2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Memetic Hybrid Method for the Molecular Distance Geometry Problem with Incomplete Information

M.S. Nobile, A.G. Citrolo, P. Cazzaniga, D. Besozzi, G. Mauri

***Abstract*— The definition of computational methodologies for the inference of molecular structural information plays a relevant role in disciplines as drug discovery and metabolic engineering, since the functionality of a biochemical molecule is determined by its three-dimensional structure. In this work, we present an automatic methodology to solve the Molecular Distance Geometry Problem, that is, to determine the best three-dimensional shape that satisfies a given set of target inter-atomic distances. In particular, our method is designed to cope with incomplete distance information derived from Nuclear Magnetic Resonance measurements. To tackle this problem, that is known to be NP-hard, we present a memetic method that combines two soft-computing algorithms – Particle Swarm Optimization and Genetic Algorithms – with a local search approach, to improve the effectiveness of the crossover mechanism. We show the validity of our method on a set of reference molecules with a length ranging from 402 to 1003 atoms.**

I. Introduction

THE determination of molecular structures is of great interest both in Chemistry and Biology, since the three-dimensional (3D) shape of a molecule is the main determinant of its function. In many contexts, such as drug discovery, metabolic engineering and catalysis, structural information is essential to understand and control the behavior of a molecular system. The great majority of structural data available today arise from two experimental techniques: X-ray crystallography and Nuclear Magnetic Resonance (NMR) [1]. NMR exploits the magnetic properties of the nucleus of isotopes (as ^{1}H, ^{13}C and ^{31}P) to identify spatial neighborhood relationships between chemical groups, which are generally given in the form of a matrix of inter-atomic distances. When this technique is applied to molecules of significant size and with a complex 3D shape, the resulting distance matrix is both sparse and noisy due to technical limitations of NMR. This is exactly the case of proteins, an ubiquitous class of biological molecules characterized by a great variability in shape and size. The *Molecular Distance Geometry Problem* (MDGP) consists in reconstructing the 3D structure of a molecule starting from its (sparse) distance matrix; the MDGP problem is a special case of the Distance Geometry Problem (DGP) [2] in which the distance matrix is obtained from NMR analysis. The peculiarity of the MDGP relies on the availability of additional constraints on the inter-atomic distances in the 3D structure, which can be defined according to the chemical and physical properties of the class of molecules under investigation.

In the case of incomplete information in the distance matrix, the MDGP was shown to be NP-hard [3] by reducing a 1-dimensional MDGP to the SUBSETSUM problem [4]. Several different approaches to the MDGP have been proposed in recent years, but they all suffer from limitations. For instance, the geometric buildup [5] is unable to find a solution to the problem for some cases of sparse distance matrices; the branch and prune algorithm [6], [7] has an exponential computational time; ABBIE [8], EMBED [9] and DGSOL [10] algorithms allow to obtain only approximate solutions to the MDGP.

In order to overcome some limitations of the existing methodologies for the MDGP, and considering that some problems of the NP class can be efficiently tackled by means of soft-computing and population-based algorithms, in this work we propose a *memetic algorithm* (MA) [11] that combines swarm intelligence [12] and evolutionary computation [13], together with a local search algorithm. In particular, we combine the swarm-based optimization of Particle Swarm Optimization (PSO) with the crossover capabilities of Genetic Algorithms (GAs).

In our methodology, a swarm intelligence is used to move atoms belonging to a candidate solution (i.e., a 3D molecular structure) within the search space. Each solution is characterized by a different position of atoms, whereby even solutions with the same fitness value can have atoms with a completely different position due to roto-translations of the whole structure. A crossover operator, typically employed in GAs, is used to exchange substructures between candidate solutions; in this context, a local optimization method is exploited to find the optimal roto-translation of the exchanged substructure within the offspring solution. Besides the inter-atomic distance matrix, our memetic algorithm makes use of additional constraints: *(i)* the size of the search space where particles move is bounded according to the number of amino-acids of the target protein; *(ii)* we consider molecular chirality, a property of asymmetry that is imposed to protein structures during the optimization process.

To the best of our knowledge this work represents the first attempt to solve the MDGP by exploiting evolutionary computation techniques only. We show that our method is able to find good solutions to the MDGP in case of sparse matrices with exact distance constraints, even by considering only inter-atomic distance values smaller than 6Å.

M.S. Nobile, A.G. Citrolo and G. Mauri are with the Department of Computer Science, Systems and Communication, University of Milano-Bicocca, Milano, Italy (email: {nobile, citrolo, mauri}@disco.unimib.it). P. Cazzaniga is with the Department of Human and Social Sciences, University of Bergamo, Bergamo, Italy (email: paolo.cazzaniga@unibg.it). D. Besozzi is with the Department of Computer Science, University of Milano, Milano, Italy (email: besozzi@di.unimi.it). P. Cazzaniga and D. Besozzi are also with the Institute for Systems Analysis and Computer Science "Antonio Ruberti", CNR, Rome, Italy. This work was partially supported by the SYSBIO Centre for Systems Biology.

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

The paper is organized as follows. In Section II we give the formal definition of the MDGP with exact distances. In Section III our method for the solution of MDGP with incomplete information is explained in detail. In Section IV results are shown for a set of ten protein structures of different size. In Section V we discuss the significance of our method and describe some directions for future research.

II. The Molecular Distance Geometry Problem

The MDGP can be formulated as follows. Let N be the number of atoms in a protein π, and let $d_{ij} \in \mathbb{R}^+$ be the given distance between atoms i and j, with $i, j = 1, \ldots, N$ and $i \neq j$, measured according to some computational or experimental methodology (e.g., NMR). These distances can be arranged into a real-valued $N \times N$ matrix $\mathbf{d}$, such that d_{ij} is the value in the i-th row and j-th column in $\mathbf{d}$.

If we denote by $\mathbf{a}_i$ the 3D coordinate vector of atom i in the Euclidean space, i.e., $\mathbf{a}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ for each $i = 1, \ldots, N$, then the value d_{ij} will formally correspond to the Euclidean norm between the two atoms, i.e., $d_{ij} = \|\mathbf{a}_i - \mathbf{a}_j\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$; note that, anyway, these coordinate vectors are unknown.

The MDGP consists in finding the set of coordinate vectors $\mathbf{a}_1, \ldots, \mathbf{a}_N$ of all atoms in π, such that the Euclidean norm $D_{ij} = \|\mathbf{a}_i - \mathbf{a}_j\|$ between any pair of atoms i and j – evaluated according to these coordinates – is equal to the measured distance d_{ij}. Formally, the MDGP is solved if $D_{ij} = d_{ij}$ for all $i, j = 1, \ldots, N$, $i \neq j$.

Several approaches to MDGP have been proposed in recent years. Dong and Wu introduced a linear time algorithm, called "geometric buildup", to solve the 3D-DGP when the exact value of distances between all pairs of atoms are given [5]; recently, this approach has been extended in order to obtain an approximate solution for the MDGP with noisy distance values and sparse matrices [14], [15]. The main limitation of the geometric buildup strategy is that in the case of sparse matrices and, in particular, when some atoms are characterized by less than four distance constraints, this method is unable to find any solution. However, to overcome this limitation, it is possible to consider additional distance constraints arising from structural features of proteins, or using optimization algorithms [16], to reconstruct the complete molecular structure from a partial substructure obtained with the geometric buildup algorithm.

A branch and prune algorithm was proposed in [6], [7]: by exploiting additional constraints about the protein structures, this method considers a discrete search space in which the amino-acids can be placed only in two different positions with respect to their precursor in the protein structure. This algorithm has an exponential time complexity, however it is able to efficiently find solutions for some instances that satisfy particular structural properties.

There exist two approaches based on graph embedding [17] in 3D Euclidean space, able to deal with both noisy data and sparse matrices. The first one, called ABBIE algorithm [8], exploits a divide and conquer strategy and structural rigidity [18]; this method first identifies subproblems (i.e., subsets of nodes) that can be solved with an exact algorithm, then it applies a global optimization algorithm to combine partial solutions. The second one, called EMBED algorithm [9], uses the measured distances to derive a set of lower and upper bounds for all other distances; this requires the identification of the shortest path between each couple of nodes in a particular bigraph in order to derive triangle inequality limits [19]. A local optimization strategy is then applied to refine the solution obtained from the complete bounds set.

Finally, the DGSOL algorithm [10] combines a methodology to select good starting points for the optimization process with the Gaussian smoothing and continuation strategy [20], a technique used to reshape the objective function. So doing, a gradient minimization can be applied to the obtained smooth function in order to optimize the protein structure. The main limitation of DGSOL is that it provides only approximate solutions in presence of noisy information and sparse matrices.

III. Methods

A candidate solution of the MDGP can be encoded as a vector $\mathbf{\Pi} = (\mathbf{a}_1, \ldots, \mathbf{a}_N)$ of 3D coordinates, representing the positions of all atoms of protein π in the Euclidean space. This representation can be exploited by a traditional evolutionary methodology as GAs, with genetic operators specifically designed to work on candidate solutions encoding real values [21]. Even though GAs might be a feasible methodology for MDGP, swarm intelligence techniques like PSO are generally more suitable than GAs, since they natively optimize real-valued problems [22]. Nevertheless, the crossover operator of GAs – which exchanges the genetic material of two promising individuals to create an improved offspring generation – is an elegant and powerful means to obtain a recombination of individuals and a better exploration of the search space. Thus, in this work we propose a hybrid methodology which combines the swarm-based optimization of PSO with the crossover capabilities of GAs. Crossover, in this case, implements the exchange of a subset of atoms between individuals, i.e., substructures of the candidate solutions. We define a substructure $\sigma = (i_1, \ldots, i_K)$, $K \leq N$, as the vector of indexes corresponding to a subset of atoms positions $(\mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_K})$ of solution $\mathbf{\Pi}$.

During the PSO optimization phase, the atoms belonging to each candidate solution move inside the search space and can be placed in completely different positions, so that even if two individuals are characterized by the same fitness value they might be rotated or translated with respect to each other in the 3D space. As a consequence, the crossover operator might move a substructure from an individual into another in such a position that the new (offspring) molecule will have a worse fitness value, because of an uncontrolled scattering of atoms. To reduce this potentially deleterious impact of crossover, we also perform a local optimization by means of a steepest descent algorithm aimed at optimizing the roto-translation that must be applied to the substructure. Global optimization methods coupled to local search are called memetic algorithms [11], thus we define our methodology as a Memetic Hybrid PSO plus GAs (MemHPG).

In what follows we provide a brief description of PSO and GAs, in order to clarify which mechanisms of both techniques are involved in our novel hybrid methodology. Finally, we provide a global definition of MemHPG.

A. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a heuristic inspired by the collective movement of birds and fish [22]. PSO exploits a set (the *swarm*) of N candidate solutions (the *particles*), which move inside a M-dimensional bounded search space in a collective effort to find the global optimal solution to a specified problem. At each iteration step t of the PSO, each particle is characterized by two vectors: the position $\mathbf{x}_i(t) \in \mathbb{R}^M$ and the velocity $\mathbf{v}_i(t) \in \mathbb{R}^M$. In the most common formulation, the movement of the i-th particle is a consequence of two attractors: the best position found by the swarm ($\mathbf{g}$) and the best position found by the particle itself ($\mathbf{b}_i$). Both attractions are perturbed by means of vectors of random numbers ($\mathbf{r}_1$ and $\mathbf{r}_2$) sampled with uniform distribution in [0,1], in order to avoid the entrapment in local minima; in addition, they are multiplied by two constants called *social* (c_{soc}) and *cognitive* (c_{cog}) factors. Hence, the velocity update formula for PSO is

$$\mathbf{v}_i(t+1) = w \cdot \mathbf{v}_i(t) + c_{soc} \cdot \mathbf{r}_1 \circ (\mathbf{g} - \mathbf{x}_i(t)) + \\ + c_{cog} \cdot \mathbf{r}_2 \circ (\mathbf{b}_i - \mathbf{x}_i(t)), \quad (1)$$

where $w \in \mathbb{R}^+$ is an inertia weight factor, used to damp the velocity. Moreover, the intensity of the velocity is generally clamped to a maximum value $v_{\text{MAX}} \in \mathbb{R}^+$, before the particles positions are updated according to

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1). \quad (2)$$

Note that, as a consequence of their position update, particles might move outside the search space; to avoid this problem, in this work we consider the "absorbing" boundary conditions strategy described in [23].

Thanks to the collective movement of particles, PSO eventually converges to an optimal solution. The algorithm is stopped when a halting criterion is met, e.g., after a fixed number of iterations.

B. Genetic Algorithms

GAs were introduced by Holland in 1975 [24] as a global search methodology inspired by the mechanisms of natural selection. GAs exploit a population P^0 composed of Q randomly created individuals that are usually defined as fixed-length strings (over a finite alphabet) that represent solutions of the problem under investigation. The individuals of the population undergo an iterative process whereby three genetic operators (selection, crossover, mutation) are applied, according to a given fitness function, to simulate the evolution process which results in a new population P^1 of possibly improved solutions.

During the selection process, individuals from P^0 are chosen and inserted into P^1 using some fitness-dependent sampling procedure [25]. Among the existing strategies, in this work we exploit the "tournament selection", in which a subset of $2 \leq q \leq Q$ individuals of population P^0 is randomly chosen and the individual having the best fitness is copied into the new population P^1. Once Q individuals have been selected, the crossover operator is used to combine promising parent individuals into new and improved offspring, which are collected into a third population P^2. Finally, in GAs, the mutation operator is used to perturb the encoding of individuals in P^2, allowing a further exploration of the search space.

After the application of genetic operators, individuals of P^0 are substituted by those in P^2 and the process iterates until a halting criterion is met, e.g., after a fixed number of generations.

C. A Memetic Hybrid Methodology for MDGP (MemHPG)

The hybrid algorithm we propose for the MDGP combines the emergent, self-organizing behavior of swarms with the strength of crossover-based recombination. In MemHPG, the self-organization is performed by a modified version of PSO which is used to arrange the atoms in the search space of a candidate structure; the crossover, on the other side, is applied to a population of independent candidate structures, to exchange their optimal substructures. As a matter of fact, our hybrid algorithm works on two layers: the inner layer of atoms (hereby called the *PSO-layer*) and the outer layer of molecule structures (the *GA-layer*) (Figure 1).

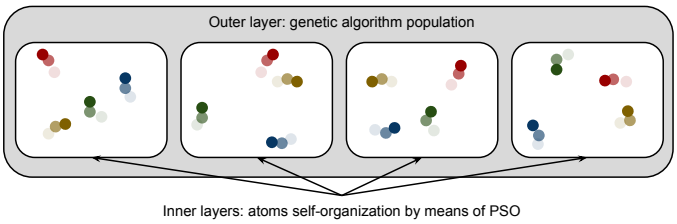

Fig. 1. Schematization of the two-layer hybrid methodology. In the outer layer, a population of candidate solutions exchange promising substructures exploiting GAs crossover, while each candidate solution evolves in the inner layer by means of PSO.

PSO-layer. In our modified version of PSO, we use a single particle for each atom, so that the position $\mathbf{x}_i$ of the i-th particle here corresponds to the 3D Euclidean coordinate vector $\mathbf{a}_i$ of the i-th atom of protein π (thus, $M = 3$). Therefore, in this particular formulation, particles do not represent a solution to the MDGP problem; instead, they represent a solution for the sub-problem of identifying the optimal spatial positioning of atoms. The size of the search space for particle positioning was defined according to the number A of amino-acids in protein π (which is known *a priori*), considering the notion of radius of gyration of proteins [26] (whose upper bound was identified as $A^{3/5}$). In MemHPG, the best setting for the search space was empirically found to be $4 \cdot A^{3/5}$Å for each dimension in the 3D Euclidean space.

The initial position of particles is randomly generated within the search space, except for the particle corresponding

to the first atom in π which is placed in position $(0, 0, 0)$ and kept fixed during the optimization. The rationale behind this choice is that, by keeping one particle fixed, the rest of the swarm is constrained to self-organize around it, thus reducing the chaoticity of the overall movements.

Once particles are distributed in the search space, we calculate an *error* ε to estimate the precision of the corresponding candidate solution, considering only the given distance constraints d_{ij}, without any additional knowledge about the original structure:

$$\varepsilon = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} g_{ij}}{\sum_{i=1}^{N} k_i}, \tag{3}$$

where

$$g_{ij} = \begin{cases} 0 & \text{if } d_{ij} \text{ is not given} \\ |D_{ij} - d_{ij}| & \text{otherwise,} \end{cases} \tag{4}$$

and the value k_i denotes the number of atoms $j \neq i$ in π for which a distance value d_{ij} is given. Since Equation 3 allows to discriminate the quality of solutions, we exploit it as the fitness function.

At each step of the PSO procedure in MemHPG, each particle considers a novel kind of attractor, named the *aggregate attractor* and denoted by $\mathbf{h}_i(t) \in \mathbb{R}^3$, which is calculated by comparing the distance between the coordinate vectors of all other atoms in the candidate solution (D_{ij}) against the distance measured with NMR experiments (d_{ij}):

$$\mathbf{h}_i(t) = \sum_{j \neq i} \frac{\delta_{ij}}{D_{ij}} (\mathbf{a}_j(t) - \mathbf{a}_i(t)), \tag{5}$$

where $\delta_{ij} = D_{ij} - d_{ij}$ is used to weight the attraction between the atoms, so that the contribution to the aggregate attractor of atoms whose distance D_{ij} is close to the measured distance d_{ij} will be reduced. It is worth noting that the aggregate attractor $\mathbf{h}_i$ can be seen as a linear combination of $N - 1$ "global" attractors of particle $\mathbf{a}_i$, each one with a different social factor equal to δ_{ij}.

When two atoms are farther than expected, they act as mutual attractors; on the contrary, when the atoms are closer than expected, they behave as repulsers. Figure 2 provides two examples of this mechanism, represented in the x-y projection plane for the sake of simplicity. According to Equation 5, the aggregate attractor for atom i is calculated as the sum of all attractive/repulsive contributions of all other atoms $j \neq i$; in Figure 3 we show an example which considers the aggregation of the contributes due to two atoms.

In order to consider only this new attractor in our modified version of PSO, Equation 1 is modified as follows:

$$\mathbf{v}_i(t+1) = w \cdot \mathbf{v}_i(t) + \mathbf{r} \circ \mathbf{h}_i(t), \tag{6}$$

where $\mathbf{r}$ is a vector of random numbers uniformly sampled in [0,1].

Once the putative velocity $\mathbf{v}_i(t+1)$ is calculated, its velocity is clamped, i.e., if $||\mathbf{v}_i(t+1)|| > v_{\text{MAX}}$ then

$$\mathbf{v}_i(t+1) = \frac{\mathbf{v}_i(t+1)}{||\mathbf{v}_i(t+1)||} \cdot v_{\text{MAX}}$$

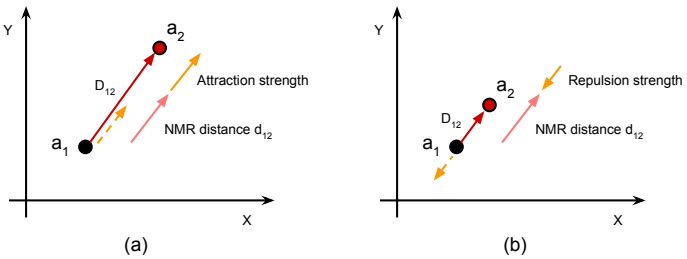

Fig. 2. Example of the attraction/repulsion mechanism of our modified PSO. For the sake of clarity, only the vectors for particle $\mathbf{a}_1$ are shown. (a) When the distance between two atoms (the red arrow between $\mathbf{a}_1$ and $\mathbf{a}_2$) is larger than the one measured by NMR (pink arrow), the atoms attract each other (dashed yellow arrow). (b) When the distance between the two atoms is smaller than the distance measured by NMR, the atoms act as repulsers.

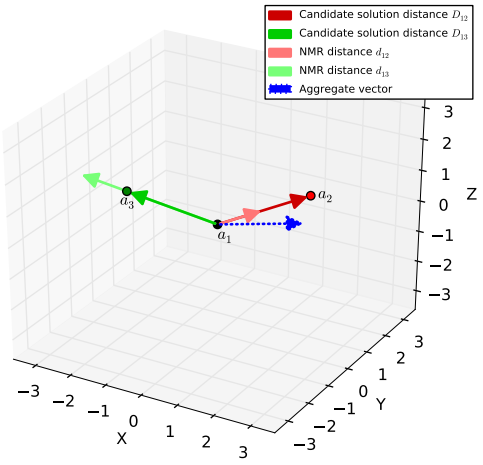

Fig. 3. Example of calculation of the aggregate attractor for particle $\mathbf{a}_1$, in a 3-atoms system. The length of the red arrows represents the distance between particles $\mathbf{a}_1$ and $\mathbf{a}_2$ according to the candidate solution (dark red) and to NMR data (light red): since the latter is shorter than the former, $\mathbf{a}_2$ acts as an attractor for $\mathbf{a}_1$. The length of the green arrows represents the distance between particles $\mathbf{a}_1$ and $\mathbf{a}_3$ according to the candidate solution (dark green) and to NMR data (light green): since the latter is longer than the former, $\mathbf{a}_3$ acts as a repulser for $\mathbf{a}_1$. The resulting aggregate attractor $\mathbf{h}_1$ is represented by the blue vector. The same process is applied to particles $\mathbf{a}_2$ and $\mathbf{a}_3$ (not shown here).

and the position $\mathbf{a}_i(t+1)$ is updated according to Equation 2. During the last generations of MemHPG, the finer positioning of atoms in the candidate structures requires smaller and more controlled movements with respect to the initial phases. For this reason, our methodology self-adapts the $v_{\text{MAX}}(t)$ value as follows:

$$v_{\text{MAX}}(t) = \begin{cases} \alpha \cdot v_{\text{MAX}}(t-1) & \text{if } \varepsilon^*(t) > \varepsilon^*(t-1) \\ v_{\text{MAX}}(t-1) & \text{otherwise,} \end{cases}$$

where $\varepsilon^*(t) \in \mathbb{R}^+$ represents the smallest error value among all particles at generation t and $\alpha \in (0, 1)$ is the velocity adaptation factor. The iterative update of velocity vectors, calculated according to the aggregate attractor, allows the set

of atoms to self-organize in a single optimal position. The inertia weight and the randomness due to $\mathbf{r}$ allow particles to avoid a chaotic behavior and local optima.

GA-layer. To help the convergence to an optimal solution, we introduce a second layer by instantiating Q multiple independent candidate solutions, which constitute the population of a GA. In this work, we do not exploit the mutation operator, which is conceptually realized by PSO: our GA performs tournament selection and crossover only. These operators are applied every I_χ iterations, and work together to generate the offspring population.

The functioning of the GA-layer is summarized in the following steps:

- a subset $P^0_{\texttt{TOUR}} \subset P^0$ of q individuals, $1 < q < Q$, is sampled using a uniform distribution;
- the best individual $P_{\texttt{BEST}} \in P^0_{\texttt{TOUR}}$ is deterministically identified (according to the fitness values) and copied into the new population P^1;
- for each atom i, such that $\mathbf{a}_i \in P_{\texttt{BEST}}$, a substructure σ_i is identified as explained below and inserted in a set Σ;
- one element $\sigma \in \Sigma$ – that is, a protein substructure – is chosen with a probability proportional to its length (i.e., the number of atoms in σ);
- one individual $P_{\texttt{RND}}$ is randomly chosen from $P^0_{\texttt{TOUR}} \setminus P_{\texttt{BEST}}$ with a uniform probability;
- the atoms in the substructure σ are positioned into individual $P_{\texttt{RND}}$ according to the best roto-translation (as explained below), thus replacing the corresponding atoms and generating a new individual $P^\sigma_{\texttt{RND}}$;
- finally, $P^\sigma_{\texttt{RND}}$ is inserted into population P^1 and the velocities of its particles are set to zero.

This procedure is repeated until $|P^1| = Q$; then, P^1 replaces P^0.

The substructures in Σ are chosen as follows. For each atom i in $P_{\texttt{BEST}}$, a substructure σ_i is determined according to the following greedy algorithm:

- atom i is inserted in σ_i;
- find atom j, $j \neq i$, such that $|\delta_{ij}| = \min\{|\delta_{ik}| \mid k = 1, \ldots, N, k \neq i\}$. If $|\delta_{lj}| < \varphi_{\texttt{min}}$, for each $l \in \sigma_i$ and $l \neq j$, then add atom j to σ_i; otherwise stop, as the substructure cannot be extended.

The value $\varphi_{\texttt{min}}$ is defined as $\varphi_{\texttt{min}} = \min\{\varphi_i \mid i = 1, \ldots, N\}$, where $\varphi_i = \frac{1}{N}\sum_{j \neq i} |\delta_{ij}|$. The procedure is iterated until no more atoms can be inserted in σ_i or its length reaches a given value $size_{\texttt{MAX}}$, that corresponds to a fixed percentage of the total number of atoms in π.

Since the chosen substructure σ can be oriented and translated in space in any possible way, we optimize its positioning in $P_{\texttt{RND}}$: the crossover embeds a local search optimization to identify the best roto-translation of σ with respect to its surrounding atoms in $P_{\texttt{RND}}$. More precisely, this is done by first calculating the centroid of σ and of the corresponding subset of atoms that will be replaced in $P_{\texttt{RND}}$, and then exploiting a gradient descent method to identify an optimal translation vector $\mathbf{t} = (t_x, t_y, t_z)$ and an optimal rotation vector $\theta = (\theta_x, \theta_y, \theta_z)$ with respect to these centroids which minimize the impact to the aggregate attractor.

After the crossover process, the PSO starts again. In addition, every 50 iterations of MemHPG the *chirality*[1] of candidate solutions is verified, since the information contained in the distance matrix is not sufficient to discriminate between a correct reconstructed molecule and molecules with a different chirality. For space limits, we do not provide a detailed description of this procedure; briefly, for each candidate solution we identify the substructures whose chirality is not correct, and we modify them by means of matrix operations implementing rotations and reflections in the Euclidean space, according to the tetrahedral geometry of the chemical bonds of the chiral carbons. This procedure is performed after each crossover process. When the chirality verification is completed, the PSO starts again.

MemHPG stops when a user-defined termination criterion is met, i.e., after a fixed number of iterations $I_{\texttt{MAX}}$.

IV. Results

In this section we present the results obtained by MemHPG for the reconstruction of the 3D structure of different proteins. At first, we performed several tests to determine the influence of the values of PSO and GA parameters on the reconstruction process, in order to find the best settings of MemHPG that were then exploited in all experiments.

These tests consisted in the variation of a single parameter at a time in the optimization process of an *in silico* generated 3-peptide molecule with a length of $N = 56$ atoms. For each parameter, each test was repeated 30 times, and the average smallest error achieved with the different MemHPG parameterizations was used to evaluate the influence of that parameter. All preliminary tests, were performed with $I_{\texttt{MAX}} = 2000$, unless otherwise specified.

In the first test we analyzed the impact of the population size Q, by considering the following values: 32, 64, 128, 256 individuals. As expected, the average smallest error achieved decreases as the population size increases (data not shown); however, for $Q > 32$, the improvement of the solutions quality is so slight that it does not justify the larger use of computational resources that it would require. Therefore, the value used in all consecutive tests was set to $Q = 32$.

In the second test we analyzed the impact of different values for the coefficient γ used to clamp the particles initial maximum velocity, i.e., $v_{\texttt{MAX}} = D_{\texttt{MAX}}/\gamma$, where $D_{\texttt{MAX}}$ is equal to the diagonal length of the search space. As shown in Figure 4, the best results were achieved when $\gamma = 10$, while smaller

[1]Chirality is a property related to a lack of symmetry that regards organic molecules such as amino-acids; we observe chirality when a carbon atom is bound to four different chemical groups (these carbon atoms are called chiral carbons) [27]. For each protein substructure composed of one chiral carbon and the atoms bound to it, two different but isometric conformations are possible; therefore, we must apply a geometric transformation to adjust the right positioning of atoms in order to impose the specific conformation of these groups, that is typical of protein molecules [28]. The subset of atoms that can lead to a wrong reconstruction can be identified *a priori* by analyzing carbon atoms and their bound chemical groups.

values (e.g., $\gamma = 1$) and higher values (e.g., $\gamma > 500$) lead to worse results.

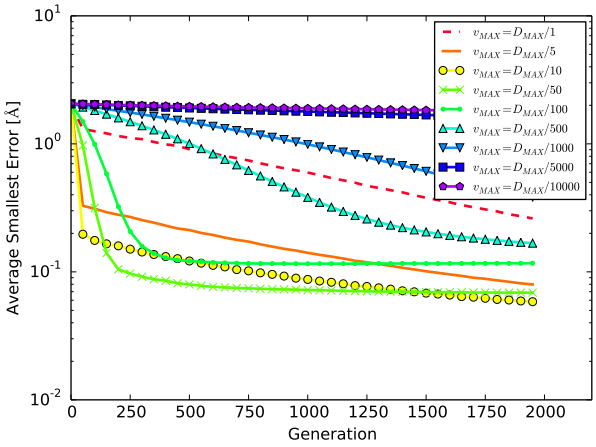

Fig. 4. Average smallest error computed over 30 runs of MemHPG varying the coefficient γ in $v_{\text{MAX}} = D_{\text{MAX}}/\gamma$. The best results were achieved with $\gamma = 10$; note that both high and small values for the maximum velocity of particles lead to higher values of the average smallest error.

The third test consisted in varying the adaptive velocity factor α. In Figure 5 we show the average smallest error obtained with 30 runs of MemHPG with several values of factor α. In this test, where $I_{\text{MAX}} = 4000$, the best results were obtained with $\alpha = 0.999$, even if smaller values of this factor allowed a faster convergence.

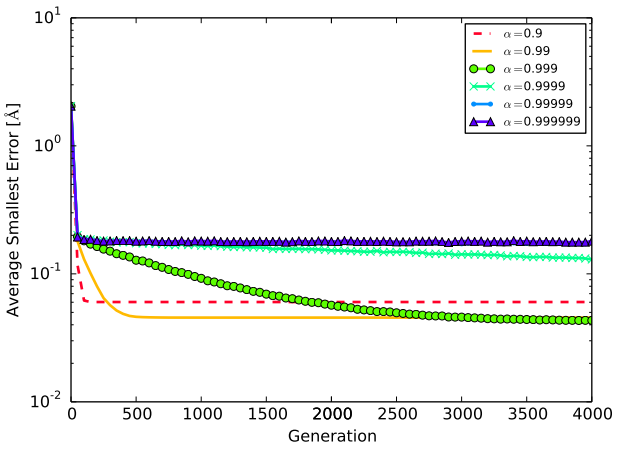

Fig. 5. Average smallest error computed over 30 runs of MemHPG varying the adaptive velocity factor α. Even though for α equal to 0.9 or 0.99 we obtained a faster convergence, the value $\alpha = 0.999$ allowed to achieve the best results.

A further test concerned the influence of the inertia weight on the particles velocity; in particular, we varied the w value in the range $[0, 1]$ and the best result was achieved with $w = 0.4$. Similarly to the case of v_{MAX}, both higher and smaller values of the inertia weight lead to higher values of the average smallest error (data not shown).

The last three tests aimed at finding the best setting for the tournament size, the crossover frequency and the maximum length allowed for a substructure involved in the crossover operation. The best tournament size value was identified around 10% of the population size Q, in order to have a high selection pressure able to maintain the population diversity throughout the generations. The crossover frequency was set to $I_\chi = 50$, meaning that every 50 generations the individuals undergo this operator. We observed that, despite the crossover improves the average quality of the candidate solutions, increasing its application frequency worsen the fitness of individuals (data not shown). Finally, the maximum length allowed for a substructure involved in the crossover operation was set to $size_{\text{MAX}} = 15\%$ of the total number of atoms in π (for higher values better results can be achieved, but the improvement of the fitness is not enough to justify the larger use of computational resources that it would require).

The results of these preliminary tests led to the following best parameter settings for MemHPG:

- population size $Q = 32$ individuals;
- initial $v_{\text{MAX}} = D_{\text{MAX}}/10$;
- adaptive velocity factor $\alpha = 0.999$;
- inertia weight $w = 0.4$;
- tournament size $q = 4$ individuals;
- crossover interval $I_\chi = 50$ generations;
- $size_{\text{MAX}} = 15\%$.

To test the validity of this setting of MemHPG we first reconstructed the 3-peptide molecule by using incomplete information. This was realized by removing from matrix $\mathbf{d}$ the distance values d_{ij} that are above a given cutoff. As shown in Figure 6, the average smallest error of the structures reconstructed by MemHPG is below 10^{-4}Å also in the case of a matrix $\mathbf{d}$ where $d_{ij} < 6$Å for all $i, j = 1, \ldots, N$.

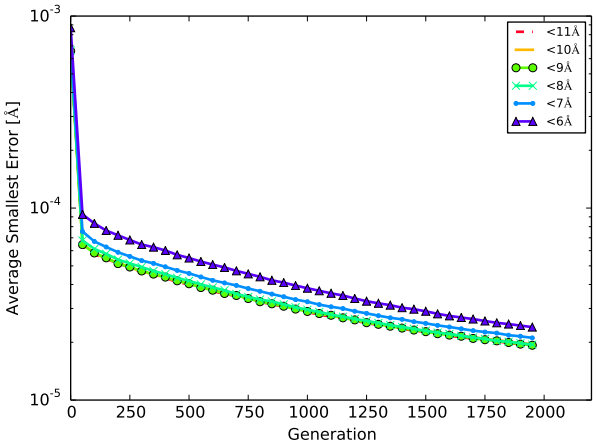

Fig. 6. Average smallest error of solutions to the 3-peptide molecule obtained in different optimization processes with incomplete information of inter-atomic distances. Note that, by exploiting only distances $d_{ij} < 6$Å, we still achieved an error lower than 10^{-4}Å with respect to the original structure.

To show the effectiveness of our methodology, in Table I we present the results obtained from the reconstruction of the

TABLE I

RESULTS OF THE RECONSTRUCTION OF PROTEIN STRUCTURES WITH MEMHPG USING ONLY DISTANCES $d_{ij} < 6$Å OR $d_{ij} < 7$Å

		$d_{ij} < 6$Å		$d_{ij} < 7$Å	
PDB ID	N	ε [Å]	RMSD [Å]	ε [Å]	RMSD [Å]
1PTQ	402	0.152	1.23	0.019	0.08
1CTF	487	0.180	1.46	0.037	0.18
1RGD	548	0.149	1.24	0.014	0.04
1HOE	558	0.172	1.63	0.130	1.7
1LFB	641	0.206	2.21	0.254	2.08
1F39	767	0.278	3.25	0.090	0.93
1PHT	814	0.291	2.02	0.123	1.86
1POA	914	0.056	0.99	0.074	1.26
1AX8	1003	0.092	2.27	0.075	1.59

structure of 9 proteins of increasing length – taken from the PDB database [16], [29] – using only inter-atomic distances $d_{ij} < 6$Å or $d_{ij} < 7$Å. In particular, for each protein, we indicate the error ε (defined in Section III) and the Root Mean Square Deviation (RMSD) [30] of the best structures found by MemHPG after $I_{\texttt{MAX}} = 20000$ iterations. These results highlight the robustness of our method since the ε value is low in all cases and, in addition, the RMSD is always lower than 3.5Å, a value that is considered to be indicative of a good reconstruction of protein structures [31].

In Figure 7, we show the structural alignment, realized with PyMOL [32], of the protein structures obtained with MemHPG (using inter-atomic distances below the 6Å cutoff) with the structures available in the PDB database. In the case of proteins 1AX8, 1HOE and 1CTF we obtained a perfect alignment between the protein structure; however, concerning protein 1F39, there is a slight discrepancy between the correct structure and the one obtained with MemHPG, probably due to an error in the reconstruction of a small portion of the protein (as better explained in the caption of Figure 7), while the overall structure is preserved also in the unaligned region.

V. CONCLUSIONS

In this paper we proposed an efficient method to solve the Molecular Distance Geometry Problem when only incomplete information about inter-atomic distances is available. Our methodology, called MemHPG, is a memetic algorithm that combines swarm intelligence and evolutionary computation along with a local search aimed at improving the effectiveness of the crossover operator. MemHPG works at two different levels: the PSO-layer is used to move particles in the 3D space, where each particle encodes the coordinates of an atom of the protein structure to be reconstructed; the GA-layer is exploited to select individuals, and to recombine them by means of a crossover operator that exchanges substructures between individuals. The crossover – aided by a local search method used to identify the best roto-translation of the exchanged substructure – is followed by a chirality correction, in which the correct orientation of amino-acids is verified and adjusted.

MemHPG was tested on a set of proteins having a number of atoms ranging from 402 to 1003; in all cases we obtained a correct 3D structure, as confirmed by the values of RMSD (see Table I). Indeed, our results indicate that the accuracy achieved by MemHPG is comparable, and in some cases even better, to the accuracy achieved by state-of-the-art methods [15], [33]. This is a remarkable result since MemHPG relies on (incomplete) distance matrices and general features of protein structures, while the other methods require additional a priori assumptions about proteins to achieve good results. Moreover, two additional advantages of our method reside in its intrinsic stochasticity and extensibility: on the one hand, the various reconstructed structures (with low error values) that can be obtained in each run of MemHPG are useful to represent the structural variability observed in biological molecules, which is a source of noise in NMR data; on the other hand, MemHPG can be easily improved by including a molecular force field in the scoring function during the final stages of the optimization process, in order to select structural models that are more realistic from a physical point of view.

MemHPG evaluates the fitness of each candidate solutions by calculating the mutual distances between its atoms positions; therefore, the computational complexity of this method is $O(N^2)$. Moreover, since we exploit a population of candidate proteins, the complexity linearly scales to $O(Q \cdot N^2)$. Nevertheless, all calculations of our method can be straightforwardly accelerated using a parallel architecture. We are considering a new implementation of MemHPG according to the general-purpose GPU computing paradigm, so that by launching $Q \cdot N$ threads we can assign a specific thread to each atom of each candidate solution, strongly reducing the complexity down to $O(N)$ on GPU-equipped machines. Even though a parallel architecture may accelerate the execution of MemHPG, an efficient non-sequential implementation of our crossover mechanism is far from trivial and currently under investigation. Since we rely on incomplete distance information, multiple runs of our methodology may yield a set of different optimal conformations: GPU acceleration will allow us to collect statistical information about the potential structures of the analyzed protein. Finally, we described in Section IV how the average error decreases as we increase the population size Q: once again, the GPU acceleration would allow us to improve the quality of the results without a relevant impact on the running time of MemHPG.

REFERENCES

[1] K. Wüthrich, "NMR studies of structure and function of biological macromolecules (Nobel lecture)." *Angew. Chem. Int. Ed. Engl.*, vol. 42, no. 29, pp. 3340–63, 2003.

[2] A. Grosso, M. Locatelli, and F. Schoen, "Solving molecular distance geometry problems by global optimization algorithms," *Comput. Optim. Appl.*, vol. 43, no. 1, pp. 23–37, 2009.

[3] J. B. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," in *Proc. of 17th Allerton Conference in Communications, Control and Computing*, 1979, pp. 480–489.

[4] J. J. Moré and Z. Wu, "Global continuation for distance geometry problems," *SIAM J. Optimiz.*, vol. 7, no. 3, pp. 814–836, 1997.

[5] Q. Dong and Z. Wu, "A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances," *J. Global Optim.*, vol. 22, no. 1-4, pp. 365–375, 2002.

[6] C. Lavor, L. Liberti, A. Mucherino, and N. Maculan, "On a discretizable subclass of instances of the molecular distance geometry problem," in *Proc. of the 2009 ACM Symposium on Applied Computing*. ACM, 2009, pp. 804–805.

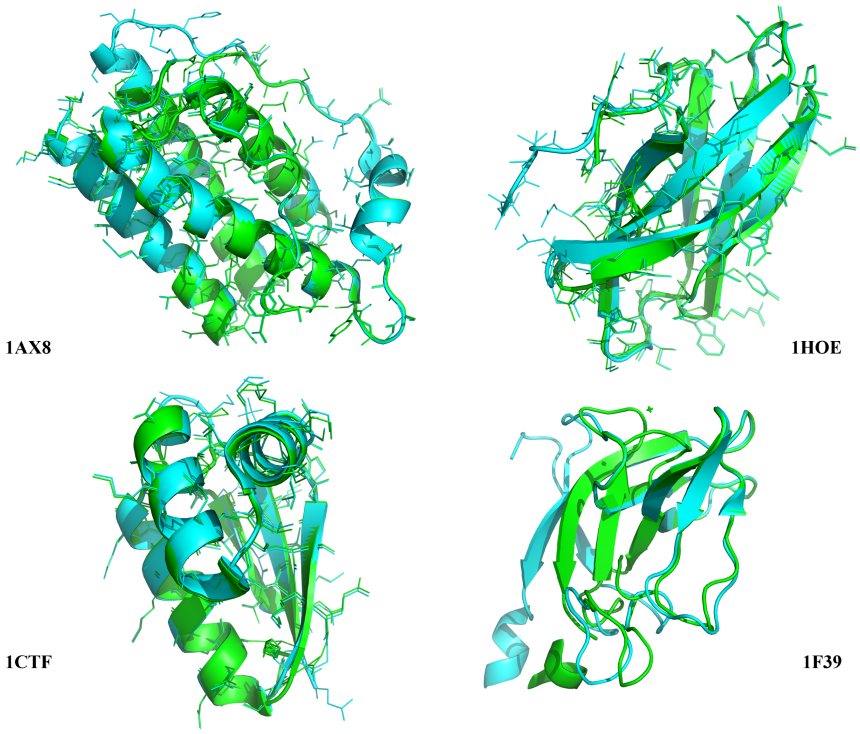

Fig. 7. Examples of the structural alignment between the structures available in the PDB database (*cyan*) and protein structures reconstructed by MemHPG, using distance matrices $\mathbf{d}$ with $d_{ij} < 6$Å (*green*). The alignments are correct, even though, in the case of protein 1F39, there is a slight discrepancy between the correct structure and the one obtained with MemHPG, probably due to an error in the reconstruction of a small portion of the protein connecting two major structural motifs, while the overall structure is preserved also in the unaligned region. This kind of errors can arise in portions of the proteins with extended structure, when a very low number of inter-atomic distances are available. Images and alignment obtained with PyMOL [32].

[7] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino, "The discretizable molecular distance geometry problem," *Comput. Optim. Appl.*, vol. 52, no. 1, pp. 115–146, 2011.

[8] B. Hendrickson, "The molecule problem exploiting structure in global optimization," *SIAM J. Optimiz.*, vol. 5, no. 4, pp. 835–857, 1995.

[9] G. M. Crippen and T. F. Havel, "Distance geometry and molecular conformation," *J. Comput. Chem.*, vol. 11, no. 2, pp. 265–266, 1990.

[10] J. Moré and Z. Wu, "Distance geometry optimization for protein structures," *J. Global Optim.*, vol. 15, no. 3, pp. 219–234, 1999.

[11] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE T. Evolut. Comput.*, vol. 9, no. 5, pp. 474–488, 2005.

[12] M. Dorigo, *Ant Colony Optimization and Swarm Intelligence: 5th International Workshop, ANTS 2006, Brussels, Belgium, September 4-7, 2006, Proceedings*. Springer-Verlag New York Incorporated, 2006.

[13] K. A. De Jong and W. M. Spears, "Using genetic algorithms to solve NP-complete problems." in *Proc. of the 3rd International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 124–132.

[14] A. Sit, Z. Wu, and Y. Yuan, "A geometric buildup algorithm for the solution of the distance geometry problem using least-squares approximation." *B. Math. Biol.*, vol. 71, no. 8, pp. 1914–33, 2009.

[15] A. Sit and Z. Wu, "Solving a generalized distance geometry problem for protein structure determination," *B. Math. Biol.*, vol. 73, no. 12, pp. 2809–2836, 2011.

[16] L. Fabry-Asztalos, I. Lorentz, and R. Andonie, "Molecular distance geometry optimization using geometric build-up and evolutionary techniques on GPU," in *2012 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2012, pp. 321–328.

[17] J. Reiterman, V. Rödl, and E. Šinajová, "Geometrical embeddings of graphs," *Discrete Math.*, vol. 74, pp. 291–319, 1989.

[18] L. Roth and B. Asimow, "The Rigidity of Graphs," *Trans. Amer. Math. Soc.*, vol. 245, pp. 279–289, 1978.

[19] T. F. Havel, "Distance geometry: Theory, algorithms and chemical applications," in *Encyclopedia of Computational Chemistry*. John Wiley & Sons, 1998, pp. 723–742.

[20] L. Piela, J. Kostrowicki, and H. A. Scheraga, "On the multiple-minima problem in the conformational analysis of molecules: deformation of the potential energy hypersurface by the diffusion equation method," *J. Phys. Chem.*, vol. 93, no. 8, pp. 3339–3346, 1989.

[21] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis," *Artif. Intell. Rev.*, vol. 12, no. 4, pp. 265–319, 1998.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of the IEEE International Conference on Neural Networks*, vol. IV, Piscataway, NJ, 1995, pp. 1942–1948.

[23] S. Xu and Y. Rahmat-Samii, "Boundary conditions in particle swarm optimization revisited," *IEEE Antennas Propag.*, vol. 55, pp. 760–765, 2007.

[24] J. H. Holland, *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[25] T. Bäck, "Selective pressure in evolutionary algorithms: A characterization of selection mechanisms," in *Proc. of the First IEEE Conference on Evolutionary Computation*, vol. 1. IEEE, 1994, pp. 57–62.

[26] L. Hong and J. Lei, "Scaling law for the radius of gyration of proteins and its dependence on hydrophobicity," *J. Polym. Sci. Pol. Phys.*, vol. 47, no. 2, pp. 207–214, 2009.

[27] A. Harris, R. Kamien, and T. Lubensky, "Molecular chirality and chiral parameters," *Rev. Mod. Phys.*, vol. 71, no. 5, pp. 1745–1757, 1999.

[28] D. Voet and J. G. Voet, *Biochemistry*. W.H. Freeman, 2011.

[29] "RCSB Protein Data Bank," www.rcsb.org.

[30] F. J. Burkowski, *Structural Bioinformatics: An Algorithmic Approach*. Chapman & Hall, 2009.

[31] T. Schlick, *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer, 2010.

[32] "The PyMOL Web Page," http://www.pymol.org.

[33] M. Souza, C. Lavor, A. Muritiba, and N. Maculan, "Solving the molecular distance geometry problem with inaccurate distance data." *BMC Bioinformatics*, vol. 14 Suppl 9, no. Suppl 9, p. S7, 2013.