4. Dois algoritmos ingênuos

Nesta seção veremos dois algoritmos para encontrar o fecho convexo de um conjunto S de pontos no plano. O primeiro identifica quais dos pontos em S são pontos extremos de S (vértices de conv(S)); ou melhor, determina quais os pontos de S são pontos não-extremos. O segundo algoritmo encontra as arestas de conv(S).

4.1. Pontos não-extremos.

O algoritmo que veremos é sugerido pela Definição 10 acima.

Lema 1. *Um ponto v de um conjunto S é ponto não-extremo de S se e somente se v pertence a algum triângulo cujos vértices são pontos de S distintos de v.* ∎

Baseado no lema acima podemos desenvolver o seguinte algoritmo.

Algoritmo `Pontos-Não-Extremos`
`Entrada:` Um conjunto finito $S = \{p_1, \dots, p_n\}$ de pontos no plano.
`Saída:` Os pontos extremos de S (vértices de conv(S)).
for cada i **do**
for cada $j, j \neq i$ **do**
for cada $k, k \neq i, k \neq j$ **do**
  for cada $l, l \neq i, l \neq j, l \neq k$ **do**
    if p_l está à esquerda ou sobre (p_i, p_j) e
      p_l está à esquerda ou sobre (p_j, p_k) e
      p_l está à esquerda ou sobre (p_k, pi)
      then p_l é não-extremo.

A complexidade do algoritmo acima é $O(n^4)$: existem quatro laços cada um com $O(n)$ iterações: para cada um dos $O(n^3)$ triângulos verificamos quais dos $n-3$ pontos restantes pertencem ao triângulo.

4.2. Arestas extremas.

É mais fácil identificarmos arestas do fecho convexo do que vértices extremos. Uma arestas xy é uma aresta *extrema* de um conjunto de pontos S se x e y forem vértices extremos e todos os pontos de S estão sobre ou de um mesmo lado da aresta.

Algoritmo `Arestas-Extremas`
`Entrada:` Um conjunto finito $S = \{p_1, \dots, p_n\}$ de pontos no plano.
`Saída:` As arestas extremas de S, i.e. as arestas de conv(S).
for cada i **do**
for cada $j, j \neq i$ **do**
  for cada $k, k \neq i, k \neq j$ **do**
    if p_k não está à esquerda ou sobre (p_i, p_j) e
      then (p_i, p_j) não é aresta de conv(S).

O algoritmo acima tem complexidade de tempo $O(n^3)$ pois existem três laços encaixados e cada um desses laços fará $O(n)$ iterações: para cada um dos $O(n^2)$ segmentos testamos se o segmento e aresta de conv(S).