

Suavização de Malhas não Estruturadas usando-se o Algoritmo Genético

Luciano Kiyoshi Araki, Elton Fernando Doehnert

Abstract

A suavização de malhas não-estruturadas é um tópico já muito estudado pois apresenta um impacto muito grande na acurácia da solução numérica em problemas envolvendo o método de elementos finitos e o método de volumes finitos. Esse artigo apresenta um método para otimização de malhas não estruturadas triangulares usando-se um algoritmo genético de ponto flutuante para encontrar a melhor posição para os vértices da malha. Esta posição é baseada na função *fitness* calculada para cada vértice interno à malha, tal função é baseada na média da qualidade dos volumes que compartilham o vértice. O algoritmo é executado para uma malha gerada com triangulação de Delaunay e os resultados mostram que houve uma melhoria na malha com relação ao parâmetro de qualidade estabelecido, no entanto, o método se mostra computacionalmente mais caro que outros métodos de suavização.

Keywords: Algoritmo Genético, Evolução, Malha não estruturada, Suavização

1. Introdução

Embora ferramentas de geração de malhas automáticas sejam amplamente usadas, essas ferramentas podem não garantir a qualidade das malhas. Não apenas no processo de tesselação, mas também no refinamento da malha, é possível
5 que alguns elementos severamente distorcidos ou fora de forma sejam criados. Mesmo quando uma malha uniforme é desejada, a ferramenta de tesselação pode gerar elementos que são muito pequenos ou muito grandes comparados com os elementos desejados. [1]

Existem vários tipos de esquemas de suavização de malhas, tais como suavização
10 Laplaciana e suavização baseada em otimização. Tipicamente cada método
possui um compromisso entre qualidade e custo computacional. Por exem-
plo, a suavização Laplaciana requer um custo computacional muito baixo, mas
frequentemente resulta em uma malha de baixa qualidade nos elementos ou
mesmo com elementos inválidos. Por outro lado, enquanto suavizações basea-
15 dos na otimização são mais prováveis em evitar elementos inválidos e obtém
uma maior qualidade na malha, o custo computacional é muito maior que a
suavização Laplaciana. [1]

Algoritmos Genéticos (GAs) são métodos adaptativos que podem ser usados
para resolver problemas envolvendo procura e otimização. Eles são baseados
20 nos processos genéticos de organismos biológicos. Ao longo de muitas gerações,
populações naturais evoluem de acordo com os princípios da seleção natural e
"sobrevivência do mais apto", primeiramente pronunciados por Charles Darwin
no livro *A Origem das Espécies*. Imitando este processo, o algoritmo genético
é capaz de "evoluir" soluções para problemas do mundo real, desde que ten-
25 ham sido corretamente codificados. Por exemplo, GAs podem ser usados para
desenhar estruturas de pontes, para a maior proporção de força/peso, ou para
determinar a menor quantidade de sobras no corte de tecidos. Ele também
pode ser usado para o controle de processos online, tais como em uma fábrica
química, ou balanceando sistemas de computadores de multi processadores. [2]

30 O GA usa uma analogia direta com a natureza, onde indivíduos de uma
população, representados no algoritmo por possíveis soluções de um problema,
recebem um nível de adaptabilidade de acordo com a sua performance na solução
do problema, tal nível é denominado *fitness*. Indivíduos com um alto valor
de *fitness* possuem uma maior probabilidade para se reproduzirem, ou seja,
35 passarem suas características para a nova população. Já membros da população
bom baixo *fitness* irão ter menor chance de reprodução, de modo que ao longo
do tempo terão suas características genéticas extintas.

No algoritmo genético, uma solução em potencial para o problema pode ser
representado por um conjunto de parâmetros. Esses parâmetros (conhecidos

40 como *genes*) são unidos juntos para formar uma cadeia de valores (frequentemente chamados de *cromossomos*). ([3] foi o primeiro a mostrar e muitos ainda acreditam que o ideal é o uso de um alfabeto binário para essa cadeia de valores.) Por exemplo, se nosso problema for o de maximizar uma função de três variáveis, $F(x, y, z)$, nós podemos representar cada variável por um número binário de 45 10-bits. Nossos cromossomos, portanto, conteriam três genes, e possuiriam 30 números binários. No entanto, segundo [4] pode-se representar essas variáveis usando-se uma codificação de ponto flutuante, e com essa representação o algoritmo se torna mais rápido, mais consistente e com maior precisão especialmente em domínios grandes, em que a codificação binária se tornaria demasiadamente 50 longa.

Em termos genéticos, o conjunto de parâmetros representados por um dado cromossomo é chamado de *genótipo*. O genótipo contém informações necessárias para construir um organismo - que é chamado de *fenótipo*.

2. Métodos

55 O método utilizado consiste na geração de uma malha usando-se triangulação de Delaunay conformada, método que gera triangulações de alta qualidade. Este método é descrito em [5] e [6]. Esta malha é, então, otimizada utilizando-se o algoritmo genético em cada um de seus vértices internos, se modo a se estabelecer a melhor posição com relação a um critério de qualidade, essa qualidade 60 é definida em [7] e é calculada para cada volume da malha conforme a equação:

$$SQ_i = \frac{4\sqrt{3}A_i}{\sum_{i=1}^3 l_i^2} \quad (1)$$

No problema da determinação da posição ideal dos vértices da malha, pode-se definir um conjunto de coordenadas cartesianas para a determinação dessa posição, tal conjunto pode ser denominado pelo *genótipo* da solução enquanto que o *fenótipo* é a média da qualidade dos volumes que compartilham esse 65 vértice. A adaptação de cada indivíduo depende da performance do seu fenótipo.

Isso pode ser inferido do seu *genótipo* - i.e. pode ser computado do seu cromossomo usando a função *fitness*.

2.0.1. Algoritmo

Cada geração será definida por uma nova população, que irá ter, em média,
70 um melhor *fitness* médio e irá, portanto, convergir para uma solução do problema. Um resumo da metodologia usando-se algoritmo genéticos é dada por [8] e é dada por:

- **Inicialização:** Inicialmente muitas soluções representando indivíduos são geradas aleatoriamente de modo a se formar a população inicial. O tamanho
75 dessa população depende da natureza do problema, mas tipicamente contém muitas centenas de milhares de possíveis soluções. Tradicionalmente, a população é gerada aleatoriamente, cobrindo-se todo o possível espaço de soluções. Ocasionalmente, pode-se priorizar determinada área desse espaço entendido como tendo uma maior probabilidade de ter a solução
80 ideal.
- **Seleção:** A cada nova geração, um conjunto da população atual é selecionada de alguma forma de modo a gerarem a nova geração. Tais indivíduos são selecionados pela sua função *fitness*. Alguns métodos calculam o *fitness* de toda a geração enquanto outros selecionam apenas alguns
85 indivíduos aleatórios, no entanto este último método pode ser muito mais custoso com relação ao tempo. Normalmente a função *fitness* escolhida é estocástica e feita de tal forma que mesmo soluções com baixo *fitness* tem uma pequena chance de serem selecionadas, tal estratégia ajuda a manter a diversidade genética da população alta, prevenindo o surgimento prematuro de soluções ruins. As formas de seleção mais estudadas e usadas
90 incluem seleção *roulette wheel* e *tournament selection*.
- **Reprodução:** O próximo passo é a geração dos novos indivíduos que irão fazer parte da nova população, que irá substituir a população antiga. Tal

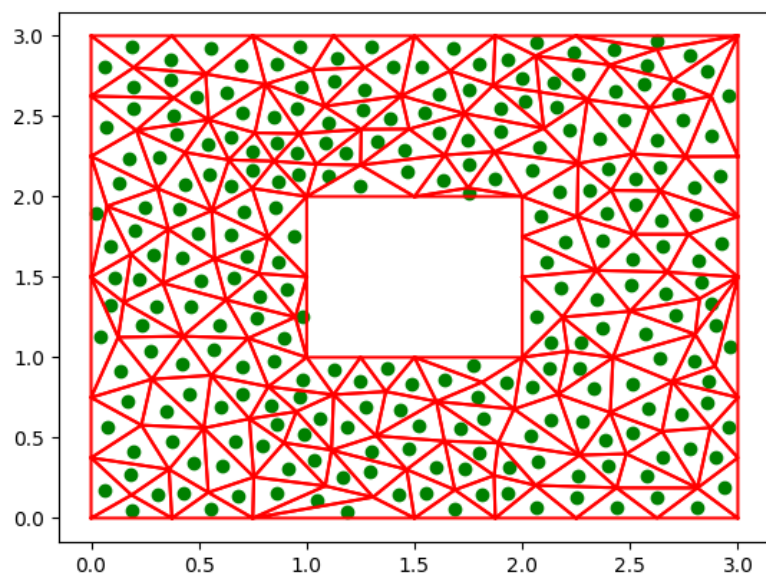
95 reprodução é feita usando-se o *genótipo* dos pais selecionados e os recom-
binando através do processo de *crossover* e *mutation*. Como a "criança"
gerada nesse processo possui genótipos dos dois pais, ela compartilhará
muitas características com seus "pais". Novos pais são selecionados para
cada nova criança e esse processo continua até toda a nova população
100 ser gerada. Embora tipicamente escolham-se apenas dois pais para cada
criança, alguns pesquisadores [9] sugerem que mais de dois pais podem
produzir uma criança com um melhor genótipo.

O processo de evolução de novas soluções(indivíduos) continua até que algum
critério de parada seja estabelecido, que pode ser:

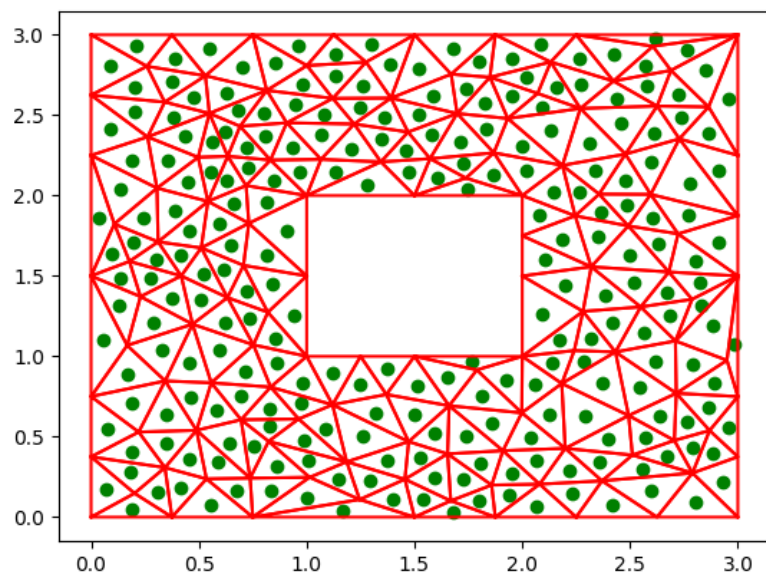
- Uma solução é encontrada satisfazendo um critério mínimo;
- 105 • Um número de gerações fixado é alcançado;
- Um tempo computacional é alcançado;
- A solução com o maior fitness da população atinge um platô tal que novas
gerações não conseguem melhorar o resultado;
- Inspeção Manual;
- 110 • Uma combinação das anteriores.

3. Resultados

O algoritmo de otimização baseado no métodos apresentado é implementado
para duas geometrias complexas e os resultados estão apresentados nas figuras
1 e 2, também é informado para cada malha os valores do maior ângulo, menor
115 ângulo, desvio padrão entre os ângulos, maior qualidade entre os volumes, menor
qualidade, desvio padrão da qualidade e a média da qualidade, que pode ser
usado como um parâmetro da qualidade geral de uma malha.

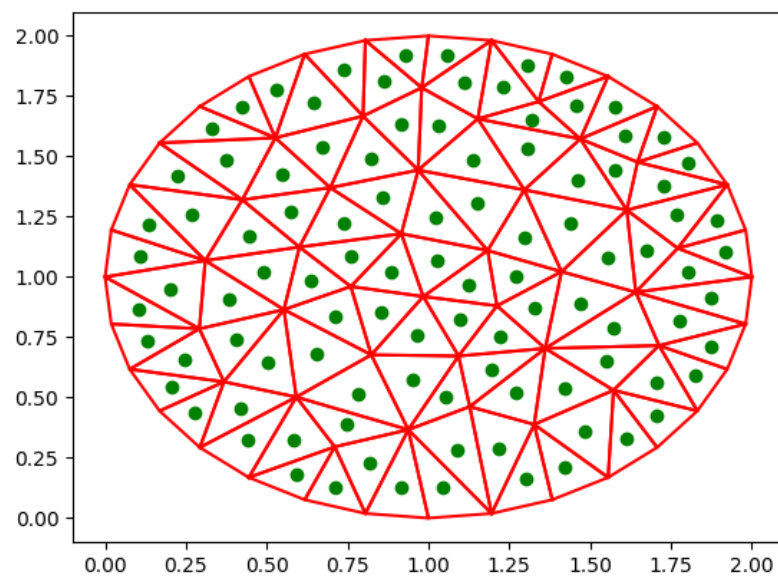


(a) Malha Original

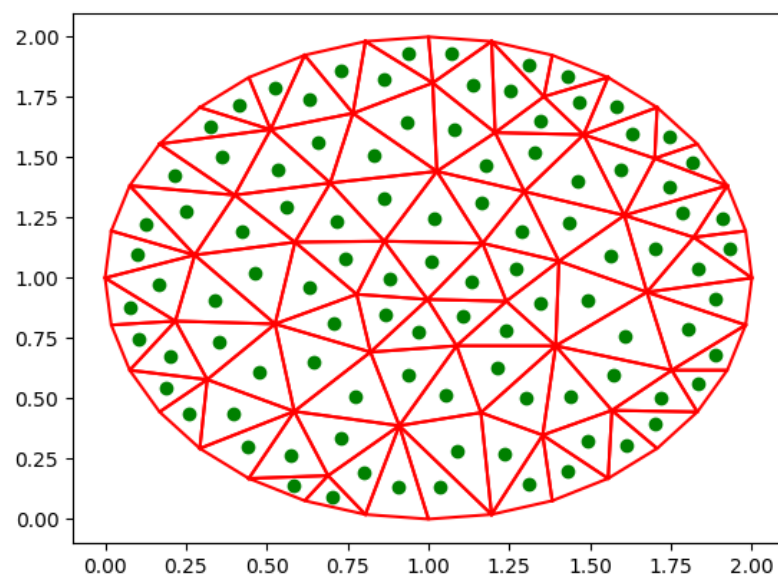


(b) Malha após GA

Figure 1:⁶ Malha 1



(a) Malha Original



(b) Malha após GA

Figure 2.7 Malha 2

Table 1: Malha 1 original

Valor	Ângulo	Qualidade
Média	60	0.830759
Desvio Padrão	22.868034	0.193268
Mínimo	9.554770	0.076371
Máximo	157.010795	0.999923

Table 2: Malha 1 com GA

Valor	Ângulo	Qualidade
Média	60	0.848804
Desvio Padrão	21.527223	0.189446
Mínimo	5.504031	0.041763
Máximo	161.853212	0.999185

Table 3: Malha 2 original

Valor	Ângulo	Qualidade
Média	60	0.873503
Desvio Padrão	17.606637	0.102456
Mínimo	26.888982	0.604936
Máximo	109.030637	0.999390

Table 4: Malha 2 com GA

Valor	Ângulo	Qualidade
Média	60	0.900385
Desvio Padrão	15.643499	0.098648
Mínimo	24.665694	0.621293
Máximo	106.827269	0.998796

Conclusão

A otimização de malhas não estruturadas é de grande importância em certos
120 tipos de aplicações, esta otimização pode ser feita com o uso de algum tipo de
heurística.

Neste trabalho duas malhas não estruturadas triangulares foram geradas
usando-se a triangulação de Delaunay, foi desenvolvido um método baseado
no algoritmo genético que otimizasse a posição dos vértices internos da malha
125 baseado em um fator de qualidade. Os resultados mostram que mesmo para uma
malha gerada pela triangulação de Delaunay, que é um método muito eficiente
de geração de malhas triangulares, o algoritmo foi capaz de melhorar a média da
qualidade dos volumes dessa malha, conforme pode ser verificados nas tabelas
2 e 4 em que a qualidade é medida através de um fator de qualidade definido
130 em 1.

Apesar do algoritmo conseguir melhorar a qualidade das malhas, sabe-se que
outros métodos de suavização são mais eficientes em relação ao custo computa-
cional assim como geram um resultado melhor.

- [1] T. Zhou, K. Shimada, An angle-based approach to two-dimensional mesh
135 smoothing, Proceedings of the 9th International Meshing Roundtable (412)
(2000) 373–384.
- [2] D. Beasley, An Overview of Genetic Algorithms : Part 1 , Fundamentals.
- [3] J. H. Holland, Adaptation in natural and artificial systems (1992).
- [4] C. Z. Janikow, Z. Michalewicz, An Experimental Comparison of Binary and
140 Floating Point Representations in Genetic Algorithms, undefined.
- [5] J. R. Shewchuk, Reprint of: Delaunay refinement algorithms for triangular
mesh generation, Computational Geometry: Theory and Applications 47 (7)
(2014) 741–778. doi:10.1016/j.comgeo.2014.02.005.
- [6] M. C. Lin, D. N. Manocha, P. ACM Workshop on Applied Computational
145 Geometry (1st : 1996 : Philadelphia, Applied computational geometry :

towards geometric engineering : FCRC '96 Workshop, WACG '96, Philadelphia, PA, May 27-28, 1996 : selected papers, Springer, 1996.

- [7] M. Falsafioon, S. Arabi, R. Camarero, F. Guibault, Comparison of Two Mesh Smoothing Techniques for Unstructured Grids, IOP Conference Series: Earth and Environmental Science 22. doi:10.1088/1755-1315/22/2/022020.
- [8] M. Kumar, M. Husian, N. Upreti, D. Gupta, GENETIC ALGORITHM: REVIEW AND APPLICATION, International Journal of Information Techonology and Knowledge Management.
- [9] A. E. Eiben, P. E. Raué, Z. Ruttkay, Genetic algorithms with multi-parent recombination, 2012, pp. 78–87. doi:10.1007/3-540-58484-6_252.