

Uma nova abordagem para dividir instâncias do *Discretizable Molecular Distance Geometry Problem* usando *gaps*

Felipe Fidalgo, **Carlile Lavor,**

Departamento de Matemática Aplicada
Instituto de Matemática, Estatística e Computação Científica, UNICAMP
13083-859, Campinas, SP
E-mail: felipefidalgo@ime.unicamp.br, clavor@ime.unicamp.br,

Jaime Rodriguez

Departamento de Matemática, FEIS, UNESP
Campus Ilha Solteira
15385-000, Ilha Solteira, SP
E-mail: jaime@mat.feis.unesp.br.

Resumo: *Motivado pelo Parallel BP, que divide instâncias do Discretizable Molecular Distance Geometry Problem (DMDGP) em sub-instâncias com o mesmo número de vértices cada, este trabalho propõe duas divisões mais flexíveis, baseadas nas informações de distâncias disponíveis. Tais abordagens foram desenvolvidas na tentativa de separar as partes com rápida determinação das partes mais lentas, em relação ao custo computacional do Algoritmo Branch & Prune, utilizando os chamados gaps. Resultados computacionais preliminares indicam que abordagem é mais eficiente.*

Palavras-chave: *Geometria de Distâncias, Discretizable Molecular Distance Geometry Problem, Algoritmo Branch & Prune, Parallel Branch & Prune, Gap, g-vértice*

1 Introdução

Unindo certo conhecimento químico, como comprimentos e ângulos de ligações entre átomos, com dados de Ressonância Magnética Nuclear (RMN) [5, 16], pode-se estimar distâncias entre pares de átomos em moléculas. Tais informações induzem a formulação de um problema inverso para determinar conformações tridimensionais, preservando suas estruturas de ângulos e distâncias. Este é chamado de *Molecular Distance Geometry Problem (MDGP)* [1].

Definição 1.1 (MDGP [8]). *Uma molécula M , com n átomos, associa-se a um grafo $G = (V, S, d)$ não-direcionado e ponderado, cujos vértices em V correspondem aos átomos, as arestas em S correspondem às ligações e os pesos das arestas, dados pela função $d : V \rightarrow \mathbb{R}$, associam-se às distâncias disponíveis entre os átomos. É possível determinar uma conformação de M $x : V \rightarrow \mathbb{R}^3$ tal que*

$$\|x(i) - x(j)\| = d_{i,j} \quad \forall (i, j) \in S \quad (1)$$

Cada conformação x recebe o nome de **Realização** de G e, por Saxe [15], este problema é **NP-difícil**. Geralmente formulado como um problema de otimização global contínua [12], ele possui vários métodos de resolução por aproximação [11].

Além dessas duas representações, cada instância $G = (V, S, d)$ do MDGP pode ser associada a uma matriz simétrica D_G chamada de **Matriz euclidiana de Distâncias** [11], onde

$$D_G = (D_G)_{ij} = (D_G)_{ji} = \begin{cases} 0, & \text{se } i = j \quad \text{ou } \{i, j\} \notin S \\ d_{ij}, & \text{se } \{i, j\} \subset S \end{cases} \quad (2)$$

O MDGP ainda possui uma generalização que lida com tópicos não-moleculares, como localização de redes de sensores sem-fio, Estática, Visualização de Dados, Robótica, dentre outros [11]. No ambiente molecular, todavia, os compostos que mais despertam o interesse de pesquisadores de diversas áreas são as Proteínas, devido à íntima ligação que suas conformações tridimensionais têm com a importância vital de suas funções nos seres vivos. Esse conhecimento pode conduzir a descobertas de grande interesse e benefício para a Humanidade como, por exemplo, a criação de novas drogas. Isto dá sentido à assídua busca por tais estruturas.

Neste sentido, notou-se que as Proteínas satisfazem duas hipóteses adicionais ao MDGP em geral. Isso induz uma relação de ordem total em seus vértices, discretizando o espaço de busca por conformações factíveis e o tornando finito. Dessa maneira, este pode ser estruturado como uma árvore binária, cujos nós correspondem a pontos em \mathbb{R}^3 provenientes da intersecção de três esferas no espaço euclidiano tridimensional [3]. Com isso, a busca por essas estruturas torna-se mais rápida e eficiente. Deve-se, então, a essa razão e importância, a criação de um subproblema, chamado **Discretizable Molecular Distance Geometry Problem (DMDGP)** [8], cuja definição é estabelecida como segue:

Definição 1.2 (DMDGP [3, 8]). Considere o subconjunto de instâncias $G = (V, S, d)$ do MDGP para as quais o subgrafo $G[U_0]$, associado ao conjunto dos três vértices iniciais $U_0 = \{v_1, v_2, v_3\}$, forma uma 3-clique em G . Além disso, valem as seguintes hipóteses:

- (1) para todo vértice $v \notin U_0$, o subgrafo $G[U_v]$, que está associado ao conjunto de quatro vértices $U_v = \{v - 3, v - 2, v - 1, v\}$, forma uma 4-clique em G e,
- (2) para todo $v \in \{3, \dots, |V|\}$, são válidas as desigualdades triangulares estritas abaixo

$$d(v - 2, v) < d(v - 2, v - 1) + d(v - 1, v).$$

Este conjunto chama-se **Discretizable Molecular Distance Geometry Problem**.

Este DMDGP também é NP-difícil [8]. Entretanto, a estrutura binária do seu espaço de busca por soluções forneceu, quase que intrinsecamente, um método recursivo eficiente para explorá-lo. Este é chamado de Algoritmo **Branch & Prune (BP)** e converge em tempo polinomial para a maioria das Proteínas, as quais são instâncias do DMDGP, como em Liberti et al. [10].

Ele é inicializado com a determinação única dos três primeiros vértices para fixar o sistema de coordenadas, a partir das distâncias entre si pois formam uma 3-clique por hipótese.

A partir disso, avança-se um nível por vez com a determinação das duas possíveis posições, x'_i e x''_i , para o próximo vértice i . Isto decorre das informações de distâncias disponíveis na clique formada pelos vértices $i - 3, i - 2, i - 1, i$, fato que também procede das hipóteses do DMDGP. Tais posições são determinadas a partir da intersecção de três esferas cujos raios são dados pelas **Distâncias de Discretização** $d_{j,i}$, com $i - 3 \leq j \leq i - 1$, que sempre estão disponíveis [11]. Este processo é chamado de **Branching**.

Agora, caso haja disponível alguma **Distância de Poda** $d_{j,i}$ entre o vértice i e algum outro vértice $j \in \{1, \dots, i - 4\}$ [11], ocorre um processo chamado **Pruning** (ou Poda): um dos dois galhos possíveis, gerados por x'_i e x''_i , é podado. Isso acontece devido à sua infactibilidade em relação a tal valor de distância de poda: temos quatro esferas se intersectando neste momento, gerando apenas um ponto de intersecção [7, 8]. Portanto, o outro ponto é descartado.

Estes dois procedimentos andam em conjunto, seguindo um caminho da raiz até a folha. Caso encontre-se alguma infactibilidade, o BP descarta aquele caminho e realiza o chamado **backtracking**: ele regressa ao último vértice que possui algum caminho ainda disponível procedendo a busca em outros galhos até que encontre uma solução com n vértices. Quanto mais distâncias de poda estiverem disponíveis, menos nós terá a árvore binária de soluções.

Além disso, notou-se que, a partir do quarto nível, árvore binária de soluções de um DMDGP pode ser dividida em duas partes igualmente simétricas. Portanto, explora-se apenas metade desta árvore através do algoritmo BP [8], fixando também o quarto vértice de modo único.

Um *outline* deste método está descrito no Algoritmo 1.

Algorithm 1 O Algoritmo *Branch & Prune* (com simetria) [3, 8].

```

1: Determina-se unicamente os quatro primeiros vértices a partir das distâncias.
2: BP( $i, n, d$ )
3: Calcule  $x'_i$ ;
4: se ( $x'_i$  é factível) então
5:   se ( $i = n$ ) então
6:     nsols = nsols + 1;
7:   senão
8:     BP( $i + 1, n, d$ )
9:   fim se
10: fim se
11: Calcule  $x''_i$ ;
12: se ( $x''_i$  é factível) então
13:   se ( $i = n$ ) então
14:     nsols = nsols + 1;
15:   senão
16:     BP( $i + 1, n, d$ )
17:   fim se
18: fim se

```

2 Uma versão em paralelo

Para paralelizar essa determinação e torná-la mais eficiente, Mucherino et al. [13] introduziu uma forma de dividir uma instância genérica do DMDGP em sub-instâncias. Desse modo, os subconjuntos de restrições de distâncias respectivos constituem sub-problemas locais, resolvidos pelo BP, usando somente distâncias “locais” em seus processos independentes. Este procedimento gera um conjunto de múltiplas árvores de realizações para cada pedaço da divisão. Entretanto, deve haver valores de distâncias entre átomos que estejam em diferentes processos, as quais não foram utilizadas ainda nenhuma vez pelo BP para realizar alguma poda de posições infactíveis. Estas são utilizadas *a posteriori*, quando iniciar-se o processo de junção desses pedaços.

Para realizar a divisão para uma instância com n átomos, escolhe-se o número p de sub-instâncias como blocos de n/p átomos. Se n não é perfeitamente divisível por p , o resto r dos átomos são acomodados em um novo processo. Inicia-se, então, a junção desses processos em um único, utilizando o esquema *cascata* como se pode ver na Figura 1 [13]. Cada solução de processos distintos estão em um sistema de coordenadas em \mathbb{R}^3 . Logo, para uni-las, utiliza-se uma translação e duas rotações, garantindo que sejam representadas em termos das mesmas coordenadas. A parte de unir as sub-soluções não é feita em paralelo [13].

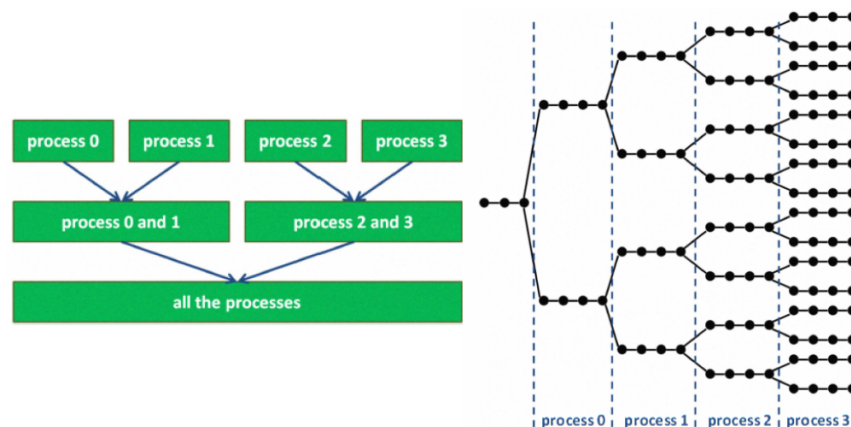


Figura 1: Esquema cascata para união dos processos independentes.

3 Dividindo uma instância utilizando *gaps*

A versão paralela do *Branch & Prune* utiliza-se de uma maneira rígida para determinar o número de sub-instâncias a serem realizadas e, logo, o número fixo e igual de vértices em cada uma delas. Considerando a diminuição do tempo total para as realizações com o BP, é possível que essa não seja a melhor maneira de se fazer uma divisão, levando em conta que cada instância possui suas características geométricas de distâncias e ângulos próprias, localmente variáveis.

Neste sentido, nossa contribuição neste trabalho é propor uma maneira de dividir uma instância do DMDGP em sub-instâncias, utilizando as próprias informações de distâncias interatômicas de cada uma. O objetivo dessa flexibilização é separar partes mais “rápidas”, na determinação pelo BP, das partes mais lentas, colocando-as em processos diferentes. Para permitir, entretanto, que as realizações de cada processo sejam unidas duas-a-duas por meio de transformações euclidianas [2], essa divisão deve, primeiramente, garantir que cada par de sub-instâncias consecutivas possuam três vértices em comum.

Considere a notação de [2], onde cada sub-instância é dada por um intervalo de vértices consecutivos com intersecção não-vazia entre pares consecutivos deles. Seja, então, M uma instância do DMDGP com n vértices e uma ordem $<$ total em seu conjunto de átomos. Para tal intersecção, considere uma divisão de M como uma união crescente de intervalos

$$M = M_1 \cup M_2 \cup \dots \cup M_k, \tag{3}$$

onde $M_j = [a_j, b_j]$, $a_1 = 1$, $b_k = n$, $1 \leq a_j \leq a_{j+1} \leq n$, $b_j - a_{j+1} \geq 2$ e $j = 2, \dots, k - 1$. Adicionalmente, seja $G = (V, S, d)$ é seu grafo associado. Nesta notação, a instância toda pode ser denotada pelo intervalo $M = [1, n]$. Além disso, cada intervalo M_j está associado a um subgrafo $G_j = (V_j, S_j, d)$ de G , para o qual o BP fornece a árvore binária T_j de realizações.

Seja, agora, i um vértice de G . Pelo BP, temos duas posições possíveis x'_i e x''_i em \mathbb{R}^3 para i . Caso haja alguma distância adicional (de poda) $d_{j,i}$, i deve possuir apenas uma delas como factível [7]. Isso deve-se ao fato de que, localmente, sua determinação se resume à intersecção de quatro esferas em \mathbb{R}^3 [8]. Tome, então, o conjunto $U_i = \{j \in V : \{j, i\} \in S, j < i\}$ que engloba todos os vértices cujas distâncias até i são conhecidas. Por definição, $|U_i| \geq 3$. Lavor et al. [7] demonstraram que, caso $|U_i| \geq 4$, então existe apenas uma posição factível. Um vértice com essa propriedade e um conjunto deles recebem nomes especiais.

Definição 3.1 (*g*-vértice, *k*-gap). *Seja $G = (V, S, d)$ uma instância do DMDGP. A todo vértice $i \in V$ tal que $|U_i| \geq 4$, chamamos de **g-vértice**. Um conjunto Γ_k contendo k g-vértices consecutivos é chamado de *k*-gap.*

Para cada *g*-vértice, a árvore final de soluções ramifica no dobro de posições possíveis, pois não há poda nestes galhos. Portanto, ao final de cada *k*-gap, o número de folhas nessa árvore é um múltiplo de 2^k . Essa ramificação torna o BP mais lento, pois leva muito tempo até podar novamente, realizando um *backtracking* muito longo. Nossa proposta é diminuir o tamanho e a quantidade dos backtrackings ao dividir em intervalos.

Considere um exemplo com dez vértices, representado pela matriz de distâncias abaixo

$$D_G = \begin{bmatrix} 0 & d_{1,2} & d_{1,3} & d_{1,4} & 0 & 0 & 0 & 0 & 0 & d_{1,10} & 0 \\ d_{1,2} & 0 & d_{2,3} & d_{2,4} & d_{2,5} & 0 & 0 & 0 & 0 & 0 & d_{2,11} \\ d_{1,3} & d_{2,3} & 0 & d_{3,4} & d_{3,5} & d_{3,6} & 0 & 0 & 0 & 0 & 0 \\ d_{1,4} & d_{2,4} & d_{3,4} & 0 & d_{4,5} & d_{4,6} & d_{4,7} & 0 & 0 & d_{4,10} & 0 \\ d_{1,5} & d_{2,5} & d_{3,5} & d_{4,5} & 0 & d_{5,6} & d_{5,7} & d_{5,8} & 0 & 0 & d_{5,10} \\ 0 & 0 & d_{3,6} & d_{4,6} & d_{5,6} & 0 & d_{6,7} & d_{6,8} & d_{6,9} & d_{6,10} & 0 \\ 0 & 0 & 0 & d_{4,7} & d_{5,7} & d_{6,7} & 0 & d_{7,8} & d_{7,9} & d_{7,10} & 0 \\ 0 & 0 & 0 & 0 & d_{5,8} & d_{6,8} & d_{7,8} & 0 & d_{8,9} & d_{8,10} & d_{8,11} \\ 0 & 0 & 0 & 0 & 0 & d_{6,9} & d_{7,9} & d_{8,9} & 0 & d_{9,10} & d_{9,11} \\ d_{1,10} & 0 & 0 & d_{4,10} & 0 & d_{6,10} & d_{7,10} & d_{8,10} & d_{9,10} & 0 & d_{10,11} \\ 0 & d_{2,11} & 0 & 0 & d_{5,10} & 0 & 0 & d_{8,11} & d_{9,11} & d_{10,11} & 0 \end{bmatrix} \tag{4}$$

Para avaliarmos a existência de um g -vértice, basta considerar a parte superior à diagonal principal de D_G . Portanto, os vértices 5, 6, 7, 8 e 9 são os únicos g -vértices no exemplo. Além disso, dada a consecutividade, o conjunto $\Gamma_5 = \{5, 6, 7, 8, 9\}$ forma um 5- gap .

Em relação à “velocidade” na determinação de posições para esta instância, do vértice 1 até o vértice 4 existe apenas uma conformação possível a ser determinada rapidamente pelo BP pois os *backtrackings* são curtos e tão logo uma solução é encontrada. Já durante o gap Γ_5 , o BP demora mais tempo para fazer as $2^5 = 32$ ramificações, já que não existem testes de poda em nenhum desses vértices, e então podar no décimo vértice, ocorrendo *backtrackings* muito longos. Por fim, o BP também determina a única conformação possível para os vértices 10 e 11 de forma rápida.

Nossa proposta, então, é dividir a instância G em pedaços cujos cortes ocorram durante os *gaps*, sem permitir que hajam muitos intervalos, pois realizar as junções tem um alto custo computacional de realizar três transformações euclideinas por vez [2]. Considere, dessa maneira, as três propostas de divisão abaixo para o exemplo meramente ilustrativo.

(1) **B-split** - o corte é realizado no primeiro vértice de cada *gap*. Para o exemplo, teríamos

$$V = [1, 5] \cup [3, 10]. \quad (5)$$

(2) **M-split** - o corte é realizado no vértice intermediário de cada *gap*. No exemplo, teríamos

$$V = [1, 7] \cup [5, 10]. \quad (6)$$

(3) **E-split** - o corte é feito no último vértice de cada *gap*. No exemplo teríamos

$$V = [1, 9] \cup [7, 10]. \quad (7)$$

Algumas características dessas maneiras de corte podem ser elencadas, em relação ao número de *backtrackings* realizados, inferindo sobre a velocidade na determinação das estruturas com o BP.

- Se lançarmos mão do *E-split*, todo o *gap* estará contido no primeiro intervalo. Assim, a árvore de sub-realizações terá muitos ramos, que é algo que não desejamos para fazer as junções. É a situação máxima em número de ramos, pois não houve nenhuma poda. Além disso, muitas informações de poda ficaram para posterior uso (após as colagens), podendo implicar na existência de *backtrackings* longos para o intervalo seguinte. Por esses argumentos, descartamos de antemão esse modo de fazer as divisões.
- Utilizando o *B-split*, teremos mais informações de poda para cada intervalo subsequente e, portanto, mais *backtrackings* a realizar. O primeiro intervalo deve ficar mais curto e com menos ramos, mas a quantidade de podas que haverá no segundo pedaço é muito grande, já que quase nenhuma informação de poda ficou de fora. A probabilidade é que o tempo para realizar o intervalo com tempo máximo seja grande, em relação a outra divisão.
- Por fim, utilizando o *M-split*, temos realmente uma situação mediana. Os intervalos possuem ramos com uma boa divisão em sua quantidade, já que os vértices sem poda são igualmente divididos para cada intervalo. Além disso, nem se inclui muitas distâncias de poda no intervalo subsequente, para haver muitos *backtrackings* como no *B-split*, nem se inclui poucas delas, para haver *backtrackings* muito longos como no *E-split*. Aparentemente, essa divisão traz uma sensação maior de equilíbrio.

4 Resultados Computacionais

Apresentamos nesta seção alguns resultados computacionais preliminares dos experimentos com as três divisões apresentadas na seção anterior. Todos os experimentos foram feitos com instâncias DMDGP geradas artificialmente, como em Lavor [6], com um corte de 4Å no conjunto de distâncias, a fim de simular dados de RMN. As rotinas foram implementadas em MATLAB e executadas em um computador com um processador Intel Core i5 com 1.6 GHz e memória RAM

de 2GB. As instâncias computadas têm todos os tamanhos possíveis, de 100 vértices até 10000 vértices, a fim de escolher um dos métodos que necessita de menos tempo de computação.

Para cada instância dividida em k intervalos, foram aferidos o tempo de CPU de cada intervalo e o maior desses resultados foi tomado como o tempo de determinação da estrutura, já que os processos devem ocorrer, teoricamente, em paralelo, apesar de nenhum teste em paralelo ter sido realizado ainda. Parte de tais resultados estão mostrados na tabela seguinte com quatro colunas: a primeira armazena a quantidade de vértices da estrutura DMDGP artificial, as segunda e terceira colunas armazenam os tempos t_B e t_M , em segundos, gastos na determinação com as divisões B-split e M-split, respectivamente. Por fim, a última coluna abriga um valor relativo entre os tempos, a fim de medir a eficiência de um em relação ao outro.

Tabela 1: Tabela com os dados de tempo de determinação usando as duas divisões de uma instância DMDGP.

No. Vértices	t_B	t_M	t_M/t_B
100	0,0380 s	0,0154 s	40,52%
500	0,2852 s	0,1426 s	50,00 %
1000	1,1147 s	0,5444 s	48,83%
5000	2,0982 s	0,6000 s	28,59%
10000	14,1819 s	5,1340 s	36,20%

Pelos valores de tempo demonstrados na tabela, em todos os casos testados, a determinação a partir da divisão M-split é a que acontece em um tempo menor. Para a instância com 5000 vértices, por exemplo, a primeira divisão gastou 2.0982 s como tempo máximo entre os intervalos, enquanto que a segunda divisão gastou 0.6000 s. Em ambos os casos, para todos os experimentos, o número de intervalos é o mesmo, variando apenas de tamanho. A sensação de equilíbrio registrada na seção anterior se justifica com a medição dos tempos de CPU para este grupo de instâncias.

5 Conclusão e Trabalhos Futuros

Neste trabalho, foram apresentados duas divisões possíveis para uma instância DMDGP a fim de aplicar a mesma idéia de Mucherino et al. em [13], partes da mesma instância submetidas ao Algoritmo *Branch & Prune* em paralelo, já que este propôs uma divisão rígida. As divisões aqui apresentadas, B-split e M-split, dependem de cada estrutura de distâncias das instâncias, sendo baseadas nos chamados *gaps* no conjunto de vértices. Os resultados computacionais mostram que a segunda divisão tem suas partes determinadas pelo BP em mais tempo do que a primeira divisão. Como trabalhos futuros, procederemos a comparação dessa divisão M-split com a divisão rígida proposta em [13] e avaliaremos a possibilidade de utilizar os *gaps* para flexibilizar ainda mais a escolha dos cortes, a fim de diminuir de maneira ótima a quantidade de *backtrackings* realizados.

Agradecimentos

Os autores agradecem às agências brasileiras de fomento CNPq, CAPES e FAPESP pelo suporte financeiro, além das Universidades Estaduais de Campinas e Paulista. Os agradecimentos são estendidos ao colaborador Douglas Gonçalves (IRISA, Université de Rennes 1) pelas preciosas discussões.

Referências

- [1] G. Crippen, T. Havel, “Distance Geometry and Molecular Conformation” , Research Studies Press, UK, 1988.
- [2] F. Fidalgo, J. Rodriguez, Quaternions as a tool for merging multiple realization trees, Proceedings of the Distance Geometry and Applications Workshop (DGA13), Manaus, Brazil, 2013.
- [3] W. Gramacho, A. Mucherino, C. Lavor, N. Maculan, A Parallel BP Algorithm for the Discretizable Distance Geometry Problem, IEEE Conference Proceedings, Workshop on Parallel Computing and Optimization (PCO12), 26th IEEE International Parallel and Distributed Processing Symposium (IPDPS12), Shangai, China, 1756-1762, 2012.
- [4] D. Grant, R. Harris, “Encyclopedia of Nuclear Magnetic Resonance”, Wiley, New York, 1995.
- [5] T. Havel, Distance Geometry. In Grant et al. [4], 1701-1710, 1995.
- [6] C. Lavor, On generating instances for the molecular distance geometry problem. In: L. Liberti et al. [9], 405 - 414, 2006.
- [7] C. Lavor, J. Lee, A. Lee-St.John, L. Liberti, A. Mucherino, M. Sviridenko, Discretization Orders for Distance Geometry Problems, *Optimization Letters*, 6 (2012) 783-796.
- [8] C. Lavor, L. Liberti, N. Maculan, A. Mucherino, The discretizable molecular distance geometry problem, *Computational Optimization and Applications*, 52 (2012) 115-146.
- [9] L. Liberti, C. Lavor, “Global Optimization: From Theory to Implementation”, Springer, Berlin, 2006.
- [10] L. Liberti, C. Lavor, N. Maculan, A Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem, *International Transactions in Operational Research* 15 (2008) 1-17.
- [11] L. Liberti, C. Lavor, N. Maculan, A. Mucherino, Euclidean Distance Geometry and Applications, *SIAM Review* 56 (2014) 3-69.
- [12] J. Moré, Z. Wu, Global Continuation for Distance Geometry Problems, *SIAM Journal on Optimization* 7 (1997) 814-836.
- [13] A. Mucherino, C. Lavor, L. Liberti, E-G. Talbi, A Parallel Version of the Branch-and-Prune Algorithm for the Molecular Distance Geometry Problem, IEEE Conference Proceedings, ACS/IEEE International Conference on Computer Systems and Applications (AICCSA10), Hammamet, Tunisia, 1-6, 2010.
- [14] A. Mucherino, C. Lavor, L. Liberti, N. Maculan, “Distance Geometry: Theory, Methods and Applications, Springer, New York, 2013.
- [15] J.B.Saxe, Embeddability of Wighted Graphs in k -Space is Strongly NP-Hard, Proceedings of 17th Allerton Conference in Communications, Control and Computing, 480-489,1979.
- [16] T. Schlick, “Molecular modeling and simulation: an interdisciplinary guide”, Springer, New York, 2002.