

Chapter 11

Global Optimization for Atomic Cluster Distance Geometry Problems

Marco Locatelli and Fabio Schoen

Abstract This chapter is devoted to a survey of global optimization methods suitable for the reconstruction of the three-dimensional conformation of atomic clusters based on the possibly noisy and imprecise knowledge of a sparse subset of pairwise distances. The problem we address is that of finding the geometry of a three-dimensional object without making use of any structural knowledge, but relying only on a subset of measured pairwise distances. The techniques we present are based on global optimization methods applied to different formulations of the problem. The methods are based on the use of standard local searches within a global optimization (GO) method which is based on local perturbation moves. Different definitions of these perturbations lead to different methods, whose performances are compared. Both sequential and population-based variants of the methods are reviewed in this chapter and some relevant numerical results are presented. From the evidence reported, it can be concluded that, when no additional information is available, such as, e.g., information about a linear order which allows for using a build-up technique, the methods proposed in this chapter represent an effective tool for difficult distance geometry problems (DGPs).

M. Locatelli
Università di Parma, viale G.P. Usberti 181/A, 43100 Parma, Italy
e-mail: locatelli@ce.unipr.it

F. Schoen (✉)
Università di Firenze, via di S. Marta 3, 50139 Firenze, Italy
e-mail: fabio.schoen@unifi.it

11.1 Introduction

In a *distance geometry problem* (DGP in what follows), given an incomplete subset of possibly noisy distance measures between N points, the aim is to reconstruct the configuration of the N points in the three-dimensional Euclidean space. More formally, we aim at determining the positions for N points $x^1, x^2, \dots, x^N \in \mathbb{R}^3$ so that, for a given subset of pairs \mathcal{D} and given bounds ℓ_{ij}, u_{ij} , the following distance constraints are satisfied:

$$\ell_{ij} \leq \|x^i - x^j\| \leq u_{ij} \quad \text{for all } \{i, j\} \in \mathcal{D}. \quad (11.1)$$

Note that if $x^{1*}, x^{2*}, \dots, x^{N*}$ is a solution to the problem, any configuration obtained by translating and/or rotating this solution also solves the problem. Therefore, from now on, we will always refer to solutions, modulo translations and rotations. Exact bounds (i.e., $\ell_{ij} = u_{ij}$ for all $\{i, j\} \in \mathcal{D}$) usually make the problem simpler. In fact, if exact bounds are given and \mathcal{D} is made up by all possible pairs of points, the problem is solvable in polynomial time $\mathcal{O}(N^3)$ through eigenvalue decomposition of the distance matrix. However, if not all distances are known and/or lower and upper bounds differ, the problem has been proved to be strongly NP-hard (see [6, 23]).

We would like to stress the fact that in this chapter we are assuming that no information is available, except for the partial list of distance measurements. Thus, even if for some numerical examples we got our data from fragments of proteins, we assume that no information on the linear ordering of residues in the protein is included. In other words, from the point of view of the approaches discussed here, all points (atom centers) are considered as indistinguishable one from the other. In addition, the methods we present here do not rely on pre-processing or other tools aimed at discovering partial structures with known geometry. As an example, when a linear ordering of atoms is known—this situation happens, e.g., when studying the problem applied to proteins—if, given the sequence of atoms, all pairwise distances are exactly known among all consecutive four-tuples of atoms, then it is easy to see that the problem, apart from a few degenerate cases which never happen in nature, is trivially solvable by elementary geometric arguments.

Thus the approaches we present here are quite more general than those based on geometric build-up strategies (see, e.g., [7]) and can be applied to even widely different conformational problems. On the other hand, when compared with methods which are strongly based on such a knowledge, our methods clearly lag behind.

Many geometrical problems where points have to be placed in the two- or three-dimensional space in such a way that some constraints are satisfied and/or some objective function is minimized or maximized, have been tackled by global optimization (GO in what follows) techniques. They include, for instance, molecular conformation problems (such as Lennard–Jones and Morse clusters, see, e.g., [8, 9, 15, 16, 18, 25]) or packing problems (see, e.g., [1]).

The quite promising results obtained for such problems suggest that also the DGP can be effectively tackled by similar GO approaches. For this reason, in this chapter, we will focus our attention on GO approaches for the DGP. We point out, however, that the DGP has been tackled also by many other different approaches in the literature as it can be seen browsing the different chapters of this volume—thus we do not review them in this chapter, referring the reader to the other sections in this volume.

Problem (11.1) turns out to be a nonlinear, non-convex feasibility problem. Given $X = (x^1, x^2, \dots, x^N) \in \mathbb{R}^{3N}$, we can define the relative error function

$$E_r(X) = \sum_{\{i,j\} \in \mathcal{D}} \left[\max^2 \left(\frac{\ell_{ij}^2 - \|x^i - x^j\|^2}{\ell_{ij}^2}, 0 \right) + \max^2 \left(\frac{\|x^i - x^j\|^2 - u_{ij}^2}{u_{ij}^2}, 0 \right) \right]. \quad (11.2)$$

Solutions to Eq. (11.1) (if any) coincide with global minimizers of the unconstrained GO problem whose objective function is Eq. (11.2) (the objective function value in any solution is 0). Similarly, one can use the absolute error function

$$E_a(X) = \sum_{\{i,j\} \in \mathcal{D}} \left[\max^2 (\ell_{ij}^2 - \|x^i - x^j\|^2, 0) + \max^2 (\|x^i - x^j\|^2 - u_{ij}^2, 0) \right]. \quad (11.3)$$

We remark that both functions E_a and E_r are smooth, which allows for the use of efficient local minimization procedures in search algorithms.

This chapter is structured as follows. In Sect. 11.2 we will outline solution approaches based on multiple local searches; in Sect. 11.3 we introduce the test instances which will be employed throughout the chapter; in Sect. 11.4 we will make some preliminary observations based on some computational experiments. In Sect. 11.5 the basic approach will be extended to a population-based version. Finally, in Sect. 11.6, we will draw some conclusions and indicate some possible directions for future research.

11.2 GO Approaches: DGSOL and Basin Hopping

An interesting approach based on the GO reformulation of the DGP was the DGSOL algorithm proposed by Moré and Wu [21, 22]. DGSOL is based on *global*

continuation techniques, and works by successive minimizations of smoothed functions $\langle E_a(X) \rangle_\lambda$, where

$$\langle E_a(X) \rangle_\lambda = \frac{1}{\pi^{3N/2} \lambda^{3N}} \int_{\mathbb{R}^{3N}} E_a(Y) \exp\left(-\frac{\|Y-X\|^2}{\lambda^2}\right) dY$$

is obtained by convolution with a Gaussian function for decreasing values of the parameter $\lambda \geq 0$ ($\langle E_a(X) \rangle_\lambda$ converges to E_a). The effect of a large λ value is that of filtering the oscillations of the original function E_a out, thus making the resulting function a convex one. Then, the λ value is slowly reduced, and the oscillations are gradually restored until we are back to the original function when $\lambda = 0$. By decreasing λ and starting a local search from the previously detected local minimizer, we may hope to jump over the oscillations separating the local minimizers until we reach the global minimizer.

The idea of filtering the oscillations out is also the basis of the approach proposed here. Following an approach already employed in the field of molecular conformation problems (see, e.g., [24]) we can modify the original function E_a as follows:

$$E'_a(X) = E_a(\mathcal{L}\mathcal{S}(X)),$$

where $\mathcal{L}\mathcal{S}$ is a local search procedure. The role played by the parameter λ in DGSOL (filtering the oscillations out) is played here by the local search procedure. However, here we need to specify how to escape from a local minimizer. Indeed, each local minimizer lies in a flat region of the function E'_a , and we need a way to escape from this region and reach a different local minimizer. In the approach proposed in this chapter, this is accomplished through the definition of a neighborhood structure between local minimizers. Let us denote such structure by \mathcal{N} . Then, following the terminology in [6], (see also [19]) a *funnel* \mathcal{F} can be defined as a maximal set of local minimizers such that for each $X \in \mathcal{F}$, there exists at least one decreasing sequence of neighbor local minimizers

$$X_0 = X \rightarrow X_1 \rightarrow \dots \rightarrow X_t = X^*$$

$$E_a(X_i) < E_a(X_{i-1}), \quad X_i \in \mathcal{N}(X_{i-1}) \quad i = 1, \dots, t$$

starting at X and ending at a common local minimizer X^* , called the *funnel bottom*. Note that the global minimizer is always a funnel bottom, independently of the neighborhood structure \mathcal{N} . The neighborhood must possess two properties (sometimes conflicting with each other):

- It should have a manageable size (in particular, it should be much smaller than the whole set of local minimizers).
- It should be such that the corresponding number of funnel bottoms is much lower than the overall number of local minimizers (in the easiest cases, the function E_a has a single funnel, whose funnel bottom is the global minimizer, in spite of the huge number of local minimizers).

Algorithm 5 The MBH algorithm with local move Φ

```

1: Let  $X_0$  be an initial local minimizer,  $k = 0$ 
2: while a stopping rule is not satisfied do
3:   Let  $Y_{k+1} = \Phi(X_k)$ 
4:   if  $E_a(Y_{k+1}) < E_a(X_k)$  then
5:     set  $X_{k+1} = Y_{k+1}$ 
6:   end if
7:   set  $k = k + 1$ 
8: end while

```

Once the notions of neighborhood structure, funnel and funnel bottom have been introduced, we need some effective algorithm to explore funnels, and detect funnel bottoms. The algorithm that we are going to describe here is called monotonic basin hopping (MBH) algorithm, introduced in [16, 24] in the context of molecular conformation problems. The key operation of this algorithm is the *local move* Φ which, for a given local minimizer X , returns a local minimizer in its neighborhood, i.e.,

$$\Phi(X) \in \mathcal{N}(X).$$

Algorithm 5 is a sketch of the algorithm. The algorithm stops either when $E_a(X_k) = 0$ (in such case a solution has been found), or when X_k has not changed for a prefixed number of iterations denoted by `MAXITER`. The choice of the local move, and hence the choice of the neighborhood it explores, is essential for the performance of MBH (all the ingenuity of the method, which is otherwise extremely simple, lies in the choice of the local move). An indicative rule for its choice is that the local move should generate a new local minimizer but without completely disrupting the structure of the current one X (the principle is very similar to that of iterated local searches in combinatorial optimization problems, see [20]). In case the function has more than one funnel bottom, it may be necessary to run MBH more than once (in a multistart fashion) before detecting the global minimizer. For the DGP we propose here two different local moves. One move is quite general and the other more problem-specific.

11.2.1 Local Move $\Phi_{1,\Delta}$

The first local move that we tested is an extremely simple one and is basically the same employed for molecular conformation problems (see [8, 16, 18, 24]). Given a local minimizer X , a local search procedure $\mathcal{L}\mathcal{S}$, and some $\Delta > 0$, we set

$$\Phi_{1,\Delta}(X) = \mathcal{L}\mathcal{S}(X + \delta),$$

where δ is randomly sampled within the box $[-\Delta, \Delta]^{3N}$. Basically, we randomly perturb the current local minimizer, by controlling the size of the perturbation through Δ , and then we start a local search from the perturbed point. The choice of the parameter Δ is not a trivial one. Driven by the observation for which the local move should, at least partially, keep the structure of the original configuration, we first tried to consider small values for Δ . In particular, in the experiments reported later, we tried to set $\Delta = 0.05D$, where D represents the length of the edges of a box surely containing the final configuration (in all the experiments, D was chosen to be 100). However, the experiments revealed that such value is too small: when starting a local search from the point returned by this perturbation, very often the result was the original configuration, i.e., such perturbation often turned out to belong to the region of attraction of the current minimizer and, in fact, the local move did not make any move at all. For this reason, we increased the value up to $\Delta = 0.5D$. At a first glance, such perturbation appears to be too large (recall that all molecules in our test instances lie within a box with edge size equal to $D = 100$). In order to verify the behavior of the perturbations, we produced the histograms representing the distances between corresponding points (after appropriate rotations and translations) in both the initial configuration X and the perturbed one $\Phi_{1,\Delta}(X)$. Such histograms were often concentrated around small distances. Therefore, it seems that, in spite of the large Δ value, the local search procedure is able to drive back the perturbed point towards a local minimizer close to the original one. For the sake of completeness we also tested a larger perturbation size $\Delta = 5D$. As expected, in this case, the histograms were usually not concentrated around small distances but more spread, i.e., the final configuration $\Phi_{1,\Delta}(X)$ appeared to be only mildly related with the initial one X .

As a final remark, it is important to point out that, while we selected a priori a single value for Δ , a more reasonable approach is that of using some adaptive rule to update the value of Δ according to the outcomes of the different iterations of the algorithm, so that the dependency of such value from the problem at hand can be taken into account.

11.2.2 Local Move $\Phi_{2,\Gamma}$

In the move described in Sect. 11.2.1, the only way to include some a priori knowledge about the problem and some information collected by the algorithm is through the parameter Δ . In fact, it may be a good idea to exploit more both information. This can be accomplished through a problem-specific local move, as the one that we are going to discuss. While defining a perturbation on the current configuration, we are supposed to take into account which of the distances in \mathcal{D} are currently violated. Therefore, one possible idea is to consider for each point r the most violated distance in \mathcal{D} within the current configuration X_k , and then to move the point r in a direction which should reduce the violation of such distance (the point r is not moved at all if no violation involving the point r occurs within the

current configuration). Even though quite reasonable, this strategy did not deliver good results. However, a strategy based on the same idea turned out to be very effective. In the new strategy, we considered the current configuration X_k , as well as another configuration, and the perturbation is based on both. Instead of considering the distances of a point r within the current configuration with respect to the other points in the same configuration, we considered the distances of a point r with respect to points in the other configuration. More precisely, we first introduced the local minimizer Z_k

$$Z_k = \arg \max \{E_a(X_{k-1}), E_a(Y_k)\}, \quad (11.4)$$

i.e., Z_k is equal to the last discarded point, which is $Y_k = \Phi(X_{k-1})$, if Y_k has been rejected at iteration $k - 1$, otherwise it is equal to X_{k-1} . Then, as previously commented, for each point $r = 1, \dots, N$, we take the point $s(r)$ which mostly violates one of the distances in \mathcal{D} involving the point r , when the point r is taken from X_k , while the other points s are taken from Z_k , i.e.,

$$s(r) \in \arg \max_{s:(r,s) \in \mathcal{D}} \{0, \|x_k^r - z_k^s\| - u_{rs}, \ell_{rs} - \|x_k^r - z_k^s\|\}.$$

The corresponding maximum distance is denoted by $\eta(r)$. Then, we try to move each point r in such a way that the value $\eta(r)$ is reduced by moving r towards the point $s(r)$ in Z_k (if we have a violation of the upper bound) and far from $s(r)$ in Z_k (if we have a violation of the lower bound). More formally, we introduce a direction vector $D_k = (d_k^1, \dots, d_k^N)$ whose r th coordinate is defined as follows:

$$d_k^r = \begin{cases} 0 & \text{if } \eta(r) = 0 \\ -(x_k^r - z_k^{s(r)}) & \text{if } \eta(r) = \|x_k^r - z_k^{s(r)}\| - u_{rs(r)}. \\ (x_k^r - z_k^{s(r)}) & \text{otherwise} \end{cases}$$

Finally, we perturb the current configuration along direction D_k and start a local search from the perturbed point, i.e., we define the local move as follows:

$$\Phi_{2,\Gamma}(X) = \mathcal{L}\mathcal{S}(X + \gamma D_k),$$

where γ is randomly sampled in the interval $[0, \Gamma]$ (in our tests we fixed $\Gamma = 4$).

11.3 Test Instances

One of the contexts from which DGPs arise is molecular chemistry, where one aims at determining the geometric structures of large molecules (e.g., proteins) from NMR experiments, which usually only deliver a subset of pairwise atom distances, and from X-ray diffraction measurements on crystallized proteins. In such a context, the problem is known as molecular DGP (MDGP), and each point in $\{1, \dots, N\}$

represents (the center of) an atom. All test instances in this chapter have been generated from data from the protein data bank (PDB, see [2]); we considered two classes of instances. First of all, we considered some instances from the work by Moré and Wu [22]; such instances have been generated from the 1_{GFV} molecule in the PDB. Moré and Wu took 100-atom and 200-atom fragments, and generated distance data for pairs in successive residues R_k, R_{k+1} :

$$\mathcal{D} = \{\{i, j\} : i \in R_k, j \in R_{k+1}\},$$

setting the bounds to

$$\ell_{ij} = (1 - \varepsilon)d_{ij}, \quad u_{ij} = (1 + \varepsilon)d_{ij}, \quad \{i, j\} \in \mathcal{D},$$

with ε set to values from 0.04 to 0.16, with $\varepsilon = 0.04$ yielding the hardest instances. We considered two 200-atom instances—here called DNA200a and DNA200b—with $\varepsilon = 0.04$ provided with the DGSOL package.¹

We then generated instances of the MDGP by a technique similar to that used in [3] for the exact MDGP. We selected a number of molecules from the PDB for various sizes N , and for each of them we kept in \mathcal{D} only the pairs with interatomic distances d_{ij} below the cutoff value $R = 6 \text{ \AA}$; then we set

$$\ell_{ij} = (1 - \varepsilon)d_{ij}, \quad u_{ij} = (1 + \varepsilon)d_{ij}, \quad \{i, j\} \in \mathcal{D}$$

with ε randomly set in the interval $(0, 0.04)$. The choice of the interval is related to the observations given in [22], where bounds $\ell_{ij} = (1 - 0.04)d_{ij}$, $u_{ij} = (1 + 0.04)d_{ij}$ are claimed to give difficult instances—indeed, computational experience not discussed in the remainder of this chapter confirms that for larger ε the resulting instances are quite easy. It is important to remark that usually, if the sequence of residuals is explicitly taken into account, test problems obtained by using a cutoff distance of 6 \AA are relatively easy to solve by geometrical arguments or by the approaches proposed by [7, 14]. Here we stress the fact that our approach does not use information on the linear sequence of amino acids. One might argue that disregarding this information makes the problem unnecessarily hard; however, from one side there are many cases in which no linear ordering is given, e.g., the case of clusters of atoms. Also, there may be situations in which the protein being analyzed has an unknown primary structure, and the aim of the NMR observation is exactly that of obtaining both the primary and the tertiary (i.e., 3D) conformation of the molecule. In these cases atoms are considered as undistinguishable, and a GO approach seems to be the best alternative available. Some properties of the tested

¹Available at www.mcs.anl.gov/~more/dgsol.

Table 11.1 Instances used for testing

Name	N	Density (%)
DNA200a	200	1.7
DNA200b	200	16.5
1PTQ	402	8.8
1HOE	558	6.5
1LFB	641	5.6
1PHT	814	5.3
1POA	914	4.1
1AX8	1,003	3.7

instances are reported in Table 11.1, where the density parameter is the fraction of known distances with respect to all distances, i.e., it is equal to

$$2|\mathcal{D}|/[N(N-1)].$$

11.4 Preliminary Observations on Numerical Experiments

In our first set of preliminary computational experiments, we aimed at testing the behavior of multiple local searches started from randomly generated initial points. Basically, we ran a multistart approach, where a given number of local searches (1,000 in our experiments) is started from randomly sampled points over a $3N$ -dimensional boxes, i.e., $D = [-R, R]^{3N}$, for some $R > 0$. We point out that, here and in what follows, local searches have always been performed through limited memory BFGS [17]. Of course, multistart is a rather inefficient approach for highly multimodal GO problems, but its simplicity allows to analyze the behavior of local searches without being influenced by other components of the solution approach. Two interesting observations can be drawn from the experiments.

The first observation is that a relevant role is played by the size of the initial box. When using the objective function (11.3), the number of times a solution was detected for the instances presented in Table 11.1 over 1,000 local searches was clearly superior for a large R value (namely, $R = 5,000$) with respect to a small R value (namely, $R = 50$).² A possible explanation for this fact is the following. When we generate the initial configuration in a very large box, most bounds for the distances in \mathcal{D} are strongly violated. Therefore, in function (11.3), strong *attractive* forces come into play, pushing the points, initially spread around, much closer to each other. According to the experiments, it seems that the attractive forces are able to drive the local search procedure in such a way that it jumps over local minimizers with high function values and reaches more

²Note that, from the set of known distances, it is possible to guarantee that a box of edge size equal to 100 is able to enclose all the molecules for the tested instances.

easily local minimizers with low function value (if not even the global minimizer). An analogous phenomenon has been observed in [15] for so-called Big-Bang algorithm for Lennard–Jones clusters. That algorithm works by generating an initial configuration of atoms in a Lennard–Jones cluster in a very narrow region and then starting a local search from such configuration. The strong *repulsive* forces which are active within the narrow region allow to spread the atoms during the local search in such a way that the detection of good local minimizers is considerably simplified.

The second interesting observation is related to the choice of the objective function. From the theoretical point of view, there is no difference between the functions (11.2) and (11.3): in both cases, a global minimizer with function value equal to 0 is a solution of our problem. However, there are clear differences from the computational point of view. By considering the same R value for both functions, the number of successes with the objective function (11.2) is clearly inferior with respect to the number of successes with the objective function (11.3). Just to cite a single significant result, with $R = 5,000$ and test PTQ, we have 49 successes out of 1,000 local searches with the function (11.3) and 0 successes with the function (11.2). We point out that the superiority of the function (11.3) has not only been observed with the multistart algorithm but also with all the other methods tested in this chapter. Thus, from now on we will always employ the function (11.3). A possible explanation for the superiority of the absolute value function with respect to the relative one is the following. Obviously, the larger the deviation from the given bounds, the deeper is the required modification of the current configuration. This is taken into account in the absolute error function but might not emerge in the relative error function. Indeed, if a large deviation occurs for a distance whose bounds are also large, the corresponding term in the relative error function is of comparable size with respect to a small deviation for a distance whose bounds are small ones. Thus, decreasing the second error has the same impact, from the point of view of the relative error function, with respect to decreasing the first error, but, from the point of view of the final configuration, the modification in the configuration when trying to reduce the second error is considerably less relevant with respect to the modification when trying to reduce the first error.

11.4.1 Computational Results for MBH

In Table 11.2, we report the experimental results over the test instances for MBH with the two local moves described above ($\Phi_{1,\Delta}$ with $\Delta = 50$, and $\Phi_{2,\Gamma}$ with $\Gamma = 4$). The results are taken from [11] and reported here for sake of completeness. The observation of the results in the table allows us to draw some interesting conclusions. The first one is that MBH with both local moves has a high number of successes over 10 runs (see columns succ). This means that both neighborhood structures

Table 11.2 Number of successes (columns SUCC) over 10 runs, overall average number of local searches per success (columns NSavg), and average number of local searches in successful runs (columns NSavg-succ) for MBH with the two local moves $\Phi_{1,\Delta}$ and $\Phi_{2,\Gamma}$

Instance	$\Phi_{1,\Delta}$			$\Phi_{2,\Gamma}$		
	Succ	NSavg	NSavg-succ	Succ	NSavg	NSavg-succ
DNA200a	10	151	151	10	39	39
DNA200b	10	24	24	10	20	20
PTQ	10	15	15	7	217	3
HOE	10	17	17	8	139	14
LFB	10	55	55	8	192	67
PHT	10	55	55	8	136	11
POA	9	180	124	8	152	27
AX8	10	107	107	9	74	18
GPV	4	888	138	6	470	137

have a quite limited number of funnels (or, at least, the funnel corresponding to the global minimizer is extremely large with respect to other funnels). Another interesting observation arises from the comparison of the average number of local searches per success (columns NSavg), and the average number of local searches in successful runs (columns NSavg-succ). We could expect that the problem-specific local move $\Phi_{2,\Gamma}$ dominates the general local move $\Phi_{1,\Delta}$. In fact, this is not always true. Especially over the smaller instances, the number of successes and the average number of local searches per success for the local move $\Phi_{1,\Delta}$ are better than that of $\Phi_{2,\Gamma}$. But if we look at the average number of local searches only in the successful runs we observe that $\Phi_{2,\Gamma}$ is almost always better (and usually much better) than $\Phi_{1,\Delta}$. Basically, the problem-specific local move $\Phi_{2,\Gamma}$ reaches more easily a funnel bottom, but most likely such a funnel bottom is not a global minimizer. In such case, before stopping MBH, we need to pay the MAXITER local searches (500 in our tests) which have to be added each time a failure occurs. On the other hand, when starting within the funnel corresponding to the global minimizer, the local move $\Phi_{2,\Gamma}$ usually reaches the global minimizer (much) faster than the local move $\Phi_{1,\Delta}$. Of course, one could overcome this difficulty if we could choose MAXITER as small as possible. Unfortunately, the selection of an appropriate value for MAXITER is not a trivial task: as suggested by the large variability of the NSavg-succ values in Table 11.2, the appropriate value might be quite different from instance to instance.

11.5 Population Basin Hopping

The experimental results discussed above for MBH show that it would be important to recognize as soon as possible when a run of MBH leads to a failure. This is true in general and is especially true for $\Phi_{2,\Gamma}$, where the NSavg-succ values reveal

that, when starting from a good point, the effort to reach a global minimizer by MBH is quite limited. As commented above, we should choose MAXITER as small as possible, but it is not clear at all how to choose it: if the value is too large, it causes an undesired computational waste, otherwise it could prejudice the ability of reaching the funnel bottom even when starting in the funnel corresponding to the global minimizer.

A possible way to overcome this difficulty is to substitute K trajectories which are independently and sequentially followed by K runs of MBH with K different trajectories which are followed in parallel (and, possibly, not independently but with some information exchange) within a population-based approach. This obviously increases not only the effort per iteration (by a factor of K) but also the probability of success, and, above all, the total number of iterations to reach the global minimizer is determined by the shortest of the K trajectories, which partially counterbalances the larger effort per iteration. Note that we still have to choose a parameter (the number K of trajectories), but a good choice for it appears to be less variable from instance to instance (in our tests, we always fixed $K = 10$).

The easiest way to follow K trajectories is to run K parallel independent runs of MBH with no information exchange between them. However, some more ingenuity can be introduced in the algorithm by allowing for some kind of information exchange. This results in the population basin hopping (PBH) algorithm already tested in [9] on cluster optimization problems. The main elements of this algorithm are a local move Φ as in MBH, and a new relevant component, the dissimilarity measure d between local minimizers (see Algorithm 6 for a sketch).

Algorithm 6 The PBH algorithm with local move Φ and dissimilarity measure d

```

1: Let  $\{X_0^1, \dots, X_0^N\}$  be an initial set of  $N$  local minimizers,  $k = 0$ 
2: while a stopping rule is not satisfied do
3:   Let  $Y_i = \Phi(X_k^i)$ ,  $i = 1, \dots, N$ 
4:   for  $j \leftarrow 1$  to  $N$  do
5:     Let  $i(j) \in \arg \min_{i=1, \dots, N} d(Y_j, X_k^i)$ 
6:     if  $E_a(Y_j) < E_a(X_k^{i(j)})$  then
7:       set  $X_{k+1}^{i(j)} = Y_j$ 
8:     end if
9:   end for
10:  set  $k = k + 1$ 
11: end while

```

Once again, we stop either when $E_a(X_k^i) = 0$ for some member of the population, or when we reach a prefixed number of iterations (in fact, in our tests only the former rule came into play). In the PBH algorithm, at each iteration k , the population $\{X_k^1, \dots, X_k^N\}$ is available. A set $\{Y_1, \dots, Y_N\}$ of candidate points, where $Y_j = \Phi(X_k^j)$ for $j = 1, \dots, N$, is generated. Then, the candidate point Y_j is not necessarily compared with X_k^j but with the member $X_k^{i(j)}$ of the population which is less dissimilar to Y_j . Dissimilarity is measured by some function d . If we define, e.g.,

$d(Y_j, X_k^i) = |j - i|$, then obviously we always have $i(j) = j$, and Algorithm 6 is equivalent to N independent runs of Algorithm 5. However, other choices for d are possible. Ideally, d should be very close to 0 if two local minimizers have a high probability of lying in the same funnel, while it should increase if such probability decreases. If this were the case, members of a funnel would tend to be compared only with members of the same funnel, so that d acts as a diversification tool which allows to explore different funnels and avoids the multiple exploration of the same one. As extensively discussed in [4, 9, 10], dissimilarity measures allow for communication and collaboration among members of the population. When no communication and collaboration occurs, PBH simply follows K independent trajectories, and the only difference with respect to K independent MBH runs is that the latter are run sequentially, while the former are followed in parallel. While this is usually a minor difference for the DGP, such difference has a relevant effect. Indeed, in the DGP we know in advance the global optimum value (which is equal to 0) so that, as soon as a configuration with objective function value equal to 0 is reached, we can stop the search along *all* trajectories. In some sense, this is a form of communication and collaboration because as soon as one trajectory reaches a global minimizer, it immediately communicates it to the other ones, thus avoiding unnecessary computational waste. But if we use more sophisticated dissimilarity measures, also other advantages may come into play. In general, a good dissimilarity measure allows to counterbalance the greedy tendency of MBH (too fast convergence to a funnel bottom not corresponding to a global minimizer) leading to a much greater efficiency with respect to the case of K independent trajectories, as observed in [9] and in the related experimental analysis in [10]. This usually happens when many funnel bottoms exist and/or the funnel bottom corresponding to the global minimizer can be reached through trajectories of considerably different lengths. In fact, for the DGP we cannot claim at the moment that we have found a dissimilarity measure which brings the above-mentioned advantages. After testing a few dissimilarity measures, we restricted our attention to a very general and simple one, the absolute value of the difference between function values:

$$d(X, Y) = |E_a(X) - E_a(Y)|. \quad (11.5)$$

The results we obtained with this measure (reported in Table 11.3) are good ones but not much better than those which can be obtained by following independent trajectories in parallel. However, a positive effect of collaboration and communication between members of the population will be discussed at the end of the section.

Before discussing the results, we just remark a small difference in the definition of Z_k with respect to Eq. (11.4) for the local move $\Phi_{2,\Gamma}$. In PBH we define a point Z_k^j for each member j of the population as follows:

$$Z_k^j = \arg \max \{E_a(X_{k-1}^{i(j)}), E_a(Y_k^j)\},$$

Table 11.3 Number of successes (columns SUCC) over 10 runs and average number of local searches per success (columns NSavg) for PBH with the two local moves $\Phi_{1,\Delta}$ and $\Phi_{2,\Gamma}$

Instance	$\Phi_{1,\Delta}$		$\Phi_{2,\Gamma}$	
	Succ	NSavg	Succ	NSavg
DNA200a	10	116	10	143
DNA200b	10	30	10	23
PTQ	10	57	10	33
HOE	10	67	10	53
LFB	10	192	10	89
PHT	10	162	10	63
POA	10	234	10	88
AX8	10	263	10	87
GPV	10	949	10	312

i.e., Z_k^j is the loser of the last competition between the new candidate point $Y_k^j = \Phi(X_{k-1}^j)$ and its competitor $X_{k-1}^{i(j)}$, the member of the population at iteration $k-1$ less dissimilar to Y_k^j . The results with both local moves $\Phi_{1,\Delta}$ and $\Phi_{2,\Gamma}$ are reported in Table 11.3. As expected from the results with MBH, a population size $K = 10$ was already enough to reach 100% of successes on all test instances. Moreover, as expected from the values in Columns NSavg-succ in Table 11.2, we observe that better results are usually obtained with the problem-specific move $\Phi_{2,\Gamma}$ with respect to $\Phi_{1,\Delta}$.

As previously mentioned, while the dissimilarity measures tested up to now do not give impressive improvements with respect to following in parallel K independent trajectories, we could in fact observe a positive effect of the dissimilarity measure. We worked in order to identify a proper value for the parameter Γ in the local move $\Phi_{2,\Gamma}$. At the beginning, we tried $\Gamma = 2$, and this choice turned out to be a bad one. Indeed, with the neighborhood structure corresponding to this choice, a large number of funnels were created. Both MBH and PBH with $K = 10$ had a very large number of failures in this case. We could get back to 100% successes only after enlarging the population size to $K = 40$. We analyzed the behavior of PBH during these runs with population size $K = 40$, and we realized that the phenomenon of survival, one of the two phenomena (the other is backtracking) which, according to the analysis in [9, 10], is responsible for the success of PBH, had a great positive impact on the final results. Survival occurs when a member X_k^j of the population generates a better child Y_{k+1}^j , i.e., $E_a(Y_{k+1}^j) < E_a(X_k^j)$, but $i(j) \neq j$ so that X_k^j will still survive in the next population. This phenomenon counterbalances the greedy tendency of MBH (in MBH X_k^j would be simply replaced by Y_{k+1}^j). Such tendency might cause too fast convergence to funnel bottoms not corresponding to the global minimizer, while survival allows to keep within the population local minimizers from which it is still possible to reach the global minimizer. Therefore, although overall the performance of PBH with $\Gamma = 2$ and $K = 40$ is inferior to that with $\Gamma = 4$ and $K = 10$, the collaboration and communication between members of the population which takes place through the dissimilarity measure is able to considerably reduce the negative effects of a wrong choice of the parameter Γ defining the local move.

11.6 Possible Directions for Future Research

In this chapter, we considered a reformulation of the DGP as a GO problem and tackled it with some approaches (MBH and its population-based version PBH) which already turned out to be particularly suitable for geometrical problems which can be reformulated as GO ones, such as molecular conformation and packing problems. The results of our computational experiments suggest that these are indeed promising approaches to tackle DGPs. However, in this field, there are still many directions which could be explored. Here we mention a few of them.

- We reformulated the DGP as an unconstrained GO problem with objective function (11.2) or (11.3). In Sect. 11.4 we observed that the performance of any GO algorithm can be considerably different when using Eq. (11.2) or (11.3). In particular, Eq. (11.3) appears to be much better (and we gave a tentative explanation of this fact). We might wonder if other (theoretically) equivalent reformulations of the DGP are possible which turn out to be more efficient than Eq. (11.3). For instance, for any increasing one-dimensional function f , the DGP is equivalent to the unconstrained minimization of $f(E_a(X))$.
- When dealing with very large instances, the CPU time required by local searches may be very large. A good idea in this case could be that of decomposing the problem: different and mildly correlated (within the subset \mathcal{D} of distances) parts are separately optimized, and then some technique is employed to merge together the different parts (the idea of decomposing solutions was also exploited in the ABBIE approach [12] and in the SDP approach proposed in [3]).
- In this chapter, we have only proposed two local moves, a general and a problem-specific one. Surely, the geometrical nature of the problem might suggest further local moves. For instance, one could think about moves like the surface-repair ones employed in [5] or atom relocation techniques like those used in [13] for molecular conformation problems. For DGP, one might try to localize the moves in such a way that only points which are involved in violated distances in \mathcal{D} are moved (in fact, this is already in the spirit of the local move $\Phi_{2,r}$).
- There is space to find more effective dissimilarity measures. At the moment, we have not found any which really enhances the performance of the proposed PBH approach (although, as we observed, the proposed one might be helpful in order to counterbalance bad choices of the parameters).

References

1. Addis, B., Locatelli, M., Schoen, F.: Efficiently packing unequal disks in a circle. *Oper. Res. Lett.* **36**, 37–42 (2008)
2. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. *Nucleic Acids Res.* **28**, 235–242 (2000)

3. Biswas, P., Toh, K.-C., Ye, Y.: A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM J. Sci. Comput.* **30** 1251–1277 (2008)
4. Cassioli, A., Locatelli, M., Schoen, F.: Dissimilarity measures for population-based global optimization algorithms. *Comput. Optim. Appl.* **45**, 257–281 (2010)
5. Cheng, L., Feng, Y., Yang, J., Yang, J.: Funnel hopping: Searching the cluster potential energy surface over the funnels. *J. Chem. Phys.* **130**(21), 214112 (2009)
6. Crippen, G., Havel, T.: *Distance Geometry and Molecular Conformation*. Wiley (1988)
7. Dong, Q., Wu, Z.: A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *J. Global. Optim.* **26**, 321–333 (2003)
8. Doye, J.P.K., Leary, R.H., Locatelli, M., Schoen, F.: The global optimization of Morse clusters by potential transformations. *INFORMS Journal on Computing* **16**, 371–379 (2004)
9. Grosso, A., Locatelli, M., Schoen, F.: A population based approach for hard global optimization problems based on dissimilarity measures. *Math. Program.* **110**, 373–404 (2007)
10. Grosso, A., Locatelli, M., Schoen, F.: An experimental analysis of population based approach for global optimization. *Comput. Optim. Appl.* **38**, 351–370 (2007)
11. Grosso, A., Locatelli, M., Schoen, F.: Solving molecular distance geometry problems by global optimization algorithms. *Comput. Optim. Appl.* **43**, 23–37 (2009)
12. Hendrickson, B.A.: *The molecular problem: determining conformation from pairwise distances*. Ph.D. Thesis, Cornell University (1991)
13. Lai, X., Xu, R., Huang, W.: Prediction of the lowest energy configuration for Lennard-Jones clusters. *Sci. China Chem.* **54**, 985–991 (2011)
14. Lavor, C., Liberti, L., Maculan, N., Mucherino, A.: Recent advances on the discretizable molecular distance geometry problem. *Eur. J. Oper. Res.* **219**, 698–706 (2012)
15. Leary, R.H.: Global optima of Lennard-Jones clusters. *J. Global. Optim.* **11**, 35–53 (1997)
16. Leary, R.H.: Global optimization on funneling landscapes. *J. Global. Optim.* **18**, 367–383, 2000.
17. Liu, D.C., Nocedal, J.: On the Limited Memory Method for Large Scale Optimization. *Math. Program.* **B45**, 503–528 (1989)
18. Locatelli, M., Schoen, F.: Efficient algorithms for large scale global optimization: Lennard-Jones clusters. *Comput. Optim. Appl.* **26**, 173–190 (2003)
19. Locatelli, M.: On the multilevel structure of global optimization problems. *Comput. Optim. Appl.* **30**, 5–22 (2005)
20. Lourenço, H.R., Martin, O., Stützle, T.: Iterated Local Search. In: Gendreau, M., Potvin, J.-Y. (eds.) *Handbook of Metaheuristics*, pp. 320–353. Springer (2003)
21. Moré, J., Wu, Z.: Global continuation for distance geometry problems. *SIAM J. Optim.* **7**, 814–836 (1997)
22. Moré, J., Wu, Z.: Distance geometry optimization for protein structures. *J. Global. Optim.* **15**, 219–234 (1999)
23. Saxe, J.B.: Embeddability of graphs in k -space is strongly NP-hard. In: *Proceedings of the 17th Allerton Conference in Communication, Control and Computing*, pp. 480–489 (1979)
24. Wales, D.J., Doye, J.P.K.: Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *J. Phys. Chem.* **A101**, 5111–5116 (1997)
25. Wales, D.J.: *Energy Landscapes with Applications to Clusters, Biomolecules and Glasses*. Cambridge University Press, Cambridge (2003)