Chapter 12
Solving Molecular Distance Geometry Problems Using a Continuous Optimization Approach

Rodrigo S. Lima and J.M. Martínez

Abstract The molecular distance geometry problem consists in finding the positions in $\mathbb{R}^3$ of atoms of a molecule, given some inter-atomic distances. In this work we formulate this problem as a nonlinear optimization problem and solve some instances using a continuous optimization routine. For each proposed experiment, we compare the numerical solution obtained with the true structure. This comparison is performed by solving a Procrustes problem.

Keywords Molecular distances • Nonlinear programming • Numerical experiments

12.1 Introduction

In this work we propose and solve some computational experiments involving instances of the molecular distance geometry problem [11]. We employ a continuous optimization software to find numerical solutions to the problem. Our objective is to reconstruct three-dimensional structures of proteins using only the distances between their atoms. To attain this goal, we need to determine a set of n points $\{x^1, x^2, \ldots, x^n\} \subset \mathbb{R}^3$ such that $\|x^i - x^j\| = \hat{d}_{ij}$, where $\hat{d}_{ij}$ is the Euclidean distance between the atoms i and j. We can formulate this task as a continuous optimization problem as follows:

R.S. Lima
Department of Mathematics and Computation, ICE-UNIFEI, Federal University of Itajubá, 37500-903 Itajubá MG, Brazil
e-mail: rodlima@unifei.edu.br

J.M. Martínez
Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, 13081-970 Campinas SP, Brazil
e-mail: martinez@ime.unicamp.br

A. Mucherino et al. (eds.), *Distance Geometry: Theory, Methods, and Applications*,
DOI 10.1007/978-1-4614-5128-0_12,

$$\text{minimize} \sum_{i,j} (\|x^i - x^j\| - \hat{d}_{ij})^2,$$

$$\text{subject to } x^i \in \mathbb{R}^3, i = 1, 2, \ldots, n. \tag{12.1}$$

The variables in Eq. (12.1) are the coordinates of points $x^i \in \mathbb{R}^3$, and the objective function is not differentiable when $x^i = x^j$, for some i, j. However, as the distances between atoms are always positive real numbers, we can apply a minimization algorithm that uses first derivatives to solve the problem (12.1). Then, if $\hat{d}_{ij} > 0$ for all i, j, the local minimizers of Eq. (12.1) are configurations that do not contain coincident points. This result was proved by Jan de Leeuw in [4]. In the computational experiments, we solve some instances of the molecular distance geometry problem using *GENCAN* [2]. This routine, available at

www.ime.usp.br/~egbirgin/tango

is able to find approximate solutions to minimization problems with box constraints. For each considered instance, the numerical solutions obtained by *GENCAN* were compared to the true structure of the analyzed protein. The comparison was carried out as follows: given the true configuration of the protein and a numerical solution, we determine a transformation that superimposes both structures in some optimal manner. This problem is known as the Procrustes problem [6, 8].

The Procrustes problem consists in finding an orthogonal matrix $Q \in \mathbb{R}^{3\times 3}$ that minimizes the function

$$g(Q) = \|M_0 - M_1 Q\|_F, \tag{12.2}$$

where M_0 and M_1 are matrices in $\mathbb{R}^{n\times 3}$ and $\|.\|_F$ is the Frobenius norm. The orthogonal matrix Q that minimizes Eq. (12.2) has a closed-form expression. In the book of Golub and Van Loan [5], a singular value decomposition is employed to determine Q. This way of solving Eq. (12.2) does not ensure that the orthogonal matrix Q is a rotation matrix. There are cases where Q is the composition of a rotation and of a reflection. Kearsley in [9] uses unitary quaternions to find Q, and, as a result, he always obtains a rotation matrix. More references about quaternions and rotations can be found in [3, 7, 10, 12]. We desire to investigate if the numerical solutions obtained by *GENCAN* to the problem (12.1) differ from the original configurations by transformations involving a pure rotation or a rotation followed by a reflection. For this, we solve the Procrustes problem applying both proposed techniques: singular value decomposition and unitary quaternions.

12.2 Numerical Experiments

We selected some proteins from protein data bank [1] and we considered only the alpha carbon coordinates (C_α) of each structure. The selected proteins and the number of atoms (n_{C_α}) are indicated in Table 12.1. We initially propose three sets

Table 12.1 Proteins used in the computational experiments

Protein	n_{C_α}
1AMU	509
1OOH	126
2O12	407
3RAT	124
6PAX	133
1lmO	3,535

Table 12.2 First set of experiments: all the distances are known

Protein	ndist	nvar	iter	evalf	$f(x^*)$	$t(s)$	pure rot.	rot. + ref.
1AMU	129,286	1,527	20	39	1.21E−19	1.76	13	7
1OOH	7,875	378	14	30	3.61E−19	0.08	12	8
2O12	82,621	1,221	17	36	1.10E−19	0.99	12	8
3RAT	7,626	372	17	27	3.84E−19	0.11	13	7
6PAX	8,778	399	18	42	7.27E−19	0.21	6	8
1lmO	6,246,345	10,605	26	68	5.59E−21	120.23	101	99

of tests with these proteins; the details are discussed below. All experiments in this work have been carried out on a single core of an Intel Core 2 CPU 2.4GHz with 2GB RAM, running MAC OS X 10.5.

12.2.1 Solving Problems Using All Distances Between Atoms

In this set of experiments we suppose that all the distances $\hat{d}_{ij}$ between the atoms are known. With the first five proteins of Table 12.1, we solve the problem (12.1) using *GENCAN* twenty times whereas the problem with 1lmO protein was solved two hundred times, where each run corresponds to a different starting point. Table 12.2 shows the results of runs for which *GENCAN* reached the lowest objective function value. The columns in this table have the following meaning: *ndist* is the total number of distances between pairs of atoms, *nvar* is the number of variables, *iter* and *evalf* are, respectively, the total number of iterations and evaluations of the objective function, $f(x^*)$ is the final objective function value, and $t(s)$ is the CPU time in seconds. The column *pure rot.* shows the quantity of runs in which the optimization routine obtained a solution that differs from the true structure by a transformation involving a pure rotation. The column *rot. + ref.* indicates the total of rounds in which the numerical solution differs from the true structure by a transformation involving a rotation followed by a reflection. The stopping criterion of *GENCAN* in all tests required that the gradient norm had to be smaller than 10^{-4}.

The final values of the objective function show that *GENCAN* finds configurations of points in $\mathbb{R}^3$ that fit all the distances. However, we noted in six tests related to the protein 6PAX that the routine obtains configurations with $f(x^*) \approx 10^3$ and gradient norm smaller than 10^{-4}. These configurations are certainly local minimizers.

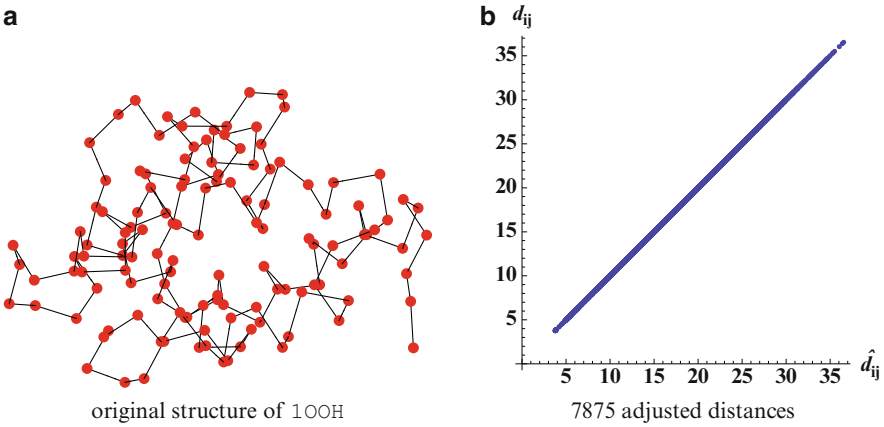

Fig. 12.1 Experiments with 1OOH protein

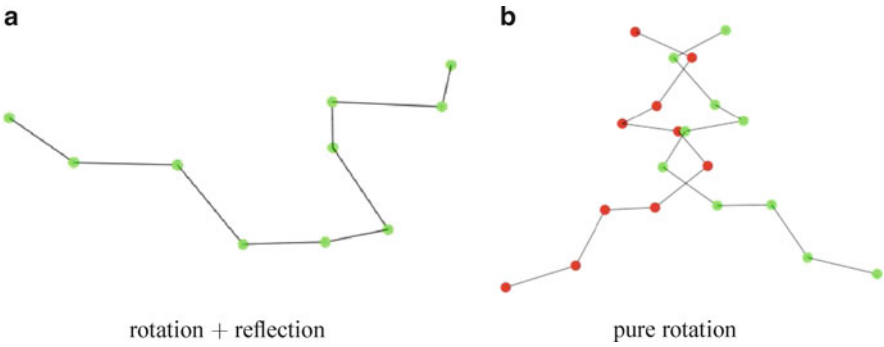

Fig. 12.2 1OOH protein: solving procrustes problem to compare structures

We chose the protein 1OOH to illustrate a test where *GENCAN* obtains a solution that differs of the true configuration by a transformation involving reflection. Figure 12.1a shows the true structure of 1OOH with 126 alpha carbons. Each atom is represented by a point in $\mathbb{R}^3$ and consecutive points are joined by lines. Figure 12.1b compares the original distances between pairs of atoms ($\hat{d}_{ij}$ axis) to the distances obtained numerically (axis d_{ij}).

The analysis with the Procrustes problem is shown in Fig. 12.2. To construct this figure we use only ten first consecutive C_α atoms of 1OOH. We solve the Procrustes problem (12.2) using the two formulations discussed above. Figure 12.2a shows the optimal superimposition of the true and numerical structures. The numerical solution (red points) does not appear in this image because it is superimposed by the true structure (green points). The transformation matrix obtained in this case involves a reflection and was obtained solving Eq. (12.2) with the strategy proposed

Table 12.3 Results of procrustes problem with 1OOH protein

Procrustes: Golub and Van Loan's strategy

$$Q = \begin{pmatrix} 0.545563 & 0.463014 & -0.698555 \\ 0.835074 & -0.229917 & 0.49979 \\ -0.0708004 & 0.856012 & 0.512085 \end{pmatrix}, \qquad g(Q) = 2.87132\text{E}{-10},$$

Procrustes: Kearsley's strategy

$$Q = \begin{pmatrix} 0.583586 & 0.810877 & 0.0436581 \\ 0.798626 & -0.563372 & -0.211681 \\ -0.147052 & 0.158401 & -0.976363 \end{pmatrix}, \qquad g(Q) = 1.37579\text{E}{+02}.$$

by G. Golub and C. Van Loan. Figure 12.2b shows the superimposition obtained by applying the strategy proposed by Kearsley. In this case, the orthogonal matrix describes a pure rotation. We note in Fig. 12.2b that the numerical configuration (red points) is the reflected image of the original one (green points). In this case, it is not possible to determine a rotation that superimposes both structures. The results obtained for the Procrustes problem are reported in Table 12.3.

12.2.2 Simulating Errors

In this set of experiments, we consider the first five proteins of Table 12.1 and we suppose that all distances between pairs of atoms were obtained with errors. To simulate this situation, we add to each value $\hat{d}_{ij}$ a random number created in the interval $[-\rho, \rho]$, with $|\rho| \leq 1$. Then, for each protein and each fixed value ρ, we solve Eq. (12.1) twenty times with a different starting point in each run. All tables below show only the results corresponding to the tests in which *GENCAN* reached the lowest final value of objective function. Table 12.4 indicates the final values of objective function attained by *GENCAN* and Table 12.5 shows the performance of the routine for solving each instance in terms of iterations, evaluations of the objective function, and CPU time. According to Table 12.4, when we increase the parameter ρ, the final values of the objective function also increase by a factor of 10^2. The performance of *GENCAN* in these problems was quite similar to the results obtained in correspondence with problems considered in the first set of experiments.

We built some figures for an experiment related to the protein 3RAT, where we fixed $\rho = 1$ (fifth line and last column of Table 12.4). Figure 12.3a shows the true structure of 3RAT with 124 alpha carbons, and Fig. 12.3b indicates the result of superimposing both structures (true and numerical) by a transformation involving a pure rotation. For building this figure, we use only the first ten atoms of structures: true (green points) and numerical (red points). In this case, we obtained the same

Table 12.4 Final values of objective function reached by *GENCAN*

Protein	$\rho = 10^{-5}$	$\rho = 10^{-4}$	$\rho = 10^{-3}$	$\rho = 10^{-2}$	$\rho = 10^{-1}$	$\rho = 1$
1AMU	4.263864E−06	4.263864E−04	4.263864E−02	4.263864E+00	4.263864E+02	4.263871E+04
1OOH	2.527256E−07	2.527256E−05	2.527256E−03	2.527253E−01	2.527222E+01	2.526905E+03
2O12	2.709833E−06	2.709833E−04	2.709833E−02	2.709833E+00	2.709808E+02	2.709746E+04
3RAT	2.453024E−07	2.453024E−05	2.453024E−03	2.453025E−01	2.453041E+01	2.453186E+03
6PAX	2.829028E−07	2.829028E−05	2.829028E−03	2.829023E−01	2.828973E+01	2.828446E+04

Table 12.5 Performance of *GENCAN* in tests with errors

	$\rho = 10^{-5}$			$\rho = 10^{-4}$			$\rho = 10^{-3}$		
Protein	iter	evalf	$t(s)$	iter	evalf	$t(s)$	iter	evalf	$t(s)$
1AMU	18	33	1.75	17	35	1.60	17	34	1.50
1OOH	16	40	0.10	20	38	0.11	17	30	0.10
2O12	21	52	1.31	21	45	1.53	18	34	1.25
3RAT	16	29	0.11	17	34	0.13	17	35	0.13
6PAX	16	34	0.17	22	48	0.26	19	50	0.15
	$\rho = 10^{-2}$			$\rho = 10^{-1}$			$\rho = 1$		
Protein	iter	evalf	$t(s)$	iter	evalf	$t(s)$	iter	evalf	$t(s)$
1AMU	20	34	1.79	24	63	1.76	18	40	1.84
1OOH	15	33	0.09	13	27	0.08	16	36	0.09
2O12	22	51	1.50	18	27	1.27	22	45	1.76
3RAT	15	19	0.11	16	30	0.12	15	28	0.10
6PAX	17	41	0.17	18	41	0.19	21	56	0.22

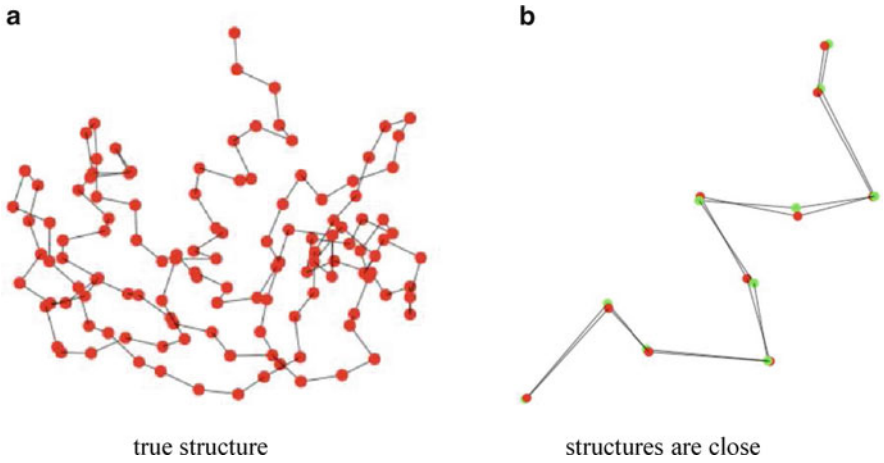

Fig. 12.3 Test with 3RAT protein

orthogonal matrix by solving the Procrustes problem with the two approaches described above:

$$Q = \begin{pmatrix} -0.728719 & -0.651336 & 0.211497 \\ 0.141985 & -0.44583 & -0.883785 \\ 0.669932 & -0.614001 & 0.417364 \end{pmatrix}, \quad g(Q) = 2.02778.$$

Figure 12.4 shows a graph where the x-and y-axes represent, respectively, the perturbed distances and the distances obtained by solving Eq. (12.1) with *GENCAN*. Although the final value of objective function is not small, the points are close to the line $y = x$.

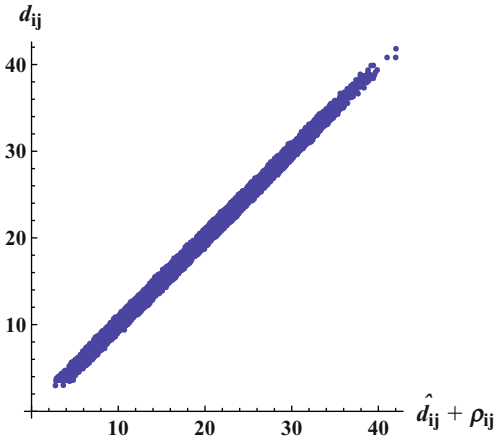

Fig. 12.4 Simulating 7,626 distances with errors

12.2.3 Solving Problems Using a Subset of Interatomic Distances

In these experiments, we use the same proteins reported in Table 12.1. However, we try here to recover the true structure using only distances not greater than a fixed parameter d_{fix}. Assuming that the distances are known exactly, we varied the parameter value d_{fix} and we analyzed the obtained results using the Procrustes technique. To each protein and each value d_{fix}, we solve the problem (12.1) fifty times with a multistart strategy: a different initial point was used in each run. In all the cases, *GENCAN* stopped when the gradient norm was lower than 10^{-4}.

Tables 12.6 and 12.7 show only information corresponding to the tests with the lowest value of the objective function attained by *GENCAN*. The total number of distances between pairs of atoms (*ndist*), the number of distances used to solve the problem (12.1) (*nd*), and the final value of the objective function ($f(x^*)$) are reported in Table 12.6. The performance of routine is indicated in Table 12.7. These results show that *GENCAN* can find configurations of points that fit all distances between atoms using less than 36 % of the known distances.

We illustrate a test with the protein `6PAX` where we attempt to recover the true structure considering only distances not greater than 10 Å. Figure 12.5a shows the true structure with 133 alpha carbons and Fig. 12.5b compares the original distances between atoms ($\hat{d}_{ij}$) with the distances in the numerical solution (d_{ij}). We can see in the graph that the points are concentrated around the line $y = x$. To create Fig. 12.6a, we solved the Procrustes problem using a singular value decomposition and we obtained a transformation involving a reflection. In the case of Fig. 12.6b, we applied the quaternion approach and, as a result, we obtained a pure rotation matrix. These results are shown in Table 12.8.

Table 12.6 Final values of objective function attained by *GENCAN*

		$d_{\text{fix}} = 6$		$d_{\text{fix}} = 10$		$d_{\text{fix}} = 15$	
Protein	ndist	nd	$f(x^*)$	nd	$f(x^*)$	nd	$f(x^*)$
1AMU	129,286	1,614	8.438471E+00	4,650	2.514611E+02	14,193	5.554569E−17
1OOH	7,875	386	1.929190E−01	1,074	3.657396E−17	2,824	1.198010E−18
2O12	82,621	1,221	4.402781E+00	3,899	5.413680E+00	11,083	2.446779E−13
3RAT	7,626	393	1.189479E+00	1,045	2.463908E−10	2,724	2.767508E−18
6PAX	8,778	386	4.858342E−03	916	2.438740E+02	2,292	1.311720E−13

		$d_{\text{fix}} = 10$		$d_{\text{fix}} = 9$		$d_{\text{fix}} = 8$	
Protein	ndist	nd	$f(x^*)$	nd	$f(x^*)$	nd	$f(x^*)$
1lmO	6246345	1085393	1.393982E−17	839689	6.667112E−15	677434	4.904778E−13

Table 12.7 Performance of *GENCAN* in the resolution of problems

	$d_{\text{fix}} = 6$			$d_{\text{fix}} = 10$			$d_{\text{fix}} = 15$		
Protein	iter	evalf	$t(s)$	iter	evalf	$t(s)$	iter	evalf	$t(s)$
1AMU	121	428	48.88	52	173	2.53	64	192	2.88
1OOH	150	304	8.490	34	93	0.15	18	35	0.08
2O12	328	733	137.91	57	176	3.06	36	127	1.60
3RAT	111	244	5.19	44	115	0.45	28	78	0.14
6PAX	66	160	3.54	142	326	6.34	49	84	1.80
	$d_{\text{fix}} = 10$			$d_{\text{fix}} = 9$			$d_{\text{fix}} = 8$		
Protein	iter	evalf	$t(s)$	iter	evalf	$t(s)$	iter	evalf	$t(s)$
1lmO	63	245	103.51	65	235	111.52	35	98	66.41

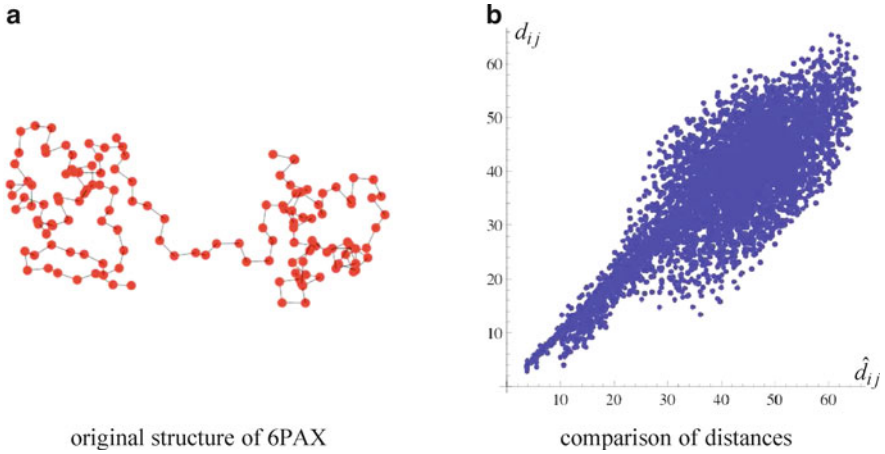

Fig. 12.5 Experiments with 6PAX protein

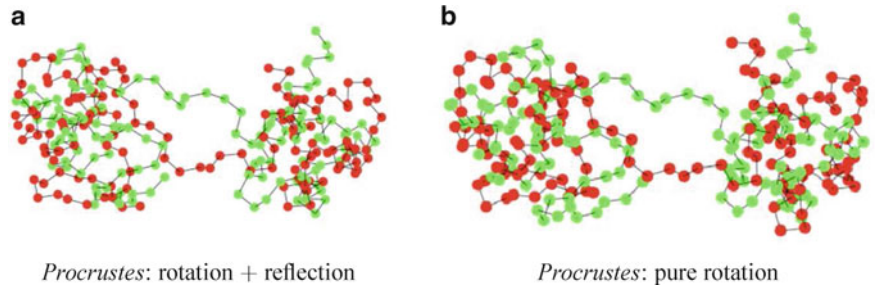

Fig. 12.6 Comparison of structures via *procrustes*

According to the experiments, we can conclude that it is possible to use a continuous optimization routine to recover a 3D structure of a protein using only a subset of known distances between pairs of atoms. In particular, if we provide to *GENCAN* a reasonable starting point, the routine solves the problem very quickly.

Table 12.8 Results of procrustes problem with 6PAX protein

Procrustes: Golub and Van Loan's strategy

$$Q = \begin{pmatrix} 0.619471 & -0.528366 & -0.580591 \\ -0.756835 & -0.59837 & -0.262972 \\ 0.208462 & -0.602315 & 0.770558 \end{pmatrix}, \qquad g(Q) = 1.17023\text{E}{+}02,$$

Procrustes: Kearsley's strategy

$$Q = \begin{pmatrix} 0.584185 & -0.546667 & -0.599903 \\ 0.787414 & 0.20257 & 0.582189 \\ -0.196741 & -0.812478 & 0.548792 \end{pmatrix}, \qquad g(Q) = 1.30287\text{E}{+}02.$$

Table 12.9 Comparing *GENCAN* and *MDJEEP*

					GENCAN		*MDJEEP*	
Protein	nat	ndist	nd	d_{fix}	$t(s)$	E_{sol}	$t(s)$	E_{sol}
1CRN	138	9,453	1,250	6.0	1.41	9.63E−08	0.001	9.63E−05
1PTQ	150	11,175	1,263	6.0	1.65	9.53E−04	0.001	9.78E−05
2ERL	120	7,140	1,136	6.0	0.44	4.17E−08	0.001	8.65E−05
1PPT	108	5,778	1,039	6.5	0.74	6.15E−04	0.001	9.51E−05
1PHT	249	30,876	2,631	6.5	5.60	3.66E−08	0.002	9.34E−05
1HOE	222	24,531	2,715	7.0	0.79	1.85E−10	0.002	8.12E−05
3RAT	372	69,006	4,567	7.0	3.37	1.53E−09	0.004	8.82E−05
1A70	291	42,195	4,472	8.0	33.48	6.37E−10	0.003	7.59E−05

12.2.4 *Comparing* GENCAN *and* MD-jeep

To finish this work, we compare the performances of *GENCAN* to the ones of a software tool named *MD-jeep*. *MD-jeep* was developed specifically to solve molecular distance geometry problems using combinatorial optimization techniques [13]. This software was written in C by Mucherino et al., and it is freely distributed at

www.antoniomucherino.it/en/mdjeep.php

In order to run the experiments, we used eight instances obtained from protein conformations downloaded from Protein Data Bank. We extracted the coordinates of atoms N, C_α, and C from each structure. For each protein, only distances not greater than $d_{\text{fix}} = 6$ Åwere considered as input for the routines. To solve the problems with *GENCAN*, we employ a multistart strategy: we perform runs until the routine provides a solution that differs from the original structure by a transformation involving a pure rotation. *GENCAN* stopped in all tests with the gradient norm smaller than 10^{-4}. The solutions of *MD-jeep* listed in Table 12.9 differ from original structure by a linear transformation involving a rotation matrix. The columns of Table 12.9 have the following meaning: *nat* is the number of atoms N, C_α, and C present in each protein, *ndist* is the total number of distances between pairs of atoms, *nd* is the number of distances not greater than $d_{\text{fix}} = 6$ Å, and $t(s)$ is the CPU time, in seconds.

After solving the problems with both packages, we analyze the quality of the solutions obtained using the error formula

$$E_{\text{sol}} = \frac{1}{n_d} \sum_{i,j} \frac{|\hat{d}_{ij} - d_{ij}|}{d_{ij}}, \tag{12.3}$$

where $\hat{d}_{ij}$ is the original distance between the atoms i, j, d_{ij} is the final distance between the points $x^i, x^j \in \mathbb{R}^3$, and n_d is the number of distances used in each test. We employed a Fortran procedure to evaluate the numerical solutions using the formula (12.3). The results are shown in the columns E_{sol} of Table 12.9. According to Table 12.9, we can see that both routines attain good solutions to the problems. *GENCAN* obtains smaller values to the error (12.3), but *MD-jeep* is much faster.

Acknowledgements The authors are thankful to PRONEX-Optimization (PRONEX - CNPq / FAPERJ E-26 / 171.164/2003 - APQ1), FAPESP (Grant 06/53768-0), and CNPq.

References

1. 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)
2. Birgin, E.G., Martínez, J.M.: Large-scale active-set box-constrained optimization method with spectral projected gradients. Comput. Optim. Appl. **23**, 101–125 (2002)
3. Curtis, M.L.: Matrix Groups. Springer, New York (1984)
4. De Leeuw, J.: Differentiability of Kruskal's Stress at a Local Minimum. Psychometrika **49**, 111–113 (1984)
5. Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press (1996)
6. Gower, J.C., Dijksterhuis, G.B.: Procrustes Problems. Oxford University Press (2004)
7. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Am. **4**, 629–642 (1987)
8. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallogr. **A34**, 827–828 (1978)
9. Kearsley, S.K.: On the orthogonal transformation used for structural comparisons Acta Crystallogr. **A45**, 208–210 (1989)
10. Kuipers, J.B.: Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality. Princeton University Press, New Jersey (2002)
11. Liberti, L., Lavor, C., Mucherino, A., Maculan, N.: Distance geometry methods: from continuous to discrete. Int. Trans. Oper. Res. **18**, 33–51 (2010)
12. Lima, R.S.: Representation of rotations: advantages and disadvantages in theory and practice. Master Thesis, Department of Applied Mathematics, IMECC, UniCamp (2007)
13. Mucherino, A., Liberti, L., Lavor, C.: `MD-jeep`: an implementation of a branch and prune algorithm for distance geometry problems. In: Fukuda, K., et al. (eds.) Proceedings of the Third International Congress on Mathematical Software (ICMS10), Lectures Notes in Computer Science, vol. 6327, pp. 186–197. Kobe, Japan (2010)