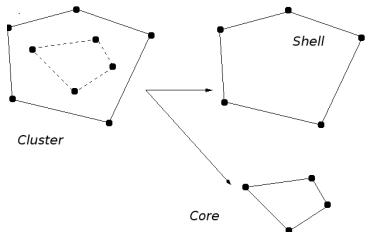

Figure 3: An instance of shell decomposition process.

hull (Barber et al., 1996), a combination between the *2-d Quickhull* algorithm with the *n-d beneath-beyond* algorithm (Preparata and Shamos, 1985). Complexity of convex hull computation in general dimension is still an open problem, although for two and three dimensions there exist efficient algorithms whose complexity is $O(n \log n)$. Relying on balancing assumptions on the algorithm execution (see (Barber et al., 1996) for details), *Quickhull* computes a three dimensional convex hull with an expected complexity of $O(n \log r)$, where n is the total number of input points and $r \leq n$ the number of processed ones.

Once we have the shell decomposition of cluster A, its descriptor will be represented by the list of its n_A shells, i.e.,

$$T(A) = (sh^A(1), \ldots, sh^A(n_A)).$$

Now, let A and B be two clusters, and let n_A, n_B be the corresponding number of shells. Then, a possible distance between their descriptors is the following

$$F(T(A), T(B)) = \epsilon_{AB} \sum_{i=1}^{n} d(sh^A(i), sh^B(i)) \tag{4}$$

where d is one of the previously defined dissimilarity measures, and:

- $n = \min(n_A, n_B)$;
- $\epsilon_{AB} = |n_A - n_B| + 1$;

This dissimilarity is not a metric even when d is a metric, since the triangular inequality doesn't hold in general, basically because of the different number of shells to be compared.

We remark that in (4) we are restricted to use only dissimilarity measures which can be applied to clusters having a different number of atoms, since in general the shell decomposition generates layers with different cardinality. For this reason, in the experiments with shells decomposition, we employed only the statistical moments and the *edf* based dissimilarities.

Some simple descriptors

We might wonder whether the use of quite elaborated descriptors is strictly necessary for the problem at hand. Indeed, it is possible to define simpler (and, in some cases, also more general) descriptors. In what follows we will introduce three of them and we will discuss the corresponding drawbacks.

A very simple descriptor for a cluster A is the identity one, i.e., $T(A) = A$, while we might choose F equal to the p-norm. However, the resulting dissimilarity measure does not fulfil the required semi-metric properties (it is sensitive to point permutations, to translations and to rotations of a cluster). We remark that even for more general GO problems this dissimilarity measure is unable to detect possible symmetries between different solutions.

Another very simple descriptor of a cluster A is its energy value (or the function value for general GO problems), i.e., $T(A) = E(A)$. In this case, since the descriptor is a real value, we might define F as the absolute value of the difference between the descriptors. Unfortunately, according to our experiments, the resulting dissimilarity measure is not particularly efficient for cluster optimization. Indeed, clusters with considerably different geometrical structure might have quite similar energy value. However, this simple dissimilarity measure might be a good one for other problems (see, e.g., the good results on the Schwefel test function reported in (Grosso et al., 2007b)).

Finally, we mention here as a possible descriptor of a cluster, its g value as defined in (Hartke, 1999). This is a real value based on the two-dimensional projection of the cluster. Such value is particularly efficient in discriminating between clusters with different geometrical structure (in particular, in discriminating between icosahedral, decahedral and FCC clusters), but is not able to

discriminate between clusters with the same geometrical structure. Some computational experiments have confirmed that the dissimilarity measure based on the g value is not an efficient one for the problem at hand.

A comparison between dissimilarities

In this section we briefly present an example of the use of the above dissimilarities; the aim of this example is to show that different descriptors in general produce a different ranking of individuals in a population and a different discrimination between "similar" and "dissimilar" solutions.

In order to show the effect of different dissimilarities, we choose from the Cambridge Cluster Database (Wales et al., 2007) five conformations with $N = 45$ atoms; to each of those conformations we applied a single local optimization in order to obtain a stable conformation for $\rho = 14$ and then we computed the dissimilarity matrix. In order to be able to compare different dissimilarity criteria, we present in the following tables, for every dissimilarity and every pair of different clusters, the relative percentage difference between each measure and the average.

Moments		45C'	45D	45E	45F
	Energy	-174.096	-174.510	-174.512	-174.398
45B	-169.713	-97%	15%	97%	-99%
45C'	-174.096		39%	129%	-98%
45D	-174.51			-89%	11%
45E	-174.512				93%

Centre		45C'	45D	45E	45F
	Energy	-174.096	-174.510	-174.512	-174.398
45B	-169.713	-91%	-5%	88%	-78%
45C'	-174.096		46%	159%	-52%
45D	-174.51			-78%	-27%
45E	-174.512				39%

Edf		45C'	45D	45E	45F
	Energy	-174.096	-174.510	-174.512	-174.398
45B	-169.713	-21%	5%	16%	39%
45C'	-174.096		-12%	2%	4%
45D	-174.51			-58%	8%
45E	-174.512				18%

Pairs		45C'	45D	45E	45F
	Energy	-174.096	-174.510	-174.512	-174.398
45B	-169.713	-47%	-1%	32%	13%
45C'	-174.096		-9%	31%	-20%
45D	-174.51			-80%	23%
45E	-174.512				60%

Hist		45C'	45D	45E	45F
	Energy	-174.096	-174.510	-174.512	-174.398
45B	-169.713	6%	35%	9%	13%
45C'	-174.096		48%	22%	-34%
45D	-174.51			-66%	-2%
45E	-174.512				-30%

Mom+Shells		45C'	45D	45E	45F
	Energy	-174.096	-174.510	-174.512	-174.398
45B	-169.713	18%	147%	154%	102%
45C'	-174.096		-39%	-33%	-63%
45D	-174.51			-99%	-94%
45E	-174.512				-92%

Edf+Shells		45C'	45D	45E	45F
	Energy	-174.096	-174.510	-174.512	-174.398
45B	-169.713	-51%	53%	59%	26%
45C'	-174.096		36%	41%	-1%
45D	-174.51			-66%	-50%
45E	-174.512				-48%

Table 1: Parameter sets for two-phase local searches

Parameter set	D	μ	w_1	w_2
P1	3.5-4	0.1	1.0	1.0
P2	3-3.5	0.0	1.0	0.7
P3	2.5-3	0.0	0.8	0.8
P4	2-2.5	0.0	0.75	0.5
P5	2-2.5	0.0	0.65	0.65
P6	2-2.5	0.0	0.5	0.5

Among other considerations, it seems worth to point out that closeness in energetic value does not always mean closeness in shape, as it can be seen in the comparison between clusters 45D and 45F. Also, by looking at the above tables, we can notice that some pairs of clusters are tagged as "close" with some criteria (negative deviation with respect to the average), while they are considered as more dissimilar than the average when using different criteria.

6 Computational results

6.1 Description of the experiments

As already remarked in (Doye et al., 2004; Grosso et al., 2007b), in minimizing Morse energy through *MBH* or *PBH* it is particularly useful to substitute the standard local searches with so called *two-phase* local searches, where in the first phase a modified energy, depending on few shape parameters D, μ, w_1 and w_2, is optimized, while in the second phase the original Morse energy is restored. We refer to (Doye et al., 2004) for a discussion about the meaning and the effect of the different shape parameters. Only a small set of parameter values have been considered in (Doye et al., 2004; Grosso et al., 2007b) and these are reported in Table 1.

In this paper we are mainly interested in comparing the efficiency of different dissimilarity measures. We measure the efficiency in terms of number of (two-phase) local searches required to reach the known putative global optimum for each number N of atoms. For a fixed N value we have not tested all the six parameter sets but just the *best* one. To be more precise, since the parameters w_1 and w_2 are strictly related to the moments of inertia of the clusters, for a given number N of atoms we have chosen the set of parameters for which the w_1 and w_2 values are closest to the moments of inertia of the putative optimum for that number of atoms.

We considered clusters composed of N atoms, where N is in the range

$41 \dots 80$; the size m of the population was chosen as a function of N according to the following rule

$$m = \begin{cases} 40 & 41 \leq N \leq 50 \\ 80 & 51 \leq N \leq 70 \\ 160 & 71 \leq N \leq 80 \end{cases}$$

For each N value and each tested *PBH* variant we performed 10 runs and stopped each run as soon as one of the following stopping rules was satisfied:

- a maximum number of iterations, *maxStep*, was reached; an iteration corresponds to calling the two-phase local optimization phase for all the elements in the population. *maxStep* was fixed to 1500 in the experiments;
- a maximum number of iterations *LocalMaxNoImprove* without any change of any individual in the population was reached; this was fixed to 500 in the computations
- a putative global optimum is found (see (Wales et al., 2007) for a list of them);

Of course, the last stopping criterion is only employed here in order to compare the efficiency of the different dissimilarity measures.

We need to specify the details of the mutation/perturbation operator Φ. Following (Grosso et al., 2007b), we add to each individual cluster in the population a uniform random perturbation within the hypercube $[-0.4, 0.4]^{3N}$, followed by a (two-phase) local search performed through the L-BFGS algorithm (see (Liu and Nocedal, 1989)).

For each tested dissimilarity measure we report in Table 2 the corresponding *PBH* identifier, its descriptor T, its metric F and finally, in the last column, a few other values which, in some cases, have to be specified in order to define it. Note that, since the `greedy` variant does not depend on any dissimilarity measure, we have not specified any descriptor and metric for it. Also, note that while for the `pairs` and `mom` variants we have employed as a metric F the 2-norm, the 3-norm has been employed for the `centre` variant. This is due to the fact that in (Grosso et al., 2007b) it was observed that the 3-norm was slightly better than the 2-norm (at least for the experiments performed in that paper) so that we decided to keep the 3-norm also in this paper. Finally, besides the *PBH* variants presented in Table 2, we have also tested the `nodist` approach introduced in Section 2.

Before discussing the computational results, we would like to point out that our main aim in performing the experiments was to computationally confirm the following claims:

PBH identifier	Descriptor T	Metric F	Other Info
`pairs`	Pairwise Distances	2-norm	-
`centre`	Centroid Distances	3-norm	-
`hist`	Histograms	Ad-hoc norm	$\ell = 2$ $\tau_1 = 1.25$ $\tau_2 = 1.55$
`mom`	Statistical Moments	2-norm	$\ell = 3$
`edf`	Edf	Minkowsky 1-norm	-
`edf_qh`	Shells+Edf	Shells Contrib.	
`mom_qh`	Shells+Stat.Mom.	Shells Contrib.	$\ell = 3$
`greedy`	-	-	$DCut = 0$

Table 2: List of the tested *PBH* variants

1. cooperation within the population is worthwhile, i.e., the `nodist` approach is usually less efficient than the different *PBH* variants based on cooperation between members of the population;

2. the `greedy` approach is *not* a robust one, i.e., it might be extremely efficient in some cases but also extremely bad in other cases.

As we will see below, these claims will be indeed confirmed by the computations.

6.2 Discussion of the results

The numerical experiments we performed involved a huge amount of computations. In fact we performed, for every N in $41, \ldots, 80$, 10 independent runs of PBH for each of the considered dissimilarity measures, using a two-phase local optimization in which the parameters are chosen as the "best ones" among the six possible parameter sets described in the previous paragraph. Thus we collected numerical results of PBH over a total of $40 \times 10 \times 9 = 3\,600$ runs, with an average population size of 90 individuals; the experiments, thus, correspond to more than 300 000 runs of a basin-hopping algorithm. The experiments were performed on several machines, including some high performance parallel ones.

In Table 3 we report, for each N and each tested dissimilarity, the average number of local searches per success (i.e. the total number of local searches in 10 runs divided by the number of successes in obtaining the putative global optimum); INF denotes cases where no success was obtained. In Table 4 we report the number of successes over 10 runs.

As a first general remark we can notice that no dissimilarity measure dominates the other ones. This was largely expected: the great variety for the shapes of the clusters is such that we can hardly expect that a single measure is always able to discriminate between different clusters better than all the other measures.

Table 3: Average number of local searches per success

N	centre	edf	edf_qh	greedy	hist	mom	mom_qh	nodist	pairs
41	790.4	690.3	498.3	997.1	709.8	610.7	551.5	573.3	713.2
42	2185.1	2212.5	2712.8	1185.7	2509.6	3685.3	3898.2	4233.1	4712.5
43	38606	58001.3	45933.3	89562	55699.6	30871.8	77320	30447.9	45125
44	INF	189266.7	INF	15505.8	249632.5	INF	547858	26006	578211
45	123729	82120	214958	29655.7	76262.7	72970.5	170219.3	131771.8	78433.4
46	7008.4	15448.9	11864.6	7805.3	8242.2	11192	23596.6	5949.8	14533.5
47	4120.2	14907.9	10948.8	230101	4258.5	6600	22099.3	21615	25716.9
48	667.7	645.2	518.3	532.3	597.5	571.9	695.6	595.6	370.2
49	1452	1445.9	1104.8	3588	972	1312	1358.8	1584	1224
50	84	126	138.3	88	100	80	138.3	52	112
51	373.3	259.4	370.5	287.9	362.8	385.7	238.9	280	594.4
52	3408	3603.9	4087.7	3768	2680	6472	3433.2	7640	4524.6
53	4496	3088	2085.4	2085.4	2694.3	3119.2	2176	4980.9	3200
54	347653	78600	41144	37884.4	96873.1	381413.3	257044	INF	117405.7
55	159920	143666.7	284520	62453.3	55760	62195.6	169775.7	1186880	103691.4
56	121611.4	37896	75150	INF	46168	96057.1	174501	489320	64337.8
57	5348.4	7928	11328	117760	8280	6047.5	9804.2	40515.6	10251.4
58	169.9	168	168	183.7	144	229.7	190.4	128	143.1
59	319	328	336	330.1	304	501	373.4	240	448.4
60	6696	3680	4296	3704	5112	21840	3537.7	3880	3896
61	18680	14776	20360	21866.7	19584	23856	32013.2	52306.7	10360
62	12360	7848	8120	11013.3	8216	5392	12397.6	16072	4088
63	1877.3	1256	1440	1446.6	1584	1871.1	1315.4	1552	928
64	3967.8	2480	3040	1639.4	2592	3401	3020.4	4536	2775.7
65	5596.7	3864	7688	4746.7	4472	4299.8	4291.2	8856	4084.5
66	6115.9	2904	5688	9979.5	4264	4884.7	5299.4	7304	3667
67	15752	9416	37626.7	141711.8	19672	30474.7	21137	15104	9536.2
68	8626.6	3784	12136	8776	7872	7128	11055.8	10520	4864
69	22427.4	9584	14728	64581.8	17702.5	14208	27857.8	17592	9560
70	5868.5	4952	4760	10896	3274.4	8312	6096	5184	4328
71	16976	15728	14944	88720	15552	31920	28976	20128	11792
72	2711.9	2368	4336	1248	2388.4	2896	2320	3024	2496
73	4775	2736	7296	2043.1	3741.3	4861.4	5104	3152	1728
74	367.5	288	352	287.1	300	416	256	256	272
75	1039	1072	976	770.1	557.9	1240	928	608	1040
76	5193.6	2656	3760	1488	2938.5	3730.6	3264	2896	2512
77	4637.3	3152	5056	1520	4089.7	3625	3152	4774.4	3754.6
78	13355.4	10864	6896	3264	10614.3	5840	7280	9089.4	9506.6
79	82292.7	662986.7	1084880	1073120	508840	2345280	685653.3	235360	606918.3
80	375082.6	2343520	2374240	INF	512920	INF	1050080	701320.7	2235520

Table 4: Number of successes over 10 runs

N	centre	edf	edf_qh	greedy	hist	mom	mom_qh	nodist	pairs
41	10	10	10	10	10	10	10	10	10
42	10	10	10	10	10	10	10	10	10
43	8	7	6	2	7	8	5	9	8
44	0	3	0	8	2	0	1	10	1
45	4	5	2	6	6	6	3	4	5
46	10	10	10	10	10	10	9	10	10
47	10	9	9	1	10	10	9	8	8
48	10	10	10	10	10	10	10	10	10
49	10	10	10	10	10	10	10	10	10
50	10	10	10	10	10	10	10	10	10
51	10	10	10	10	10	10	10	10	10
52	10	10	10	10	10	10	10	10	10
53	10	10	10	10	10	10	10	10	10
54	3	8	10	9	7	3	4	0	7
55	6	6	4	6	8	9	6	1	7
56	7	10	8	0	10	7	5	2	9
57	10	10	10	3	10	10	10	9	10
58	10	10	10	10	10	10	10	10	10
59	10	10	10	10	10	10	10	10	10
60	10	10	10	10	10	10	10	10	10
61	10	10	10	9	10	10	10	6	10
62	10	10	10	9	10	10	10	10	10
63	10	10	10	10	10	10	10	10	10
64	10	10	10	10	10	10	10	10	10
65	10	10	10	10	10	10	10	10	10
66	10	10	10	10	10	10	10	10	10
67	10	10	9	4	10	10	10	10	10
68	10	10	10	10	10	10	10	10	10
69	10	10	10	5	10	10	9	10	10
70	10	10	10	10	10	10	10	10	10
71	10	10	10	6	10	10	10	10	10
72	10	10	10	10	10	10	10	10	10
73	10	10	10	10	10	10	10	10	10
74	10	10	10	10	10	10	10	10	10
75	10	10	10	10	10	10	10	10	10
76	10	10	10	10	10	10	10	10	10
77	10	10	10	10	10	10	10	10	10
78	10	10	10	10	10	10	10	10	10
79	9	3	2	1	4	1	3	6	3
80	5	1	1	0	4	0	2	3	1

In order to analyze in more detail such results we will now present three distinct performance profiles.

The first one refers to all the 40 tested cases. A curve is drawn for each of the tested dissimilarities. The curve represents the number of instances for which the average number of local searches per success is within $x\%$ of the lowest such value among all tested dissimilarities (i.e., at $x = 0$ we have the number of distinct instances for which a single dissimilarity has the lowest number of local searches per success with respect to the others, at $x = 100$ we have the number of instances for which the average number of local searches is within 100% of the lowest average number of local searches among all tested dissimilarities, and so on). According to this graph, we notice that the `greedy` approach turns out to be quite good for small x values, but as x increases it gets quite worse and in the end it is, by far, the worst approach. This seems to suggest that when the `greedy` approach is able to detect the solution, it detects it quite fast but, on the other hand, on some instances it fails to reach a solution within a reasonable amount of time. For $x \geq 100$ the best dissimilarity appears to be the `hist` one, closely followed by `edf`. The `nodist` approach appears to be quite bad and only for large x values it gets close to the best dissimilarities, showing that the use of dissimilarities is indeed profitable.

Figure 4 reports the performance profiles relative to the number of local searches per success, while Figure 5 displays the performance profiles of the number of successes (in 10 independent experiments).

A more refined analysis has been performed by taking into account that the 40 tested instances are not completely independent from each other. Indeed, it often turns out that once the solution for some N value has been obtained, it is possible with a *forward* operation (addition of an atom in a random position) to get a solution for the instance with $N+1$ atoms, and with a *backward* operation (deletion of a randomly chosen atom) to get a solution for the instance with $N-1$ atoms. Using the forward and backward operations we defined a reachability graph where the nodes are the distinct N values and a directed arc between N and $N+1$ ($N-1$) is inserted if a forward (backward) move from the solution for N atoms allows to detect the solution for $N+1$ ($N-1$). Once the graph has been derived, we create distinct groups, corresponding to the distinct connected components of the graph. Once the solution for one of the members of a group has been reached, then by extremely cheap forward an/or backward operations we are also able to get the solutions for all the other members within the group. Taking into account the fact that in a group of reachable clusters, detecting any of them enables to quickly detect all, we consider as the average number of local searches for the group, the minimum of the average number of local searches within the group. The groups that we have identified (a total number of 17) are

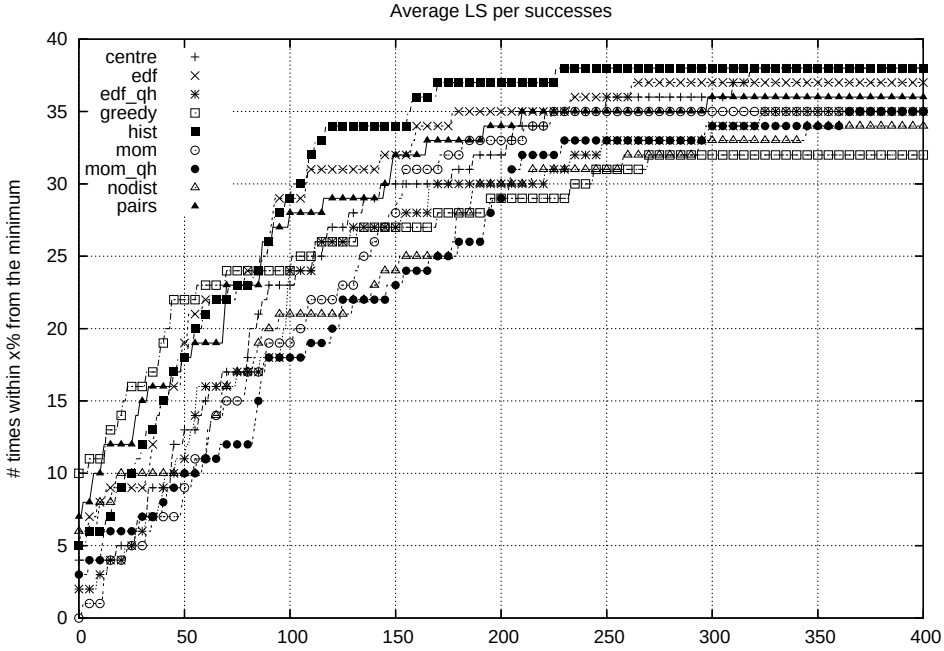

Figure 4: Performance profile: number of local searches per success

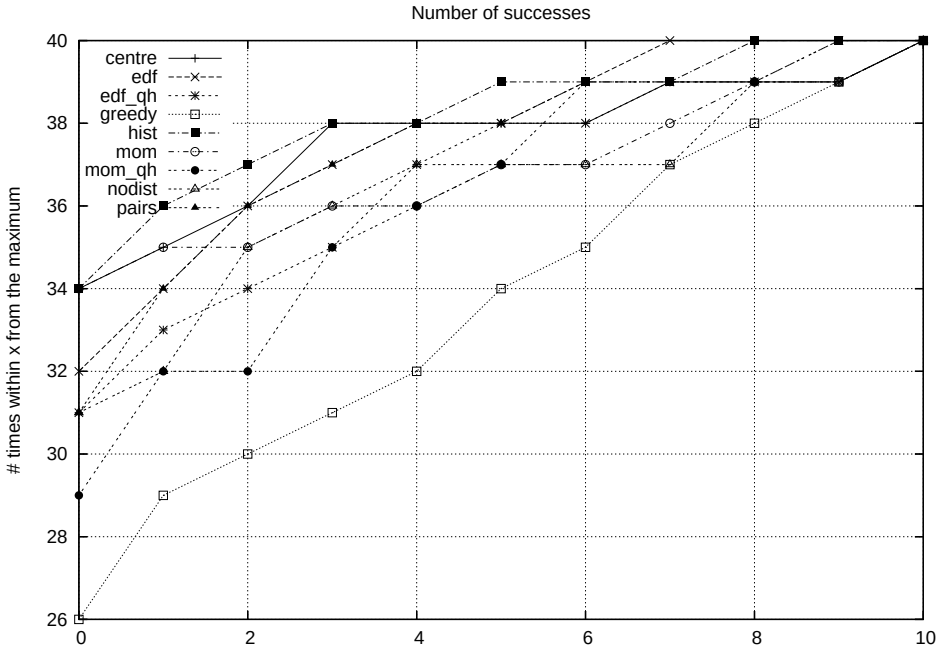

Figure 5: Performance profile: number of successes

the following:
$\{41, 42, 43, 44\}$, $\{45\}$, $\{46, 47\}$, $\{48, 49\}$, $\{50, 51, 52, 53\}$, $\{54, 55, 56, 57\}$, $\{58\}$, $\{59, 60\}$, $\{61\}$, $\{62\}$, $\{63, 64, 65, 66\}$, $\{67\}$, $\{68\}$, $\{69, 70, 71, 72, 73\}$, $\{74, 75, 76\}$, $\{77, 78\}$, $\{79, 80\}$.

The performance profile relative to groups of similar clusters is displayed in Figure 6.

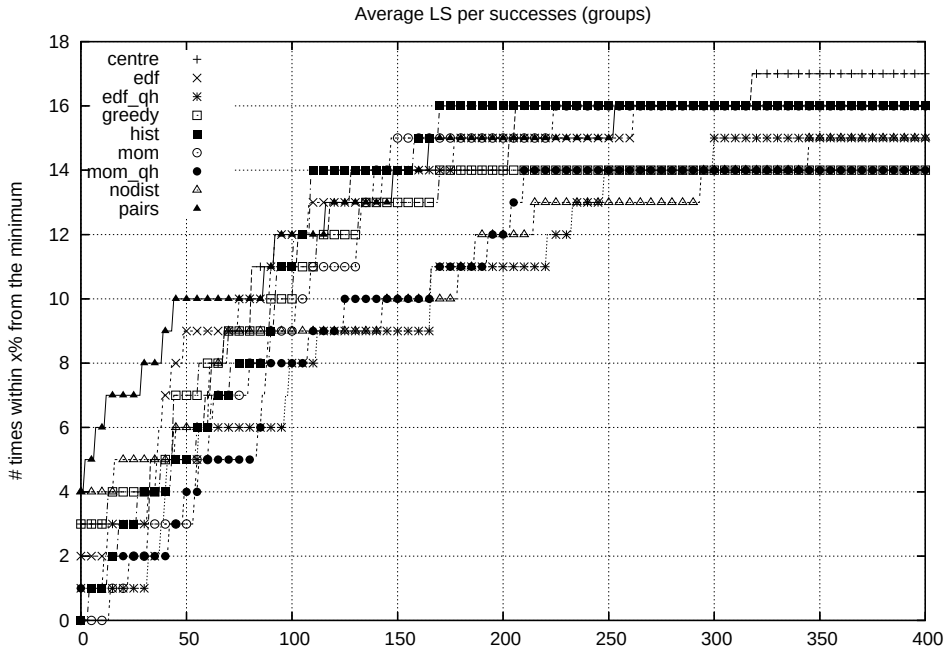

Figure 6: Performance profile: number of local searches per success in a group

The behavior of the `greedy` approach is similar to the one previously discussed: good (but no more the best) for small x values, but quite poor as x increases. It is interesting to notice that for small x values (up to 100) the `pairs` dissimilarity is quite clearly the best one. The `hist` dissimilarity is still quite competitive (although for small x values its behavior is not particularly brilliant), but now the `mom` and, even more, the `centre` dissimilarities get quite good for large x values. We have a confirm that the `nodist` approach is quite inferior with respect to the approaches based on dissimilarities. Unfortunately, we also have to remark the non particularly impressive behavior of the `edf_qh` and `mom_qh` dissimilarities. We performed some tests with variations of these dissimilarities in which a larger weight is given either to the outer shells or to the inner ones, but also such tests did not deliver significantly good results. In spite of these negative results, we can not conclude that the idea of subdividing the clusters into layers and measuring the dissimilarity between layers simply

does not work. It is possible that other strategies for the subdivision into layers can be defined and are able to deliver much better results.

Since for some groups the detection of solutions turned out to be extremely quick, no matter which dissimilarity we employed, we further decided to exclude from considerations all those groups for which the average number of local searches was below 1000 for all the tested dissimilarities. Basically, we identified this way only the more challenging groups. This operation reduced the number of groups to 11. However, as we can see form Figure 7, this further

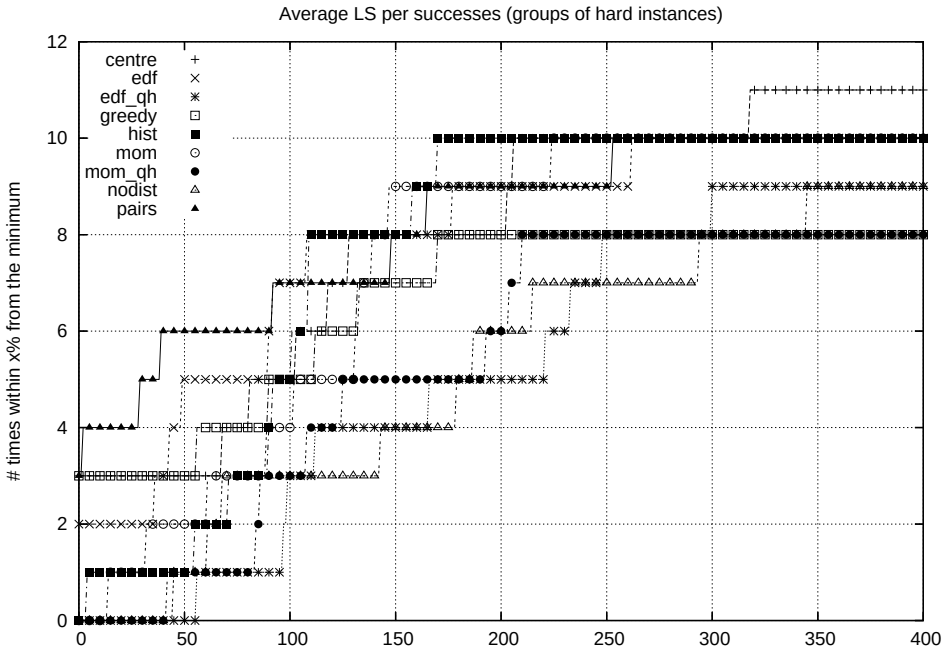

Figure 7: Performance profile: number of local searches per success in a group

refinement does not change the conclusions drawn by considering all possible groups.

7 Conclusions

The aim of this paper was to test the effectiveness of different dissimilarity criteria in a population based global optimization method applied to a notoriously difficult problem in atomic cluster conformation. We performed a very large set of computational experiments in order to try to understand if a population based approach for this problem is indeed an advantage and, if this is the case, to understand whether the dissimilarity criterion empoyed significantly influences the results. After our experimentation we think it is possible to give

an affirmative answer to both questions. Indeed the `nodist` approach, which simulates the execution of simultaneous runs of the monotonic basin hopping method with no exchange of information, is a loser with respect to population based approaches, at least for the majority of dissimilarity criteria we tested. Actually, in few cases, the introduction of a dissimilarity criterion in a PBH method produced worse results than those obtained with no exchange of information in the population. In particular, this was the case for the two criteria based upon shells decomposition. This negative result might be due to the fact that more investigation is needed in order to fully exploit the potential of decomposition – indeed a definition of a shell which does not require convexity might be better suited in the case of molecular clusters; we plan in the future to investigate the effect of choosing, e.g., the outer shell as the surface composed of all the "visible" triangles in a complete Delaunay triangulation of the cluster. In other words, the bad performance of shells decomposition might be due to a non optimal choice of the decomposition itself.

By the way, this negative result is a proof of the fact that different dissimilarities produce methods with very different performance. Indeed, a few criteria stand out as the best ones: this are the `pairs` and the `hist` approaches. The `greedy` approach, as it could be easily guessed, is excellent for easy test cases, but its performance rapidly decades.

As a final remark, we would like to observe that most of the dissimilarity measures introduced in this paper can be employed also for optimization problems different from the one we chose to analyze. In fact most, if not all, of them can be applied to optimization problems in which variables can be thought of as points in a suitable euclidean space. We may cite the fact that using an dissimilarity based on the `centre` descriptor, we implemented a PBH method for the problem of packing unequal disks into the smallest circular container in $\mathbb{R}^2$: this method led our team to win an international competition on circle packing (Addis et al., 2007).

Acknowledgments

This research has been carried out under the PRIN project "New Problems and Innovative Methods in Non Linear Optimization" and the HPC-EUROPA project (RII3-CT-2003-506079), with the support of the European Community - Research Infrastructure Action of the FP6.

A special thanks goes to the EPCC staff at the University of Edinburgh who helped in the development of a parallel version of *PBH* and made computational resources available to our group.

References

Addis, B., Locatelli, M., and Schoen, F. (2007). Efficiently packing unequal disks in a circle. *Operations Research Letters*, in press.

Barber, B. and Huhdanpaa, H. (2003). *QHULL Manual*. The Geometry Center, University of Minnesota, Minneapolis MN. www.qhull.org.

Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw*, 22(4):469–483.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522.

Doye, J. P. K., Leary, R. H., Locatelli, M., and Schoen, F. (2004). Global optimization of morse clusters by potential energy transformations. *INFORMS Journal On Computing*, 16:371–379.

Grosso, A., Locatelli, M., and Schoen, F. (2007a). An experimental analysis of a population based approach for global optimization. *Computational Optimization and Applications*, (to appear).

Grosso, A., Locatelli, M., and Schoen, F. (2007b). A population based approach for hard global optimization problems based on dissimilarity measures. *Mathematical Programming*, 110(2):373–404.

Gunsel, B. and Tekalp, M. A. (1998). Shape similarity matching for query-by-example. *Pattern Recognition*, 31(7):931–944.

Hartke, B. (1999). Global cluster geometry optimization by a phenotype algorithm with niches: location of elusive minima, and low-order scaling with cluster size. *J.Comp. Chem.*, 20:1752–1759.

Kolmogorov, A. and Fomin, S. (1968). *Introductory Real Analysis*. Dover Publications.

Leary, R. H. (2000). Global optimization on funneling landscapes. *J. Global Optim.*, 18:367–383.

Lee, J., Lee, I., and Lee, J. (2003). Unbiased global optimization of lennard-jones clusters for $n \leq 201$ using the conformational space annealing method. *Phys.Rev.Lett.*, 91(8):080201/1–4.

Liu, D. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45:503–528.

Locatelli, M. (2005). On the multilevel structure of global optimization problems. *Computational Optimization and Applications*, 30:5–22.

Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2002). Shape distributions. *ACM Trans. Graph.*, 21(4):807–832.

Peura, M. and Iivarinen, J. (1997). Efficiency of simple shape descriptors. *Aspects of Visual Form*, pages 443–451.

Preparata, F. P. and Shamos, M. I. (1985). *Computational geometry: an introduction*. Springer-Verlag New York, Inc., New York, NY, USA.

Ross, S. M. (1987). *Introduction to Probability and Statistics for Engineers and Scientists*. John Wiley & Sons.

Russel, S. and Norvig, P. (1995). *Artificial Intelligence, A modern approach*. Artificial Intelligence. Prentice Hall, 2 edition.

Schachinger, W., Addis, B., Bomze, I. M., and Schoen, F. (2007). New results for molecular formation under pairwise potential minimization. *Computational Optimization adn Applications*, To appear.

Veltkamp, R. C. (2001). Shape matching: similarity measures and algorithms. *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 188–197.

Veltkamp, R. C. and Hagedoorn, M. (1999). State-of-the-art in shape matching. External research report, Utrecht University: Information and Computing Sciences.

Wales, D., Doye, J. P. K., Dullweber, A., Hodges, M., Naumkin, F., Calvo, F., Hernandez-Rojas, J., and Middleton, T. (2007). The cambridge cluster database. `http://www-wales.ch.cam.ac.uk/CCD.html`.

Wales, D. J. and Doye, J. P. K. (1997). Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *J. Phys. Chem. A*, 101:5111–5116.

Author addresses:

Andrea Cassioli,
Dipartimento di Sistemi e Informatica
Università degli Studi di Firenze
via di Santa Marta, 3
50139 Firenze (Italy) E-mail: cassioli@dsi.unifi.it

Marco Locatelli,
Dipartimento di Informatica
Università di Torino Corso Svizzera, 185
10149 Torino
E-mail: locatell@di.unito.it

Fabio Schoen, (corresponding author)
Dipartimento di Sistemi e Informatica
Università degli Studi di Firenze
via di Santa Marta, 3
50139 Firenze (Italy)
E-mail: fabio.schoen@unifi.it