

Algoritmo de Caminho mais Curto baseado em Multiplicação de Matrizes

Considere um grafo G com n nós, com pesos de arestas dados por uma matriz W , $n \times n$. O algoritmo de multiplicação de matrizes constrói uma matriz D , $n \times n$, onde cada elemento (i, j) é o custo do caminho mais curto entre os nós i e j , da seguinte forma.

Uma matriz inicial D^0 , $n \times n$, contendo pesos de caminhos de tamanho no máximo 1, é construída a partir da matriz de pesos do grafo:

$$d_{i,j}^1 = \begin{cases} 0 & \text{if } i = j \\ w(i,j) & \text{se existe aresta (i,j)} \\ \infty & \text{caso contrário} \end{cases}$$

Cada iteração $r + 1$ contrói uma nova matriz de distâncias, usando, para construir cada elemento, a linha i ($d_{i,\cdot}^r$), calculada na iteração anterior, e a coluna j ($d_{\cdot,j}^0$) da matriz D^0 inicial:

$$d_{i,j}^{r+1} = \min_{1 \leq k \leq n} (d_{i,k}^r + d_{k,j}^1)$$

Assim, em cada passo $r + 1$, substituímos o menor peso de caminho entre i e j que passa por no máximo r nós, pelo menor peso de caminho que passa por no máximo $r + 1$ nós.

Usando essa idéia, precisaremos de $n - 1$ iterações para construir a matriz de caminhos mais curtos para um grafo de n nós. Como só estamos interessados nesse resultado final, podemos calcular D^{n-1} com $\log(n - 1)$ passos combinando D^r consigo mesma em cada iteração:

$$d_{i,j}^{2r} = \min_{1 \leq k \leq n} (d_{i,k}^r + d_{k,j}^r)$$

Com isso, o tempo de execução do algoritmo sequencial passa a ser $\Theta(n^3 \log n)$.