Trabalho 1 - Fecho Convexo

Aluno Kleyton da Costa (2312730)
Professor Waldemar Celes (DI/PUC-Rio)

1 Motivação

Este trabalho tem como objetivo descrever a aplicação de um algoritmo para a determinação do fecho convexo 2D para uma nuvem de pontos. Os experimentos levaram em consideração o algoritmo Jarvis March (ou *Gift Wrapping*).

2 Metodologia

Um fecho convexo pode ser definido como o conjunto de todas as combinações convexas para um determinado conjunto de pontos $S = p_1, \ldots, p_n$. Definindo uma combinação convexa de S como $\lambda_1 p_1 + \lambda_2 p_2, \ldots, \lambda_n p_n$. Dessa maneira, o conjunto de todas as combinações convexas (fecho convexo) é dado por

$$conv(S) = \{\lambda_1 p_1 + \lambda_2 p_2, \ldots, \lambda_n p_n\} | \lambda_i \geq 0, \sum \lambda i = 1\} \tag{1}$$

Existem diversos algoritmos que podem ser implementados com a finalidade de computar o fecho convexo de S. No entanto, neste trabalho foi considerado o algoritmo Jarvis March, também conhecido como *Gift Wrapping*. Este algoritmo foi proposto por Jarvis (1973) e tem como intuição os seguintes passos: considerando uma distribuição aleatória de pontos, o fecho convexo é determinado pelo primeiro ponto mais baixo e mais à direita e usando a origem para calcular um ângulo para cada ponto. O ângulo pode ser aproximado através de um produto vetorial ou um produto escalar. Considerando o próximo ponto do fecho convexo como aquele com o menor ângulo interior, desenhamos um segmento de reta entre os dois pontos. Com isso, usamos os dois pontos conhecidos para calcular novamente o ângulo entre todos os outros pontos na nuvem de pontos. Repetimos a seleção ponto com o menor ângulo interior e movemos a simulação para a frente. Esse processo é repetido até que se retorne ao ponto original. E, dessa maneira, o conjunto de todos os pontos que foram escolhidos será considerado o fecho convexo.
Complexidade esperada: $O(nh)$

3 Experimentos

3.1 Nuvens de pontos fixa

A partir das nuvens de pontos 1 e 2, o algoritmo foi implementado na linguagem Python e tem os seus resultados apresentados na Figura 1.

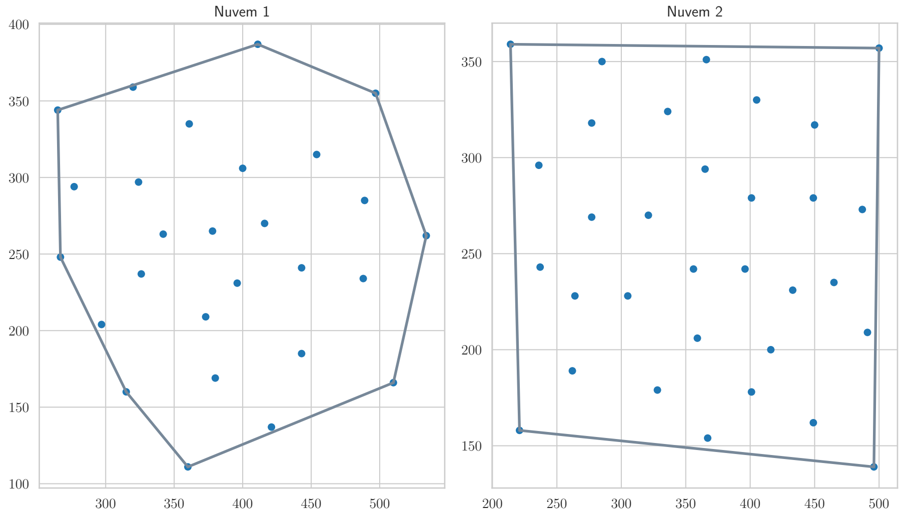

Figura 1: Nuvem1.txt e Nuvem2.txt

3.2 Simulação de pontos aleatórios

Além das nuvens de pontos definidas anteriormente como pontos de entrada, foi considerado um experimento no há geração aleatória de pontos. Cada subfigura em 2 possui 10.000 pontos aleatórios.

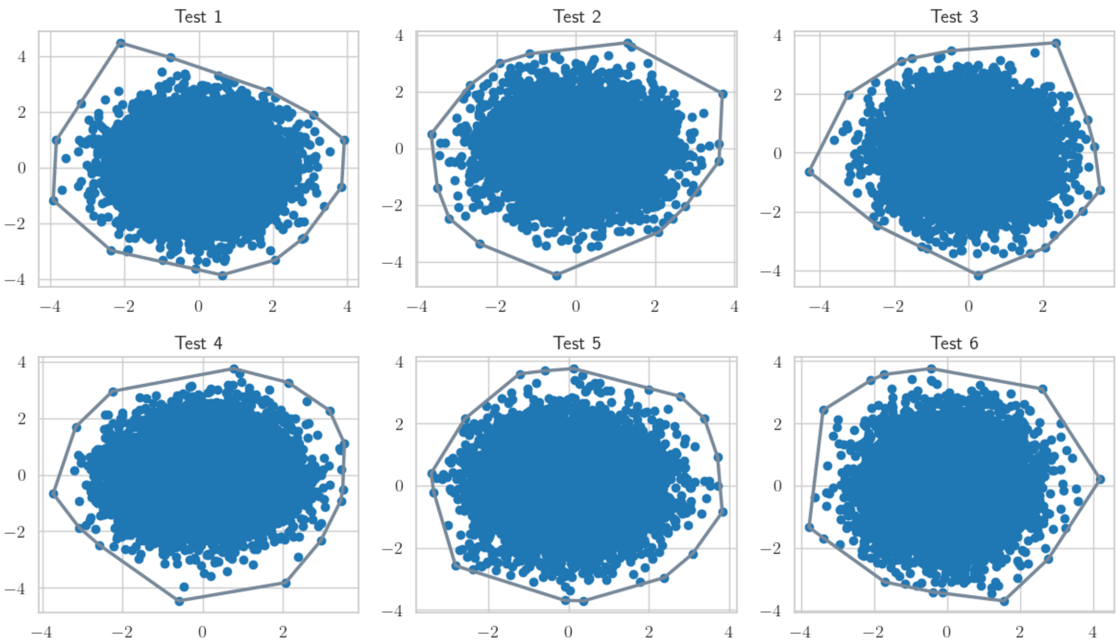

Figura 2: Nuvem de pontos aleatórios

Além disso podemos observar na Figura 3 o tempo de execução para diferentes tamanhos de entrada. Como esperado, temos que o tempo de execução é proporcional ao produto do tamanho da entrada vezes o número de pontos no fecho convexo.

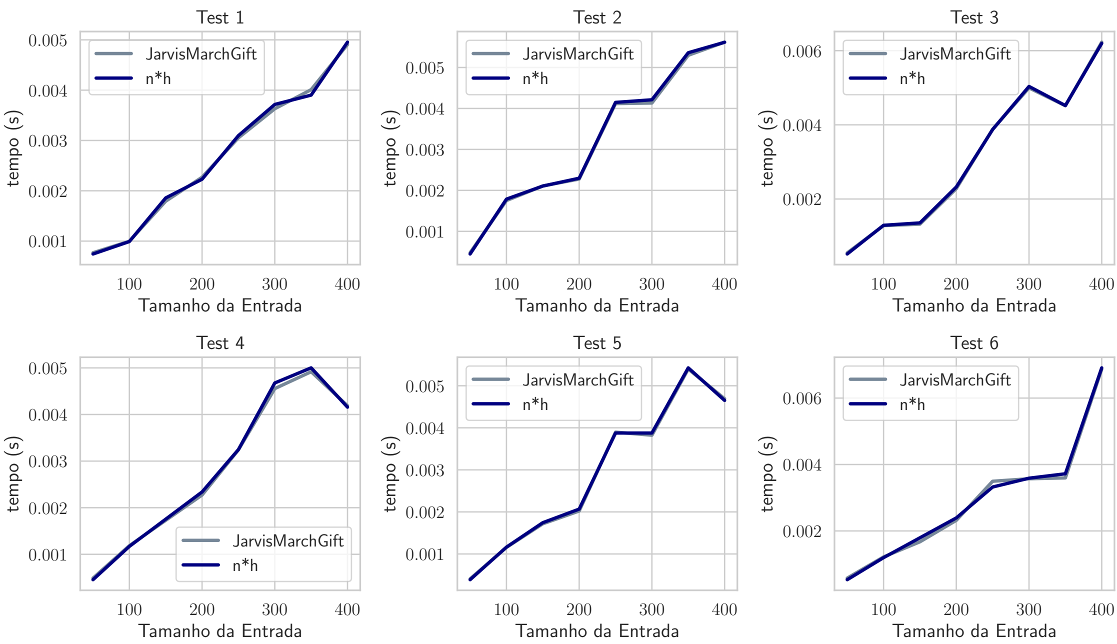

Figura 3: Nuvem de pontos aleatórios

4 Conclusão

Neste trabalho, foi apresentado o algoritmo Jarvis March, também conhecido como Gift Wrapping, para computar o fecho convexo de um conjunto de pontos.

Além disso, foi realizado um experimento com a geração aleatória de pontos para avaliar o desempenho do algoritmo para diferentes tamanhos de entrada. Como esperado, o tempo de execução foi proporcional ao produto do tamanho da entrada vezes o número de pontos no fecho convexo.

Existem diversos algoritmos para computar o fecho convexo de um conjunto de pontos, e o algoritmo Gift Wrapping é um dos mais simples e eficientes. No entanto, a escolha do algoritmo mais adequado depende do contexto específico em que ele será aplicado.

Referências

R.A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2(1):18–21, 1973. ISSN 0020-0190. doi: https://doi.org/10.1016/0020-0190(73)90020-3. URL `https://www.sciencedirect.com/science/article/pii/0020019073900203`.