

Proposta de modelo de programação não-linear inteira mista para o *unassigned distance geometry problem* considerando imprecisões no conjunto de distâncias

Petra Maria Bartmeyer

Faculdade de Engenharia Elétrica - Unicamp
Av. Albert Einstein - 400 Cidade Universitária - Campinas/SP
petra.bartmeyer@gmail.com

Christiano Lyra Filho

Faculdade de Engenharia Elétrica - Unicamp
Av. Albert Einstein - 400 Cidade Universitária - Campinas/SP
chrlyra@densis.fee.unicamp.br

RESUMO

Propomos um modelo de otimização não-linear inteiro misto para o *unassigned distance geometry problem*, considerando a possibilidade de imprecisões no conjunto de distância. Esta é uma primeira exploração sobre o desempenho deste tipo de modelo na resolução do problema. Como o modelo contém variáveis binárias combinadas a um conjunto de restrições quadráticas, o trabalho propõe uma alternativa de relaxação para as variáveis binárias. A relaxação proposta permite a obtenção de soluções muito próximas a soluções inteiras através do emprego de um solver para otimização de problemas com variáveis contínuas, o que permite dispensar a etapa de factibilização inteira comum em métodos de enumeração implícita, e possibilita a redução dos tempos de solução. Testes computacionais avaliaram o desempenho do modelo e sua sensibilidade em relação à dimensão das instâncias e aos parâmetros usados na relaxação.

PALAVRAS CHAVE. *Unassigned distance geometry problem*, programação não-linear, relaxação quadrática

ABSTRACT

This work proposes a mixed-integer linear optimization model for the *unassigned distance geometry problems* considering data inaccuracy. It is an initial exploration of the performance of the exact approach to solving the problem. Due to the presence of binary variables combined with a set of quadratic constraints, we apply a quadratic relaxation to handle binary variables. The binary relaxation achieves a closer binary solution with the use of a solver for continuous variables avoiding the feasibility steps from implicit enumeration methods, reducing the solution time. Computational studies analyze the performance of the model related to the instance size and the relaxation parameter.

PALAVRAS CHAVE. *Unassigned distance geometry problem*, non-linear integer programming, mathematical modeling

1. Introdução

O problema de geometria de distâncias (PGD) é um dos temas de problemas inversos [Liberti et al., 2014] onde, dado um conjunto de distâncias $\mathcal{D}$, deseja-se encontrar a posição dos pontos cujas distâncias correspondem aos elementos de $\mathcal{D}$. Ao longo do trabalho, o PGD será representado com um problema em grafos de vértices pertencentes ao espaço euclidiano de dimensão $K > 0$ e as distâncias calculadas segundo a norma euclidiana denotada por $\|\cdot\|$.

A caracterização do PGD depende das informações atribuídas ao grafo $G = (V, E)$. Para os problemas *assigned geometry distance problem* (aDGP), além do conjunto $\mathcal{D}$, é fornecido a aresta $(i, j) \in E$ que corresponde a cada uma das distâncias de $\mathcal{D}$. Ou seja, define-se o grafo $G = (V, E)$ de vértices em $i, j \in V$ e arestas $(i, j) \in E$. Nesse caso, a solução do problema é a posição $x_i \in \mathbb{R}^k$ e $x_j \in \mathbb{R}^k$ dos vértices i e j que satisfazem a distância $d_a \in \mathcal{D}$ referente a aresta $(i, j) \in E$.

Esse problema é caracterizado como um problema de satisfatibilidade $\|x_i - x_j\|^2 = d_a^2, \forall ij$. A conversão dele para um problema de otimização irrestrita pode ser feita como em (1). Observa-se que função objetivo tem valor zero somente quando cada termo é igual a zero, uma vez que $(d_a^2 - \|x_i - x_j\|^2)^2 \geq 0$, ou seja, a função objetivo só é nula quando todas as distâncias entre pares de vértices forem satisfeitas.

$$\min \sum_{i=1}^n \sum_{j=1}^n (d_{ij}^2 - \|x_i - x_j\|^2)^2 \quad (1)$$

Já para o *unassigned geometry distance problem* (uDGP), a informação sobre as arestas não é fornecida a priori. O problema pode ser caracterizado da seguinte forma: dado um conjunto de distâncias $\mathcal{D} = \{d_1, \dots, d_m\}$, e uma quantidade pré-fixada de vértices n , o objetivo é determinar a posição $x_i \in \mathbb{R}^k, \forall i = 1, \dots, n$ dos n vértices do grafo e a designação de cada uma das distâncias do conjunto $\mathcal{D}$ para uma única aresta do grafo. A solução do problema fornece o conjunto de arestas $(i, j) \in E$, além das posições dos vértices $i, j \in V$.

Ambos os problemas apresentam aplicações em áreas como de detecção de estruturas de nanopartículas [Billinge et al., 2016] e proteínas [Moré e Wu, 1999; Liberti et al., 2011], e localização de sensores em redes wireless [Cao et al., 2006; Biswas e Ye, 2006]. As aplicações trazem aspectos adicionais para os problemas, associados a dimensão do espaço em que os vértices estão localizados e a confiabilidade na acurácia do conjunto de distâncias [Lavor et al., 2019; Islam et al., 2019]. A literatura sobre esses desdobramentos é mais abrangente quando se trata do aDGP, o que fundamenta a afirmação de Liberti e Lavor [2018] de que o uDGP é um tema desafiador e em aberto na área de geometria de distâncias.

Este trabalho estuda a abordagem do uDGP por meio de um modelo de programação não-linear não-convexa inteira mista, considerando imprecisões no conjunto de distâncias. No melhor do nosso conhecimento, a literatura na área ainda não registra investigações nessa direção. Por outro lado, resultados recentes na abordagem de problemas com características semelhantes motivam a investigação proposta [Ye et al., 2019; Galvan et al., 2019].

Os obstáculos encontrados na resolução do modelo de programação não-linear não-convexa aumentam com a existência de variáveis inteiras. As técnicas mais bem estabelecidas para a resolução de problemas de otimização com variáveis inteiras são técnicas de enumeração implícita, cujo custo computacional cresce com número de variáveis. A dificuldade de solução ainda cresce devido a existência de variáveis inteiras associadas a restrições não-lineares, pois um sub-problema de programação não-linear deve ser resolvido para cada variável no processo de enumeração implícita.

Para os casos onde o subproblema é resolvido por técnicas que usam derivada, o custo de armazenamento das matrizes envolvidas pode também não ser viável. Em resumo, a natureza combinatória do problema faz com que custo computacional sempre cresça em função da quantidade de variáveis inteiras.

A abordagem explorada no trabalho procura reduzir o custo computacional decorrente da existência das variáveis inteiras binárias através da relaxação dessas variáveis, transformando o problema original em um problema de programação não-linear não-convexo. A relaxação adotada transforma todas as variáveis binárias em variáveis contínuas através da inclusão de uma única restrição quadrática. O modelo relaxado pode ser abordado por técnicas de programação não-linear não-convexa, mas inclui aproximações que obriga a inclusão de uma análise a posteriori sobre a factibilidade e qualidade da solução encontrada.

A avaliação foi realizada por meio de experimentos computacionais com uma instância da literatura. A instância contém todas as distâncias entre cada par de átomos de C_{60} . Essa instância é composta por 21 distâncias distintas, onde cada distância pode se repetir entre 30 e 120 vezes no conjunto $\mathcal{D}$ — o número de repetições é denominado multiplicidade da distância. Os resultados analisam o *tradeoff* entre qualidade da relaxação e a interferência dela no valor da função objetivo.

Ao final, apresentamos testes computacionais com quatro instâncias sintéticas simulando a posição de moléculas em um espaço de três dimensões. Esses testes ilustram a relação entre o tempo de resolução e a dimensão da instância, e mostram como o conjunto de restrições não-lineares torna o modelo atrativo para instâncias com multiplicidade no conjunto de distâncias.

Este trabalho dá seguimento da seguinte maneira. A Seção 2 define o *unassigned distance geometry problem* e a apresenta existência de imprecisões no conjunto de distâncias. A Seção 3 trás a abordagem proposta para o problema e a modelagem matemática aplicada, apresentando a relaxação utilizada e a interação da mesma com o modelo utilizado. O estudos computacionais são apresentados na Seção 4. A discussão dos resultados e as conclusões do trabalho são apresentadas na Seção 5.

2. Unassigned Distance Geometry Problem

A solução do uDGP encontra a designação dos pares de vértices $i, j \in V$ a uma única aresta $(i, j) \in E$ de valor $d_a \in \mathcal{D}$. Dada a designação, a posição dos vértices no espaço de dimensão k deve satisfazer $\|x_i - x_j\| = d_a$ para todos os vértices do grafo $G = (V, E)$. Essa caracterização não leva em conta as imprecisões possível na lista de distâncias. Para esses casos, temos que $\|x_i - x_j\| = d_a$ será substituído por $\underline{d}_a \leq \|x_i - x_j\| \leq \bar{d}_a$, onde $\underline{d}_a$ e $\bar{d}_a$ são os limites inferior e superior do intervalo de distância. Existem métodos exatos, heurísticos e híbridos para abordar o uDGP com e sem imprecisões.

Duas abordagens heurísticas conhecidas para a solução do uGDP se apresentam como algoritmos de construção de solução [Duxbury et al., 2016]. A primeira abordagem é denominada TRIBOND [Gujarathi et al., 2014]. Esse algoritmo possui duas etapas, na primeira encontra um *core* para o grafo, ou seja, dentre todos os possíveis arranjos ele deve selecionar o conjunto de $(k + 2)$ cliques que para o qual a distâncias entre os vértices corresponda as distâncias do conjunto $\mathcal{D}$. A partir desse *core* os demais vértices são adicionados ao grafo verificando a factibilidade das distâncias. Caso não haja um vértice a ser adicionado que satisfaça o conjunto de distância o processo é reiniciado a partir de um novo *core*.

A segunda abordagem é denominada LIGA [Juhás et al., 2006] e é um algoritmo estocástico capaz de tratar casos com distâncias imprecisas ou ainda uma lista de distâncias incompletas. Constituído de dois passos, onde o primeiro dele é encontrar um *core*. A partir do *core*, os

outros vértices são adicionados um a um. A escolha do vértice a ser adicionado é feita por meio de uma função de custo.

Mesmo que as abordagens heurísticas tenham tomado a frente na solução do uDGP, abordagens exatas continuam a ser estudadas e algumas delas fazem parte dos algoritmos LIGA e TRIBOND. Algumas das técnicas de otimização global empregadas no problema uDGP são algoritmos do tipo *branch-and-bound* espacial [Liberti e Maculan, 2006], de busca em vizinhança [Hansen e Mladenović, 2001], e de técnicas de suavizações progressivas [Moré e Wu, 1999].

No que se refere a modelos matemáticos, uma caracterização geral é dada a seguir [Duxbury et al., 2016]:

$$\min_{\alpha: L \rightarrow E} [\min_x \sum f(|x(u) - x(v)| - d_{uv})],$$

onde $f \in [0, +\infty]$ é uma função de penalização, $x(u)$ é a u -ésima coluna da matriz $X \in \mathbb{R}^{N \times K}$, onde N é o número de vértices e K a dimensão do espaço onde esses vértices se encontram. O mapeamento bijetivo entre um par de vértices e uma aresta do grafo é realizado por $\alpha: L \rightarrow E$.

Uma outra modelagem interessante é proposta em Huang e Dokmanić [2018] e é capaz de resolver instâncias geradas aleatoriamente com até 100 vértices. Nessa abordagem, o conjunto de distâncias é utilizado para estimar uma função de densidade de probabilidade para uma distribuição de distâncias Q , as possíveis posições dos vértices são discretizadas e representadas por meio de uma matriz A . Dessa forma, o modelo proposto busca a alocação dos vértices que minimiza a diferença entre a função de distribuição Q e as distâncias calculadas pelos vértices designados a cada posição referenciada pela matriz A .

3. Proposta e Modelagem Matemática

Esse trabalho propõe um modelo matemático para o uDGP que minimiza as diferenças entre os elementos de $\mathcal{D}$ e a distância entre vértices do grafo, como explicitado na Equação (1). Assim, para o caso onde essa diferença é zero temos que todas as distâncias foram satisfeitas e a solução é ótima. Para os casos onde a função objetivo é diferente de zero, e a solução é um ótimo global, então o problema contém imprecisões nos dados.

O modelo possui três conjuntos de variáveis, variáveis relacionadas a posição dos vértices, variáveis de designação entre vértices e arestas e o último são as variáveis relacionadas aos desvios. O conjunto das variáveis relacionadas às coordenadas i -ésimo vértice é representado por x_i . O conjunto das variáveis binárias y_{aij} relacionadas a designação de uma aresta (i, j) do grafo a uma distância d_a pertencente ao conjunto $\mathcal{D}$, sendo que a cada distância d_a está relacionada a uma multiplicidade $k_a \in \mathbb{N}$. E por último um conjunto de variáveis p_{aij} e n_{aij} que representam a diferença entre a distância dos vértices i e j e a distância d_a .

- x_i é o vetor de entradas x_{1i}, x_{2i}, x_{3i} correspondente a cada uma das coordenadas espaciais do vértice i no espaço $\mathbb{R}^3$, onde $x_{ki} \geq 0, \forall i, k$;
- $y_{aij} = 1$ caso a aresta entre os vértices i e j esteja designada à distância d_a , zero caso contrário;
- p_{aij} e n_{aij} são as diferenças positivas ou negativas, respectivamente, entre a distância calculada entre os vértices i e j a distância d_a a qual foram designados. Nesse trabalho essas diferenças serão denominadas desvios; $p_{aij}, n_{aij} \geq 0$.

$$\min \sum_{a=1}^m \sum_{i=1}^n \sum_{j=1}^n (p_{aij} + n_{aij}) \quad (2)$$

$$\text{s.a} \quad k_a d_a \sum_{i=1}^n \sum_{j=1}^n y_{aij} = \sum_{i=1}^n \sum_{j=1}^n (p_{aij} - n_{aij} + \|x_i - x_j\|), \forall a = 1, \dots, m \quad (3)$$

$$\sum_{a=1}^m y_{aij} = 1, \forall i, j = 1, \dots, n \quad (4)$$

$$\sum_{i=1}^n \sum_{j=1}^n y_{aij} = k_a, \forall a = 1, \dots, m \quad (5)$$

$$x_i \geq 0, n_{ij} \geq 0, p_{ij} \geq 0, y_{aij} \in \{0, 1\} \quad (6)$$

A função objetivo (2) minimiza a diferença entre a distância $d_a \in \mathcal{D}$ e a distância entre os vértices x_i e x_j designados a distância d_a pela variável y_{aij} .

A restrição (3) permite que hajam desvios para cada par de vértices $x_i, x_j \forall i, j = 1, \dots, n$. Observe que esse conjunto não cresce com o número de vértices da instância e sim com o número de distâncias distintas. Esse conjunto de restrições é não-linear e não-convexo. O conjunto de restrições (4) restringe o número de designações de um par i, j a uma única distância d_a . O conjunto de restrições (5) atribui a cada distância $d_a \in \mathcal{D}$ a exatamente k_a arestas $(i, j) \in E$, onde k_a é a multiplicidade da distância d_a .

Devido a igualdade do conjunto de restrições (4), a modelagem apresentada necessita que o número de distâncias seja igual ao número de arestas. Contudo, sabemos da possibilidade de instâncias onde o conjunto de distâncias não é completo, ou seja, nem toda aresta do grafo terá um correspondente em $\mathcal{D}$. Para poder abordar esses casos, adaptamos a dimensão de $\mathcal{D}$ para que uma distância artificial seja adicionada e as arestas faltantes serão designadas para essas nova distância. Na formulação contendo a distância artificial, os desvios em relação a distância artificial não fazem parte da função objetivo, e o conjunto de restrições (4) passa a ser de desigualdade. Essa distância artificial será designada para as arestas que não existem no problema original, sem perda de generalidade, essa distância artificial será denotada por d_{m+1} . Assim, o conjunto de restrições (3) passaria a ser escrito como:

$$k_a d_a \sum_{i=1}^n \sum_{j=1}^n y_{aij} = \sum_{i=1}^n \sum_{j=1}^n (p_{aij} - n_{aij} + \|x_i - x_j\|), \forall a = 1, \dots, m + 1.$$

onde k_a para $a = m + 1$ é igual ao número de distâncias faltantes para completar o conjunto de distâncias.

Algumas simplificações do modelo vem das observações de que: $d_{ij} = d_{ji}, \forall i, j$; $d_{ii} = 0, \forall i$; e de que tomar $x_i \in \mathbb{R}_+^3$ não acarretam perda de generalidade.

Como mencionado anteriormente, a combinação entre as variáveis binárias e as restrições não-lineares impossibilitam a resolução de instâncias de grande porte. Duas ideias possíveis para evitar esse problema seriam a linearização das restrições quadráticas ou a relaxação das variáveis binárias. Ambas as ideias acrescentam novas variáveis e restrições ao modelo. Neste trabalho optamos pela relaxação das variáveis binárias do modelo (2)-(6). A escolha pela relaxação das variáveis binárias foi motivada pela natureza do modelo (2)-(6). A restrição quadrática a ser adicionada é

similar as restrições do conjunto (3) e as variáveis a serem adicionadas ao modelo são variáveis contínuas. Assim, a relaxação não requer mudança nos métodos de solução.

3.1. Relaxação do modelo

A relaxação do modelo torna as variáveis binárias y_{aij} contínuas no intervalo $[0, 1]$ e insere um novo conjunto de variáveis contínuas $w_{aij} \in [0, 1]$ e uma nova única restrição quadrática, Equação (7). O conjunto de variáveis w_{aij} terá a mesma dimensão do conjunto a ser relaxado y_{aij} portanto $n \times n \times m$ variáveis contínuas serão adicionadas ao modelo.

$$\sum_{a=1}^m \sum_{i=1}^n \sum_{j=1}^n (y_{aij} - w_{aij})^2 = mn^2 \quad (7)$$

Equação (7) é composta pelo quadrado da diferença entre os valores das variáveis $y_{aij} \in [0, 1]$ e $w_{aij} \in [0, 1]$. O valor máximo do quadrado da diferença $(y_{aij} - w_{aij})^2$ é um; esse valor é atingido em duas situações, ou quando $y_{aij} = 1$ e $w_{aij} = 0$, ou vice-versa. Assim, sempre que o máximo é atingido temos uma solução binária $\{0, 1\}$ mesmo que as variáveis tenham domínio contínuo.

O modelo contendo a relaxação das variáveis binárias é apresentado nas equações (8)-(12). Neste modelo, a restrição (7) foi adicionada a função objetivo e tratada como uma penalização. A ideia de tomar a restrição como penalização permite explorar o fato de que $\sum_{a=1}^m \sum_{i=1}^n \sum_{j=1}^n (y_{aij} - w_{aij})^2$ é uma função convexa, e que a sua adição a função objetivo não altera a convexidade da mesma. Essa abordagem foi utilizada durante os testes computacionais a fim de ter mais controle sobre a factibilidade binária do problema por meio do ajuste do parâmetro M .

$$\min \quad \sum_{a=1}^m \sum_{i=1}^n \sum_{j=1}^n (p_{aij} + n_{aij}) + M(mn^2 - \sum_{a=1}^m \sum_{i=1}^n \sum_{j=1}^n (y_{aij} - w_{aij})^2) \quad (8)$$

$$\text{s.a} \quad k_a d_a \sum_{i=1}^n \sum_{j=1}^n y_{aij} = \sum_{i=1}^n \sum_{j=1}^n (p_{aij} - n_{aij} + \|x_i - x_j\|), \quad \forall a = 1, \dots, m+1 \quad (9)$$

$$\sum_{a=1}^m y_{aij} \leq 1, \quad \forall i, j = 1, \dots, n \quad (10)$$

$$\sum_{i=1}^n \sum_{j=1}^n y_{aij} = k_a, \quad \forall a = 1, \dots, m \quad (11)$$

$$x_i \geq 0, n_{ij} \geq 0, p_{ij} \geq 0, y_{aij} \in [0, 1], w_{aij} \in [0, 1] \quad (12)$$

4. Estudo Computacional

A fim de ilustrar o comportamento do modelo (8)-(12), estudamos a resolução de cinco instâncias com 4, 6, 8, 20 e 60 vértices no espaço $\mathbb{R}^3$. A instância de 60 vértices retirada de Gujarathi et al. [2014] é composta por um conjunto de 1770 distâncias no intervalo $[1, 8]$, como ilustra a Figura 1.

A Figura 1 foi retirada de Gujarathi et al. [2014] e apresenta o conjunto de distâncias (lado esquerdo) e a configuração 3D da instância (lado direito). O conjunto de distâncias é dado por um histograma das distâncias entre os átomos da molécula de C_{60} .

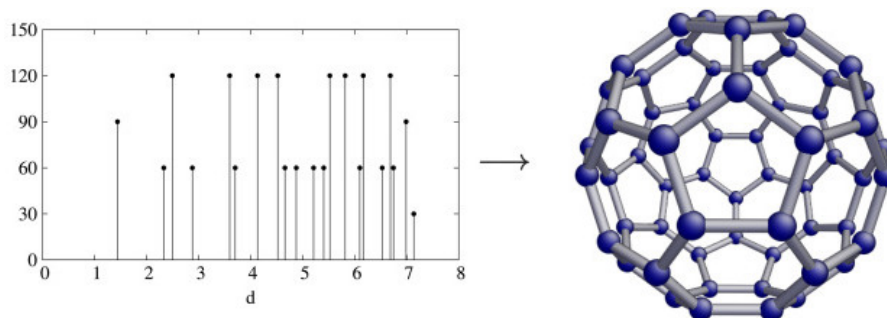


Figura 1: Figura adaptada de Gujarathi et al. [2014].

As instâncias de 4, 6, 8 e 20 vértices foram geradas de forma aleatória a fim de simular a posição de moléculas em um espaço de três dimensões. Para cada instância, as coordenadas de cada um dos vértices foram geradas de forma aleatória e depois computada a distância euclidiana entre cada par de vértices. Os vértices foram gerados no intervalo $[0, 10] \times [0, 10] \times [0, 10]$. Ao final, o conjunto de distâncias se encontra no intervalo $[2, 12]$. As instâncias estão disponíveis em ¹.

A principal diferença entre a instância apresentada na Figura 1 e as instâncias geradas é a multiplicidade de cada uma das distâncias. Enquanto que a instância de 60 vértices possui 21 distâncias com multiplicidade entre 30 e 120, para o conjunto de instâncias geradas as distâncias tem multiplicidade um. Os testes computacionais foram realizados usando o *solver* K-nitro [Waltz e Plantenga, 2011] por meio da linguagem AMPL [Fourer et al., 1993] em um *desktop* com processador i7, 32GB de memória em um sistema operacional Ubuntu 18.04. Para todos os testes o ponto inicial foi gerado pelo próprio *solver*. Todas as soluções apresentadas pelo *solver* são soluções ótimas locais devido a não-convexidade do modelo.

Os testes computacionais estão divididos em dois conjuntos. O primeiro conjunto usa a instância de 60 vértices para avaliar a qualidade da relaxação quadrática em relação ao parâmetro M , ao tempo de solução (em segundos) e ao valor das diferenças positivas (n_{aij}) e negativas (p_{aij}). O segundo conjunto analisa o desempenho do modelo para as quatro instâncias geradas de forma aleatória. A proposta é entender o comportamento do modelo em relação ao crescimento do número de distâncias sem multiplicidade; o valor dos desvios positivos e negativos são utilizados para medir a qualidade da solução.

A relação entre a factibilidade binária e o valor da penalização M é apresentada na coluna **Fact.**, onde o valor apresentado é calculado como $factibilidade = \max_{aij} |y_{aij} - int|$, onde int é o inteiro mais próximo de y_{aij} . Uma vez que a factibilidade binária do problema está relacionada à escolha do parâmetro M , fizemos um breve estudo da relação desse parâmetro com os desvios n_{aij} e p_{aij} . Os resultados são apresentados na Tabela 1; como esperado, quanto maior o valor de M , maiores os valores dos desvios. Relacionamos esse comportamento a dois fatores principais. O primeiro é relacionado à factibilidade binária; quando as variáveis estão se aproximando da factibilidade e o problema encontra-se em um ótimo local, desvios maiores são necessários para satisfazer o conjunto de restrições quadráticas. O segundo fator é que o crescimento do valor de M faz com que a função objetivo perca sensibilidade para os valores dos desvios.

A Tabela 2 apresenta dois conjuntos e informações: o primeiro conjunto contém uma descrição das instâncias geradas, considerando o número de vértices, o número de distâncias dis-

¹<https://github.com/petrabartmeyer/SBPO2019>

M	Fact.	Desvios (p_{aij}, n_{aij})			Tempo (s)
		Soma	Média	Desvio Padrão	
10	10^{-7}	8.1×10^{-4}	4.7×10^{-7}	1.04×10^{-3}	327
10^2	10^{-8}	0.001	6.0×10^{-7}	7.1×10^{-10}	648
10^4	10^{-5}	0.10	5.7×10^{-5}	6.6×10^{-15}	619
10^6	10^{-7}	0.23	1.3×10^{-6}	3.4×10^{-5}	511
10^8	10^{-7}	0.19	1.1×10^{-6}	1.1×10^{-5}	608

Tabela 1: Relação entre os os valores da penalização e os desvios em relação as medidas das distâncias.

Número de		Distâncias		Fact.	Desvios (p_{aij}, n_{aij})			Tempo (s)
Vért.	Dist.	Soma	Média		Soma	Média	Desvio Padrão	
4	7	34.47	5.74	10^{-5}	2.54	0.42	0.64	0.1
6	15	76.19	5.07	10^{-9}	0.07	0.01	0.0007	2.3
8	28	179.99	6.42	10^{-8}	1.50	0.05	0.06	5.4
20	190	1181.53	6.21	10^{-9}	1.88	0.01	0.01	870

Tabela 2: Relação entre os os valores da penalização e os desvios em relação as medidas das distâncias.

tintas, o valor da soma de todas as distâncias e a distância média — observe que o número total de distâncias leva em considerações as simplificações mencionadas na Sessão 3; o segundo conjunto de informações são os resultados em relação a factibilidade binária da variável y_{aij} , aos desvios n_{aij} e p_{aij} e ao tempo de resolução (em segundos). Para esse conjunto de instâncias usamos $M = 10^6$.

Para as quatro instâncias geradas de forma aleatória, podemos ver na Tabela 2 como o tempo de solução (em segundos) cresce exponencialmente em função da dimensão do problema. Para a instância de 8 vértices temos quatro vezes mais distâncias que para a instância de 4 vértices, contudo o tempo computacional é 50 vezes maior. Quando comparada a instância de 6 vértices, que possui quase metade das distâncias, o tempo computacionais é ainda mais que o dobro. Isso porque, para essas instâncias o conjunto de restrições quadráticas cresce com a dimensão do problema uma vez que não há multiplicidade no conjunto de distâncias. Contudo, a qualidade da solução medida pelos desvios e a factibilidade binária não mostram ter relação com a dimensão da instância.

5. Discussão e Conclusões

Este trabalho desenvolveu um modelo de otimização não-linear não-convexo inteiro misto para o problema de geometria de distâncias sem designação. Para tornar o modelo tratável do ponto de vista computacional, foi desenvolvida uma reformulação do problema original como um problema de programação não-linear não-convexo, utilizando uma técnica de relaxação para as variáveis binárias.

O desempenho da abordagem proposta foi avaliado em testes computacionais envolvendo cinco instâncias. Os testes procuraram avaliar tanto a relação das dimensões das instâncias com os tempos de solução, como identificar a forma como os parâmetros da relaxação das variáveis binárias influenciam a solução do problema.

O modelo se mostrou sensível ao ponto inicial, uma vez que a versão do *solver* utilizado não garante a obtenção de soluções ótimas globais quando o modelo é não-convexo. Em trabalhos futuros esse aspecto deve ser melhor explorado, com o uso de múltiplos pontos iniciais gerados por heurísticas e com a avaliação de técnicas de suavização progressiva da função.

O recurso de relação das variáveis binárias permitiu a abordagem do modelo com um *solver* comercial. Contudo, a qualidade da solução foi sensível ao valor da penalização M adotada.

Estratégias que escolham o valor de M de forma adaptativa devem ser avaliadas para melhorar esse comportamento.

A exploração realizada no trabalho sugere que o uso de modelos de programação não-linear pode ser uma estratégia atraente para abordar o *unassigned geometry distance problem* (uDGP). A aplicação das ideias apresentadas, combinadas aos avanços em *solvers* comerciais, pode tornar o modelo proposto uma alternativa para solução de instâncias de maior porte e para atingir soluções ótimas globais para o problema.

Agradecimento. Os autores agradecem o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Referências

- Billinge, S. J., Duxbury, P. M., Gonçalves, D. S., Lavor, C., e Mucherino, A. (2016). Assigned and unassigned distance geometry: applications to biological molecules and nanostructures. *4OR*, 14 (4):337–376.
- Biswas, P. e Ye, Y. (2006). A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization. In *Multiscale optimization methods and applications*, p. 69–84. Springer.
- Cao, M., Anderson, B. D., e Morse, A. S. (2006). Sensor network localization with imprecise distances. *Systems & control letters*, 55(11):887–893.
- Duxbury, P. M., Granlund, L., Gujarathi, S., Juhas, P., e Billinge, S. J. (2016). The unassigned distance geometry problem. *Discrete Applied Mathematics*, 204:117–132.
- Fourer, R., Gay, D. M., e Kernighan, B. W. (1993). *AMPL: a modeling language for mathematical programming*, volume 1. Scientific Press San Francisco.
- Galvan, G., Lapucci, M., Levato, T., e Sciandrone, M. (2019). An alternating augmented lagrangian method for constrained nonconvex optimization. *Optimization Methods and Software*, p. 1–19.
- Gujarathi, S., Farrow, C., Glosser, C., Granlund, L., e Duxbury, P. (2014). Ab-initio reconstruction of complex euclidean networks in two dimensions. *Physical Review E*, 89(5):053311.
- Hansen, P. e Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467.
- Huang, S. e Dokmanić, I. (2018). Reconstructing point sets from distance distributions. *arXiv preprint arXiv:1804.02465*.
- Islam, M. L., Shatabda, S., Rashid, M. A., Khan, M., e Rahman, M. S. (2019). Protein structure prediction from inaccurate and sparse nmr data using an enhanced genetic algorithm. *Computational biology and chemistry*.
- Juhás, P., Cherba, D., Duxbury, P., Punch, W., e Billinge, S. (2006). Ab initio determination of solid-state nanostructure. *Nature*, 440(7084):655.
- Lavor, C., Liberti, L., Donald, B., Worley, B., Bardiaux, B., Malliavin, T. E., e Nilges, M. (2019). Minimal nmr distance information for rigidity of protein graphs. *Discrete Applied Mathematics*, 256:91–104.
- Liberti, L. e Lavor, C. (2018). Open research areas in distance geometry. In *Open Problems in Optimization and Data Analysis*, p. 183–223. Springer.
- Liberti, L., Lavor, C., Maculan, N., e Mucherino, A. (2014). Euclidean distance geometry and applications. *Siam Review*, 56(1):3–69.
- Liberti, L., Lavor, C., Mucherino, A., e Maculan, N. (2011). Molecular distance geometry methods: from continuous to discrete. *International Transactions in Operational Research*, 18(1):33–51.
- Liberti, L. e Maculan, N. (2006). *Global optimization: from theory to implementation*, volume 84. Springer Science & Business Media.

- Moré, J. J. e Wu, Z. (1999). Distance geometry optimization for protein structures. *Journal of Global Optimization*, 15(3):219–234.
- Waltz, R. e Plantenga, T. (2011). Knitro documentation release 8.0. *Ziena Optimization LLC [en ligne] disponible sur <http://www.ziena.com> (November 28, 2011)*, 7:33–34.
- Ye, N., Roosta-Khorasani, F., e Cui, T. (2019). Optimization methods for inverse problems. In *2017 MATRIX Annals*, p. 121–140. Springer.