Gabriel Aflalo Robassini – RA 091237

O Problema

Dados n tipos de moedas, tal que wi ∈ ℤ+ é o valor do i-ésimo tipo de moeda e pi ∈ ℤ+ o seu peso, e seja W um número inteiro positivo que representa o valor do troco, o objetivo é resolver o seguinte modelo:

$$\text{Maximize} \quad \sum_j p_j x_j$$

$$\text{Sujeito} \quad \sum_j w_j x_j = W$$

O algoritmo implementado funciona com complexidade O(nW), como justificado por trecho abaixo.

Reduzindo o problema do troco para o problema do Caminho Mínimo

Seja um grafo com n(W+1) vértices denotados por j^k onde *j*=1,2,…,n e *k*=0,1,…,W em que *j* representa o indice da moeda e *k* o valor total somado ao chegar naquele vértice. Assim,o vértice j^k conecta-se com outro vértice $(j+1)^{k'}$ com aresta de peso $p_{(j+1)} * ((k'-k)\%w_{(j+1)})$ desde que $(k'-k)\%w_{(j+1)}$ seja zero e $k' \geq k$.
Um vértice de origem *s* também é adicionado ao grafo e ***s*** conecta se ao vértice 0^k com peso 0 se *k*%p_j for diferente de zero, caso contrário com peso (*k* / 0^k) * p_0.
Assim o menor caminho de s a n^W indica a combinação de moedas que somam W no valor e tem o menor peso total possível.

Complexidade

São criados n(W+1) vértices e cada vertice *j* conecta-se somente com os vértices adjacentes se as condições acima forem satisfeitas. Para a criação e conexão dos vértices toma-se O(nW). Com o grafo pronto, roda-se o algoritmo de dijkstra que toma O(V + E), já que trata-se de uma DAG. Por fim, percorre-se o caminho retornado no passo anterior, fazendo a impressão do número de moedas utilizadas O(V). No total, temos que o algoritmo roda em um tempo O(nW) o que não significa ser polinomial uma vez que W não representa o tamanho da entrada do problema e para maior precisão o algoritmo roda em O(n2^w) em que w = logW, ou seja, bits necessários para representar W. Assim, como a solução não roda em tempo polinomial, nada podemos concluir sobre o problema do troco.