10. Cota inferior

Vimos dois algoritmos que encontram o fecho convexo de um conjunto de pontos em tempo $O(n \log n)$. Uma pergunta natural agora seria: Será que se pode fazer melhor do que isto? Poderia ser o caso (como ocorreu com o Problema de Triangularização de Polígonos) que exista um algoritmo, digamos, de complexidade de tempo $O(n)$ para o problema, mas este algoritmo está ainda aguardando para ser descoberto. Nesta seção nos mostraremos (ou argumentaremos) que um tal algoritmo não existe e que $O(n \log n)$ é o melhor que alguém pode esperar conseguir, i.e. $\Omega(n \log n)$ é cota inferior para o Problema do Fecho Convexo.

Observação. O material desta seção foi extraído do Capítulo 2 de [3], Capítulo 1 e 3 de [14], Capítulo 1 e 4 de [4], e Capítulo 3 de [12].

10.1. **O que pretendemos provar.** Recordemos que $O(f(n))$ é o conjunto de funções

$$O(f(n)) = \{g(n) \quad | \quad \text{existem contantes positivas } c \text{ e } n_0 \text{ tal que } 0 \leq g(n) \leq cf(n) \text{ para } n \geq n_0\},$$

ou seja, $g(n)$ é limitado por $cf(n)$ para valores de n suficientemente grandes.

O conceito de límite inferior ou cota inferior (*lower bound*) é semelhante,

$$\Omega(f(n)) = \{g(n) \quad | \quad \text{existem contantes positivas } c \text{ e } n_0 \text{ tal que } 0 \leq cf(n) \leq g(n) \text{ para } n \geq n_0.\}$$

O que queremos mostrar nesta seção é que a função complexidade de tempo de qualquer algoritmo que resolva o Problema do Fecho Convexo é $\Omega(n \log n)$, ou seja, qualquer algoritmo que contrói o fecho convexo de um dado conjunto de n pontos faz $\Omega(n \log n)$.[2]

10.2. **Redução entre problemas.** Em geral é um problema difícil encontrar cotas inferiores para problemas. A difivuldade é que contas inferiores devem valer para os algoritmos que já foram escritos e para qualquer outro que possa ou não algum dia ser descoberto. Ou seja, a cota inferior é para *todos* os algoritmos que resolvem o problema, e portanto é uma complexidade intrínsica do problema. No entanto, foram encontradas cotas inferiores para alguns problemas importantes, como por exemplo ordenação. Isto faz com que o trabalho de muitas pessoas fique mais fácil, pois agora cotas inferiores podem ser obtidas via "redução entre problemas".

Suponha que seja sabido que um Problema A tem uma cota inferior $\Omega(n^4)$. Agora suponha que o Problema A pode ser *reduzido* para o Problema B no sentido em que um algoritmo que resolve o Problema B pode ser usado para resolver o Problema A (com um 'pequeno' esforço adicional). Logo, se existe um algoritmo que resolva B rapidamente então este algoritmo pode ser usado para resolver o Problema A 'rapidamente'. Isto frequentemente implica uma cota inferior para o problema B. De fato, suponha que o esforço adicional

[2] Aqui algum comentário extra é devido. Como estamos falando em operações elementares então o *modelo de computação* deve neste momento estar bem definido. O que nos faremos nesta seção é 'passar por cima' de alguns detalhes técnicos que são muito importantes.

de resolver o Problema A fazendo uma única chamada a um algoritmo que resolva o Problema B é $O(n)$ (ou seja o algoritmo para resolver A usa um algoritmo que resolve B como subrotina). Podemos portanto concluir que, como $\Omega(n^4)$ é cota inferior de A, então $\Omega(n^3)$ é cota inferior para o Problema B. Esta será a maneira que usaremos para estabelecer uma cota inferior para o Problema do Fecho Convexo.

10.3. **A redução.** Sabe-se que ordenação tem cota inferior $\Omega(n \log n)$.[3] Reduziremos o Problema de Ordenação (este é o Problema A) ao Problema do Fecho Convexo (este é o Problema B). Esta redução é devida a Shamos [15] e é bem simples. Suponha que nos é dada um lista não-ordenada de números $(x_1, x_2, \ldots, x_n)$, $x_i \geq 0$ $(i = 1, \ldots, n)$; este são os dados do problema A. Suponha algora que nos temos um algoritmo `Fecho` que contrói o fecho convexo (o algoritmo dá como resposta um polígono convexo tendo como vértices pontos do conjunto e contendo todos os demais pontos) de n pontos em tempo $T(n)$ (função complexidade de tempo). Usaremos o Algoritmo `Fecho` para ordenarmos a lista de pontos dados em tempo $T(n) + O(n)$, onde o termo $O(n)$ corresponde ao o tempo adicional de converter uma solução do Problema de Ordenação para o Problema do Fecho e vice-versa.

Forme um conjunto de pontos no plano constituída pelos pontos (x_i, x_i^2) $(i = 1, \ldots, n)$ (veja Figura 15). Execute o Algoritmo `Fecho` tendo estes pontos como entrada para encontrar o fecho convexo destes pontos. Encontre o ponto mais baixo $p := (x_k, x_k^2)$ do fecho convexo contruído, este passo pode se feito em $O(n)$. A ordem em que os vértices do fecho convexo ocorrem ao percorrermos a sua fronteira no sentido anti-horário a partir de p nos fornece uma lista dos números dados em ordem crescente. Logo, o Algoritmo `Fecho` pode ser usado para ordenar n números em tempo $O(T(n))$. Logo, $T(n) = \Omega(n \log n)$.

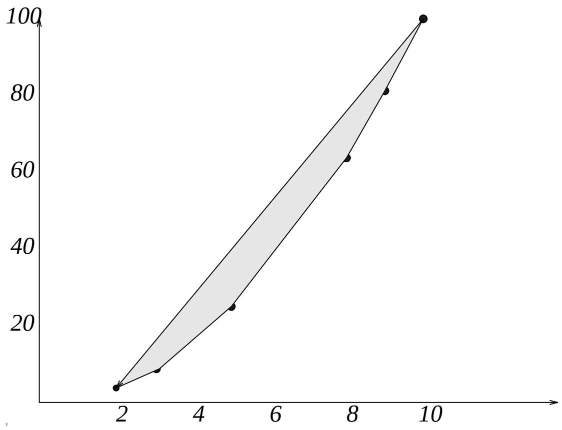

FIGURA 15. Parábola obtida na ordenação dos números $2, 3, 5, 8$, e 10.

Observe que na redução acima foi fundamental o fato do Algoritmo `Fecho` devolver um polígono convexo como resposta. Não é claro se conseguiríamos

[3] Está afirmação é imprecisa, e nos custaria muito tempo fazê-la precisa.

fazer a mesma redução se os pontos extremos do fecho convexo fossem dados como resposta. Foi um problema aberto durante muitos anos descobrir se o problema de achar os pontos extremos de um conjunto era de fato mais fácil do que encontrar o fecho convexo desses pontos representado por um polígono. Pesquisadores estabeleceram que $\Omega(n \log n)$ é também uma cota inferior para este outro problema (veja Preparata & Shamos [14], páginas 101–103).