

Uso da Otimização por Colônia de Formigas (ACO) na determinação da trajetória de pulsos ultrassônicos em elementos de concreto

Vinícius M. Giglio¹, Vladimir G. Haach¹

¹Depto. de Engenharia de Estruturas, Escola de Engenharia de São Carlos, Universidade de São Paulo
Av. Trabalhador São Carlense, 400, Parque Arnold Schmidt, 13566-590, São Carlos/SP, Brasil
vimourag@usp.br; vghaach@sc.usp.br

Resumo. Os ensaios não destrutivos têm se mostrado eficientes na detecção de heterogeneidades do concreto, se firmando como uma alternativa menos invasiva e com menor variabilidade estatística. Dentre estes, o ensaio de ultrassom tem se desenvolvido bastante. É possível combiná-lo com a técnica de tomografia computadorizada para gerar imagens do mapa de velocidades do pulso ultrassônico em seções transversais de estruturas. No entanto, para calcular essas velocidades, é necessário que se assuma uma hipótese a respeito da trajetória seguida pelos pulsos. O presente artigo tem como objetivo determinar a trajetória de pulsos ultrassônicos em elementos de concreto utilizando a otimização por colônia de formigas na variação ACS (*Ant Colony System*). Para isso, um modelo do algoritmo foi implementado no software de geração de imagens tomográficas em estruturas TUSom. A fim de comparar os parâmetros tempo de processamento e qualidade das soluções fornecidas, o ACS foi comparado com o algoritmo determinístico de Dijkstra em um exemplo de seção de concreto com defeitos simulados. Os resultados demonstraram que Dijkstra apresentou melhor desempenho em problemas pequenos (até 150 nós), enquanto o ACS se mostrou muito superior conforme a malha cresce, se firmando como uma alternativa mais rápida para solucionar o problema das trajetórias.

Palavras-chave: Concreto, Ensaios não destrutivos, Tomografia ultrassônica, Otimização por colônia de formigas.

1 Introdução

O concreto é um dos materiais mais utilizados pela humanidade. Sua grande resistência à compressão, aliada a baixo custo e evolução no campo de estudo dos materiais, contribuiu para a sua popularização e permite a construção de qualquer tipo de obra. No entanto, estas apresentam vida útil finita. Além disso, estruturas de grande porte necessitam de constante monitoramento de sua integridade estrutural. Estas situações geram complicações na adoção de ensaios destrutivos convencionais. Nesse contexto, os ensaios não destrutivos têm se mostrado ferramentas com muito potencial, pois permitem ensaios rápidos (Ramírez [1]), menos invasivos e que eliminam muitas vezes a variabilidade estatística presente nos ensaios destrutivos, permitindo a análise de parâmetros ao longo do tempo (Hoła e Schabowicz [2]). Sua utilização têm diversas finalidades, sendo as principais a determinação de propriedades mecânicas e a detecção de heterogeneidades e do grau de deterioração de estruturas.

Dentro desta categoria se enquadra o ensaio de ultrassom, em que se utilizam transdutores que emitem e recebem pulsos dentro de elementos estruturais, medindo seu tempo de viagem. Este ainda pode ser aliado à tomografia computadorizada: à partir de medições em pontos diferentes da estrutura, representa-se a distribuição do campo de velocidades do pulso ultrassônico em uma seção transversal da mesma (chamada tomograma).

Quando se executam experimentos com o uso do ultrassom, se conhecem os tempos de viagem dos pulsos. No entanto, as suas trajetórias ao longo da estrutura são incógnitas, visto que o pulso caminha pelos pontos de maior homogeneidade, que geram maior velocidade. Estas trajetórias geralmente são supostas lineares, o que corresponde à um material homogêneo, aproximação não coincidente com a realidade. Quando a hipótese anterior é abandonada e trajetórias mais próximas das reais são obtidas, o são de forma pouco eficaz através de algoritmos determinísticos, como o algoritmo de busca de Dijkstra, que testa todas as possibilidades de viagem do pulso dentro de uma malha discretizada no elemento de concreto, apresentando grande custo computacional.

A otimização por colônia de formigas (*Ant Colony Optimization* - ACO) abrange métodos de otimização discreta de natureza probabilística, que têm se mostrado muito eficazes na solução de problemas em diversas áreas, surgindo como uma alternativa viável e menos custosa computacionalmente à solução do problema das trajetórias.

2 Revisão bibliográfica

2.1 Ensaio de ultrassom

A propriedade mais importante do concreto é a resistência à compressão, pois costuma ser o fator limitante no dimensionamento. Essa e outras propriedades geralmente são obtidas em ensaios destrutivos de pequenas amostras. No entanto, a sua moldagem ou retirada nem sempre é possível. Nesse caso, utilizam-se ensaios não destrutivos, que detectam propriedades e o grau de deterioração de estruturas de forma não invasiva (Mehta e Monteiro [3]).

O ensaio de ultrassom faz parte do grupo dos ensaios não destrutivos. Ele consiste na emissão e recepção de pulsos ultrassônicos (frequência de impacto acima de 20kHz) através de sensores chamados transdutores. A Figura 1 mostra o arranjo do ensaio de ultrassom na modalidade transparência, em que um transdutor emite e outro recebe o sinal. Como a distância entre os transdutores é conhecida e o equipamento do ensaio fornece o tempo de viagem do pulso ultrassônico, calcula-se a sua velocidade de propagação através da eq. (1): V representa a velocidade (m/s), ΔS representa a distância entre os transdutores (m) e Δt representa o tempo de viagem da onda (s).

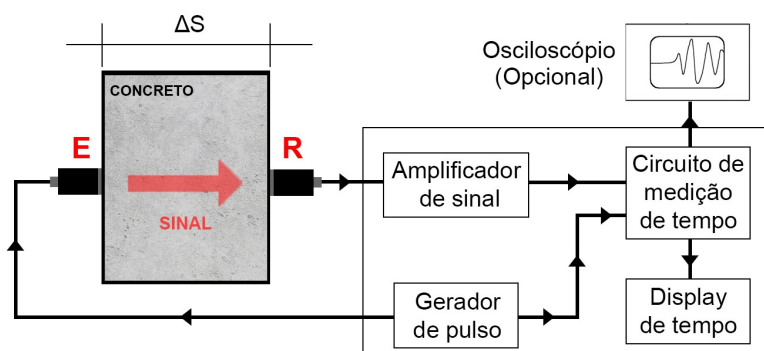


Figura 1. Arranjo do ensaio de ultrassom na modalidade transparência (Adaptado de Malhotra e Carino [4]).

$$V = \frac{\Delta S}{\Delta t}. \quad (1)$$

2.2 Tomografia ultrassônica

A representação convencional do ensaio de ultrassom representa o que ocorre em um corpo por meio de um gráfico unidimensional (Fig. 2.a). Aliando a técnica de tomografia computadorizada ao ensaio é possível reconstruir a distribuição de velocidades na seção transversal de uma estrutura (Fig. 2.b). Isso é feito à partir de medições de velocidade dos pulso ultrassônico em vários pontos da estrutura. Na figura, a área em vermelho representa uma região de baixa velocidade na seção, permitindo a identificação de um defeito.

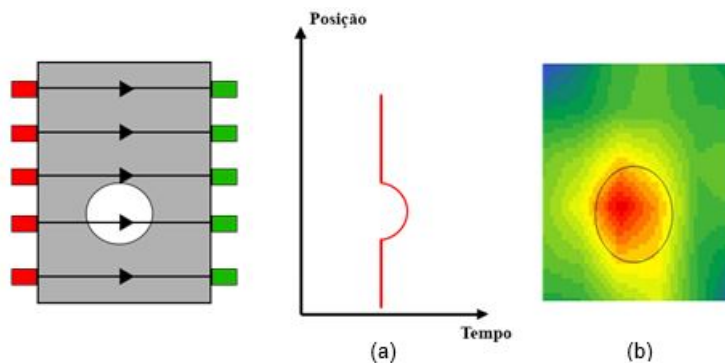


Figura 2. Representação de um defeito com o uso do ultrassom: a) Unidimensional; b) Bidimensional com tomografia (Adaptado de Ramírez [1]).

2.3 Software TUSom

O software TUSom foi desenvolvido pelo Prof. Dr. Vladimir Guilherme Haach, com a colaboração de alunos de mestrado do Departamento de Engenharia de Estruturas da EESC/USP. À partir de medidas de tempo do ensaio de ultrassom, o software mapeia as velocidades do pulso ultrassônico em seções de uma estrutura, gerando tomogramas. O programa possui seis etapas básicas:

- a) Entrada da geometria e da malha de discretização da seção transversal, que depende do transdutor utilizado (malhas 3x3 cm na utilização de transdutores de 250kHz e 5x5 cm para 54kHz);
- b) Definem-se as coordenadas dos pontos de medição, tomadas como o ponto médio das arestas da malha;
- c) Definem-se as trajetórias que as leituras realizam dentro da estrutura. Muitos autores, como Kwon et al. [5], Yanli [6], Aggelis et al. [7], Perlin [8], Ferraro et al. [9], entre outros, adotam a aproximação de trajetórias retilíneas. Essa aproximação é razoável, mas não totalmente condizente com a realidade, visto que o pulso ultrassônico viaja pelos pontos de maior homogeneidade da estrutura. Segundo Jackson e Tweeton [10], quando a diferença das velocidades apresentadas no meio não é maior que 10%, a aproximação das trajetórias dos pulsos como retilíneas não resulta em grandes distorções e é quase sempre aceitável. Para valores acima desta faixa, o resultado pode ser aceitável dependendo do usuário, mas recomenda-se a adoção de estratégias que mapeiem o trajeto do pulso ao longo da seção experimentada;
- d) Entrada de dados de medições experimentais (tempos de viagem). O software calcula parâmetros como média, desvio padrão e coeficiente de variação, permitindo o descarte de medidas pouco representativas;
- e) Cálculo das parcelas da trajetória que os pulsos irão seguir em cada elemento da malha;
- f) Processo de reconstrução iterativa de imagens, baseado na solução algébrica de um sistema de equações. Atualmente, o programa dispõe de três métodos para geração de imagens: O método dos mínimos quadrados, a técnica de reconstrução algébrica e a técnica de reconstrução iterativa simultânea.

Buscando abandonar a hipótese de trajetórias retilíneas para o pulso ultrassônico, Perlin e Pinto [11] utilizam o algoritmo de Dijkstra para calcular as trajetórias ótimas entre transdutores na malha discretizada na seção.

2.4 Algoritmo de Dijkstra

Antes de introduzir o seu funcionamento, é necessário apresentar o conceito de grafos. Um grafo é um tipo de estrutura de dados. Ele possui nós/vértices, que contêm informações, e arestas, que ligam os nós e estabelecem relações entre eles. Graficamente, os nós são representados por círculos e as arestas por linhas (Cormen et al. [12]).

O algoritmo foi concebido inicialmente por Dijkstra [13] e é aplicado da seguinte maneira: em um grafo, seleciona-se um vértice como a origem (O). Estima-se a distância entre O e ele mesmo como nula e entre O e os outros vértices do grafo como “infinita” (processo de inicialização). Em seguida, seleciona-se o vértice à menor distância da origem O. Marca-se este como visitado e calcula-se a distância entre ele e todos os vértices que se ligam a ele (adjacentes). Caso a nova distância calculada seja menor que a estimativa atual, a última assume o novo valor (processo de relaxamento da aresta). Depois da atualização das distâncias, seleciona-se o novo vértice mais próximo e o processo se repete até que todo o grafo seja visitado. Ao final, tem-se as menores distâncias entre a origem O e todos os vértices do grafo.

A trajetória seguida pelos caminhos que geram as menores distâncias pode ser obtida através do armazenamento dos pontos precedentes a todos os vértices (p[V]). Na inicialização do algoritmo, estes ainda não estão alocados (-). A Figura 3 apresenta um exemplo de aplicação do algoritmo a um grafo de 4 vértices, 5 arestas e pesos positivos. A origem é o vértice *a* e os vértices são preenchidos com a cor azul conforme são visitados. Supondo-se que o desejo é encontrar o menor caminho entre os vértices *a* e *d*, destaca-se este na cor vermelha.

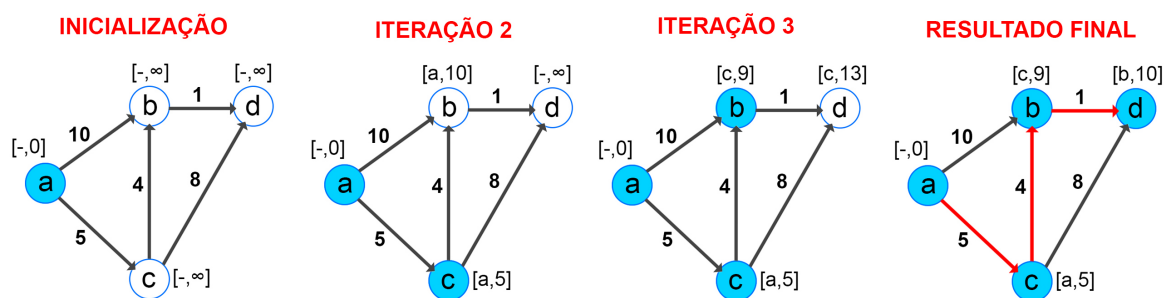


Figura 3. Exemplo de aplicação do algoritmo de Dijkstra (Elaborado pelo autor).

2.5 Otimização por colônia de formigas

O ACO é uma família de algoritmos de otimização discreta de natureza estocástica, inspirada na busca por alimento das formigas. Os algoritmos utilizam formigas artificiais que desempenham duas tarefas: construção de soluções e *feedback* positivo das melhores soluções (Santos [14]). Individualmente, as formigas possuem estrutura simples, mas graças ao princípio da estigmergia, que consiste na capacidade destas se comunicarem indiretamente através de alterações no ambiente (uso de uma substância química chamada feromônio), a solução de problemas complexos se torna possível, em um processo descentralizado de tomada de decisões (Dorigo e Stützle [15]).

O ACO foi introduzido por Dorigo [16] com o algoritmo *Ant System* (AS). Com o passar dos anos, melhorias foram propostas, dando origem ao ACS (Gambardella e Dorigo [17]), que converge mais rápido para melhores soluções. Este é aplicado da seguinte maneira: num grafo conhecido, atribuem-se à todas arestas uma quantidade de feromônio τ_0 (parâmetro escolhido pelo usuário), que juntamente com a informação heurística η (agrega conhecimento específico do problema que se quer resolver), define a atratividade de cada vértice. Em seguida, as m (parâmetro) formigas artificiais iniciam a construção de caminhos do vértice de origem (formigueiro) até o destino (alimento), utilizando uma regra pseudoaleatória de escolha (exemplificada na eq. (2) para uma formiga que vai do nó i ao j): sorteia-se um número aleatório q . Se este for menor que q_0 (parâmetro), a formiga escolhe a melhor viagem possível (maior produto feromônio-heurística entre todas arestas). Caso contrário, aplica-se uma regra probabilística para a escolha, onde a chance de cada vértice ser escolhido como destino é proporcional ao seu produto feromônio-heurística (regra da roleta). Caso o vértice j já tenha sido visitado, sua probabilidade de escolha é nula (restrição do problema). Na equação 2, β é um parâmetro que determina a importância da informação heurística e τ_{ij} representa a quantidade de feromônio na aresta que liga os nós i e j (análogo para a heurística).

$$j = \begin{cases} \max \{ \tau_{ij} \cdot \eta_{ij}^\beta \} & \text{se } q < q_0; \\ \frac{\tau_{ij} \cdot \eta_{ij}^\beta}{\sum_i \tau_{ii} \cdot \eta_{ii}^\beta} & \text{se } q > q_0. \end{cases} \quad (2)$$

As formigas se movimentam paralelamente e, após um passo de todas as formigas, a taxa de feromônio nas arestas que foram atravessadas diminui de acordo com a eq. (3), que define o valor do feromônio para a próxima iteração. Na equação, ξ representa a taxa de evaporação local (parâmetro).

$$\tau_{ij} = (1 - \xi) \cdot \tau_{ij} + \xi \cdot \tau_0. \quad (3)$$

O processo se repete até que todas as formigas atinjam o destino. Por fim, calculam-se as distâncias (função objetivo) totais percorridas por cada formiga. A melhor formiga desde o início da execução do algoritmo, ou seja, aquela que percorreu o menor caminho (L_{best}), acrescenta feromônio às arestas que atravessou (eq. (4)), aumentando sua atratividade para as formigas da próxima iteração do processo, que geralmente se repete por um número predefinido de iterações. Na equação, ρ representa a taxa de evaporação global (parâmetro).

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot (L_{best})^{-1}. \quad (4)$$

3 Metodologia

No software TUSom, desenvolvido em linguagem Pascal na plataforma Lazarus, foram implementados os algoritmos de Dijkstra e ACS, alterando a etapa de definição das trajetórias que o pulso percorre dentro da estrutura.

Em sua última etapa, o software calcula as velocidades nos nós da seção, definidos como o ponto de interseção entre as linhas da malha discretizada (um exemplo 2x2 é mostrado na Fig. 4). Estes serão chamados nós principais. Existe ainda a opção de aumentar a densidade da malha inserindo nós secundários. Nestes, a velocidade é definida à partir da interpolação linear das velocidades nos nós principais. Ao todo, são 10 níveis diferentes de densidade de nós. Na aplicação dos algoritmos, é por estes nós que o pulso irá caminhar entre os transdutores origem e destino. Dessa forma, quanto maior a densidade da malha, mais próxima da real é a trajetória do pulso.

A implementação do algoritmo de Dijkstra seguiu os moldes da seção 2.4 e ele foi usado para mapear os trajetos do pulso ultrassônico em uma seção de concreto com defeitos preestabelecidos (Fig. 5), variando a densidade de nós e comparando os resultados obtidos com o ACS em tempo de processamento e qualidade das respostas. A seção possui 150x500 mm, representando o corpo de prova usado no ensaio de ultrassom, e foi dividida em uma malha de 3x10 elementos (44 nós principais), com três defeitos pontuais e um grande. Os defeitos foram simulados adotando velocidades de 2000 m/s nos nós onde se encontram, enquanto o restante da malha possui velocidade de 4500 m/s, representando um concreto de boa qualidade. Além da malha adotada e dos defeitos preestabelecidos, a Figura 5 também exibe as trajetórias retilíneas supostas inicialmente, sendo 229 ao todo.

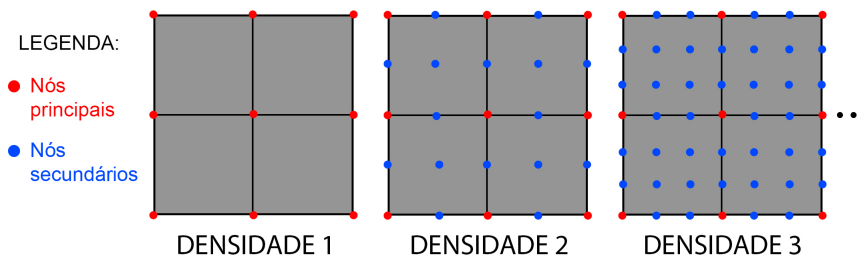


Figura 4. Exemplo de malha 2x2 elementos e níveis de densidade de nós (Elaborado pelo autor).

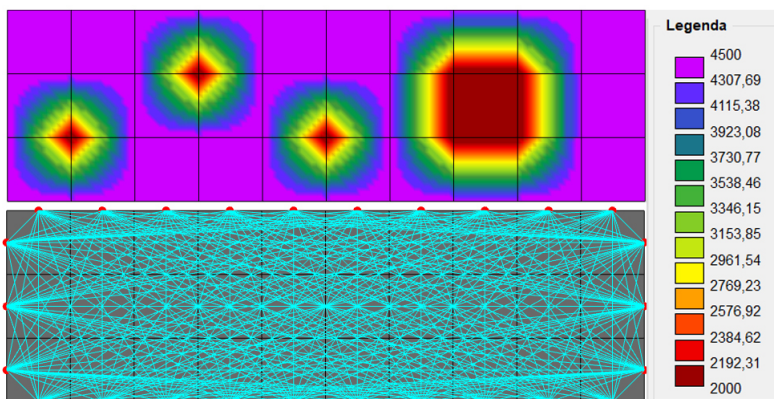


Figura 5. Exemplo utilizado no mapeamento do pulso ultrassônico (Elaborado pelo autor).

O algoritmo ACS foi implementado de acordo com a seção 2.5 e, assim como em Gambardella e Dorigo [17], utilizaram-se as chamadas *candidate lists* (CL) durante a construção dos caminhos das formigas. As CL armazenam apenas os vizinhos mais próximos de cada nó, evitando que se calcule o produto feromônio-heurística para todos os nós do grafo e melhorando o desempenho computacional. Os valores dos parâmetros foram adotados conforme recomendado por Dorigo e Stützle [15]: $m = 10$, $\beta = 2$, $\xi = \rho = 0,1$ (10%), $\tau_0 = 1/(n * est)$, em que n representa o número de nós do grafo e est uma estimativa grosseira da melhor trajetória, adotada como o caminho em linha reta entre os transdutores do ensaio. O único parâmetro alterado foi q_0 , adotado como 0,5 (50%) para favorecer a exploração e evitar a convergência a mínimos locais. Além disso, outra lógica foi adotada no cálculo da informação heurística (eq. (5)): Para uma formiga situada no nó i , a atratividade do nó j é dada pelo inverso do tempo de viagem entre os nós i e j (Δt_{ij}) somado ao tempo de viagem do nó j ao transdutor destino (Δt_{jdest}).

$$\eta_{ij} = \frac{1}{(\Delta t_{ij} + \Delta t_{jdest})}. \tag{5}$$

Para cada uma das densidades de malha adotadas no exemplo, alteraram-se o tamanho da CL e o número de iterações, buscando atingir resultados com no máximo 1% de erro médio, calculado conforme a eq. (6) para cada uma das trajetórias. O erro médio é obtido pela média do erro das 229 trajetórias do exemplo.

$$erro = \left(\frac{\Delta t_{ACS} - \Delta t_{Dijkstra}}{\Delta t_{Dijkstra}} \right). \tag{6}$$

4 Resultados e discussão

A Figura 6 exibe os resultados do tamanho da malha de nós adotada pelo tempo de processamento na aplicação dos algoritmos de Dijkstra e ACS para o exemplo estudado. Apesar dos resultados em tempo absoluto serem diferentes de acordo com as especificações de cada máquina onde o software for executado, o formato das curvas e consequentemente a diferença entre elas se mantém, visto que o código é sequencial (não paralelizado).

É possível observar através dos resultados que, até um limite de cerca de 150 nós, o algoritmo de Dijkstra apresenta menor tempo de processamento com relação ao observado no ACS. No que diz respeito à qualidade das respostas, o primeiro é sempre superior, visto que testa todas as possibilidades de viagem entre nós, encontrando sempre a trajetória ótima. Dessa forma, o uso do ACS não se justifica nesse intervalo.

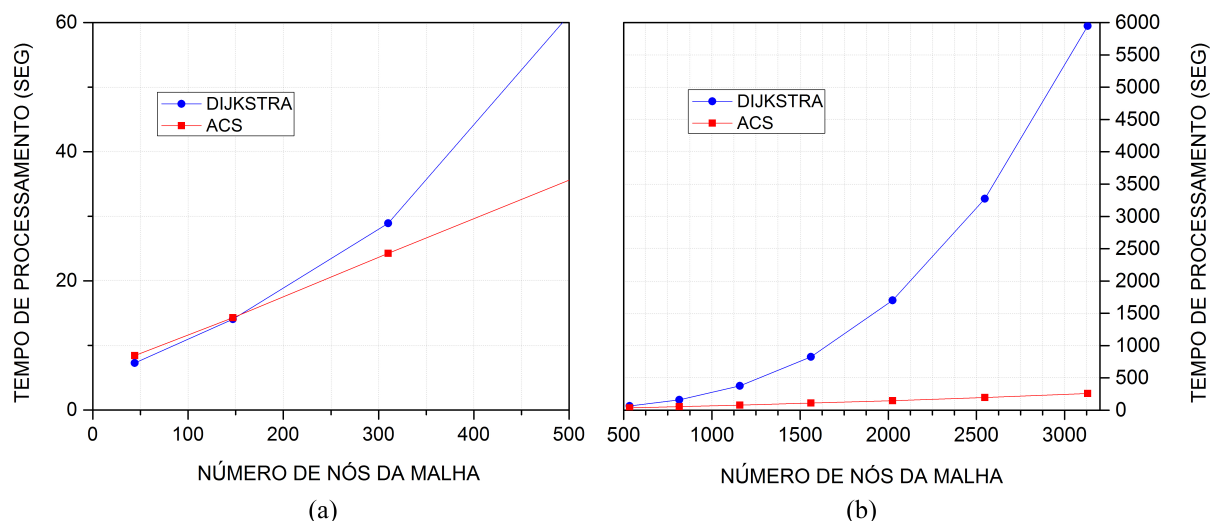


Figura 6. Número de nós x tempo de processamento dos algoritmos no exemplo estudado: a) Entre 0 e 500 nós (densidades 1 a 4); b) Entre 500 e 3200 nós (densidades 4 a 10) (Elaborado pelo autor).

Apesar disso, é possível observar que o algoritmo de Dijkstra cresce exponencialmente com o tamanho da malha, apresentando tempos de execução maiores que o ACS em todos os casos com mais de 150 nós. Isso se deve ao fato do número de possibilidades de visita na malha ser $(n + 1)!$, sendo n o número total de nós (o termo extra se deve à presença do transdutor receptor). Cada nó visitado elimina apenas uma possibilidade dentro de um espaço de busca muito grande. Em contrapartida, o ACS cresce praticamente de forma linear, pois apesar de também depender do tamanho da malha em etapas como a montagem do grafo, depende em maior proporção de parâmetros como o número de formigas, de iterações e o tamanho da CL, que foram relativamente pequenos: apenas 10 formigas, 11 vizinhos e 810 iterações foram suficientes para encontrar trajetórias com erro médio de 1,00% na maior das malhas (3131 nós), vide Tabela 1. A tabela também exibe os valores obtidos para o tempo de processamento de cada um dos algoritmos (em segundos) e a diferença percentual entre eles. É possível observar diminuições de tempo da ordem de 90% para as maiores densidades de malha na utilização do ACS, situações em que a trajetória do pulso ultrassônico mais se aproxima da real e, portanto, de maior interesse prático.

Vale destacar também que, por sua natureza probabilística, o ACS resulta em trajetórias diferentes a cada execução, diferentemente do algoritmo de Dijkstra. No entanto, como o erro exibido representa a média de 229 trajetórias (cada trajetória executa o ACS uma vez), pode-se considerar os valores como representativos.

Tabela 1. Número de iterações e tamanho da CL adotados em cada execução do ACS (Elaborado pelo autor).

Densidade da malha	Número de nós	Tamanho da CL	Iterações	Erro médio	Tempo ACS (s)	Tempo Dij. (s)	Diferença relativa
1	44	3	5	0,51%	8,42	7,30	+15,41%
2	147	4	5	0,70%	14,31	14,05	+1,81%
3	310	5	5	0,94%	24,26	28,93	-16,15%
4	533	5	15	0,99%	37,57	66,78	-43,73%
5	816	5	80	0,99%	56,67	160,59	-64,71%
6	1159	6	100	1,00%	79,93	376,95	-78,80%
7	1562	6	200	1,00%	110,19	827,25	-86,68%
8	2025	9	190	0,99%	147,66	1699,63	-91,31%
9	2548	10	320	1,00%	195,34	3273,10	-94,03%
10	3131	11	810	1,00%	258,66	5948,91	-95,65%

5 Conclusões

É possível concluir, à partir dos resultados, que para malhas menores (pouco mais de 150 nós) o algoritmo de Dijkstra se mostra uma alternativa melhor, pois é mais rápido e fornece as respostas exatas (erro nulo). À partir disso, a aplicação do ACS ao problema de determinação das trajetórias do pulso ultrassônico em estruturas foi muito promissora, atingindo erros pequenos em tempos de execução consideravelmente menores (reduções da ordem de 90% nas maiores malhas). Isso se deve principalmente à menor sensibilidade do ACS ao tamanho da malha quando comparado com o algoritmo de Dijkstra. Assim, é esperado que em malhas maiores que as do exemplo testado a diferença seja ainda maior. Além disso, malhas mais refinadas constituem situações de maior interesse prático por representarem trajetórias do pulso mais próximas das reais.

Os algoritmos determinísticos não devem ser desprezados, podendo ser utilizados em problemas com menor quantidade de nós. Outros algoritmos mais rápidos que Dijkstra, como o A* (lê-se “A-estrela”), podem ser testados, adotando por fim uma abordagem mista: algoritmos determinísticos até determinado tamanho de malha e ACS à partir deste.

Como desenvolvimentos futuros, pretende-se avaliar a influência do erro médio do algoritmo ACS na geração de imagens tomográficas, além de testar uma implementação paralela do ACS em uma linguagem de nível mais baixo (como o FORTRAN), buscando desempenhos ainda melhores. Num panorama mais amplo, pretende-se, em um momento posterior, simular numericamente o ensaio de ultrassom com fidelidade.

Declaração de autoria. Por meio deste, os autores confirmam que são os únicos responsáveis pela autoria deste trabalho e que todo material incluído no mesmo é de propriedade (e autoria) dos autores ou possui a permissão dos respectivos donos/autores para ser incluído aqui.

Referências

- [1] Ramírez, F. C., 2015. *Detecção de danos em estruturas de concreto por meio de tomografia ultrassônica*. Dissertação de mestrado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, Brasil.
- [2] Hoła, J. & Schabowicz, K., 2010. State-of-the-art non-destructive methods for diagnostic testing of building structures – anticipated development trends. *Archives of Civil and Mechanical Engineering*, vol. 10, pp. 5–18.
- [3] Mehta, P. K. & Monteiro, P. J. M., 2014. *Concreto: microestrutura, propriedades e materiais*. IBRACON.
- [4] Malhotra, V. M. & Carino, N. J., 2004. *Nondestructive testing on concrete*. CRC Press.
- [5] Kwon, J., Choi, S. J., & Song, S. M., 2005. Implementation and evaluation of the ultrasonic tof tomography for the ndt of concrete structures. In *Computational Imaging III*, pp. 47–54, San Jose, USA. International Society for Optics and Photonics.
- [6] Yanli, C., 2010. The shortest path ray tracing algorithm in concrete computerized tomography. In *International Forum on Information Technology and Applications*, pp. 392–396, Kunming, China.
- [7] Aggelis, D. G., Tsimpris, N., K., C. H., T., S., & Kobayashi, Y., 2011. Numerical simulation of elastic waves for visualization of defects. *Construction and Building Materials*, vol. 25, pp. 1503–1512.
- [8] Perlin, L. P., 2011. *Tomografia ultrassônica em concreto: desenvolvimento de ferramenta computacional*. Dissertação de mestrado, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, Brasil.
- [9] Ferraro, C. C., Boyd, A. J., & Consolazio, G. R., 2013. Evaluation of damage to bridge piers using pulse velocity tomography. *Construction and Building Materials*, vol. 38, pp. 1303–1309.
- [10] Jackson, M. J. & Tweeton, D. R., 1994. MIGRATOM – geophysical tomography using wavefront migration and fuzzy constraints. Technical report, U.S. Department of Interior – Bureau of Mines, Washington, USA.
- [11] Perlin, L. P. & Pinto, R. C. A., 2019. Use of the network theory to improve the ultrasonic tomography in concrete. *Ultrasonics*, vol. 96, pp. 185–195.
- [12] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C., 2009. *Introduction to algorithms*. The MIT Press.
- [13] Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, vol. 1, pp. 269–271.
- [14] Santos, V. C., 2017. *Uma abordagem híbrida para planejamento exploratório de trajetórias e controle de navegação de robôs móveis autônomos*. Tese de doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil.
- [15] Dorigo, M. & Stützle, T., 2004. *Ant colony optimization*. The MIT Press.
- [16] Dorigo, M., 1992. *Optimization, learning and natural algorithms*. PhD thesis, Electronic Department, Politecnico di Milano, Milan, Italy.
- [17] Gambardella, L. M. & Dorigo, M., 1996. Solving symmetric and asymmetric TSPs by ant colonies. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 622–627, Nagoya, Japan.