



PROGRAMAÇÃO DINÂMICA

Um Estudo de Caso de Programação Dinâmica com o Problema do Caixeiro Viajante com Restrição de Custo

Prof. Dr. Paulo André S. Giacomini
Departamento de Engenharias e Computação



Universidade Estadual de Santa Cruz





O PROBLEMA DO CAIXEIRO VIAJANTE COM RESTRIÇÃO DE CUSTO

Dado um conjunto de nós $\{1, 2, \dots, n\}$, uma matriz de custo não negativos $C = [c_{ij}]$, valores positivos $\{v_1, v_2, \dots, v_n\}$, e um orçamento B , encontre um sub caminho de valor máximo, iniciando e terminando no nó 1 (número), tal que o custo total não exceda B . Para detalhes, veja Sokkappa (1990).

Esse problema pode ser resolvido via programação dinâmica da seguinte forma: dado $S \subseteq \{2, 3, \dots, n\}$ e $l \in S$, deixe $C(S, l)$ ser o custo mínimo de visitar todos os nós em S partindo de 1 (número) e terminando em l (letra). Então, tem-se:

Processamento intermediário

$$C[\{l\}, l] = c_{1l}, \forall l.$$

$$C[S, l] = \min_{m \notin S} (C[S - l, m] + c_{ml})$$

Custo mínimo

$$\min_{l \in \{2, 3, \dots, n\}} (C[\{2, 3, \dots, n\}, l] + c_{1l})$$



O PROBLEMA DO CAIXEIRO VIAJANTE COM RESTRIÇÃO DE CUSTO

Explicação: o custo de visitar apenas um nó l (letra) é o custo de sair do nó de origem 1 (número) e chegar a l . O custo de um sub trajeto S_i terminando em l é o mínimo entre o custo do sub caminho S_i sem l passando por m , mais o custo de ir de m até l , onde m é um vértice ainda não visitado em S .

O custo mínimo é o custo do sub trajeto sem o nó inicial 1, mais o custo de ir de 1 até l , para todos os caminhos terminando em l .

Processamento intermediário

$$C[\{l\}, l] = c_{1l}, \forall l.$$

$$C[S, l] = \min_{m \notin S} (C[S - l, m] + c_{ml})$$

Custo mínimo

$$\min_{l \in \{2, 3, \dots, n\}} (C[\{2, 3, \dots, n\}, l] + c_{1l})$$



O PROBLEMA DO CAIXEIRO VIAJANTE COM RESTRIÇÃO DE CUSTO

Explicação: o valor de cada caminho S terminando em l (letra) é a soma do valor de cada vértice em S , incluindo l (letra), mais o valor do nó 1 (número).

O valor ótimo, ou seja, o resultado, é o máximo dentre todos os possíveis valores de $V[S, l]$, tal que o respectivo custo $C(S, l) + c_{1l}$ não exceda o orçamento, ou seja, para o valor ser máximo, o custo não precisa ser mínimo, mas deve obedecer ao orçamento dado.

Valor para cada caminho

$$V[S, l] = \sum_{i \in S} v_i + v_1$$

Valor máximo

$$V_{opt} = \max_{(S, l) | C(S, l) + c_{1l} \leq B} V(S, l)$$



TRABALHO

Programar um programa que resolva o problema do caixeiro viajante com restrição de custo, utilizando programação dinâmica, para valores válidos quaisquer de n , de valores positivos v_i , e de orçamento B . Qualquer linguagem de programação pode ser utilizada.



BIBLIOGRAFIA

CORMEN, T. H.; Desmistificando algoritmos. Elsevier. 2014.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C.; Introduction to Algorithms. MIT Press. Third Edition. 2009.

DROZDEK, Adam. Estruturas de Dados e Algoritmos em C++. Thomson Pioneira, 2001

FEOFILOFF, Paulo; Minicurso de análise de algoritmos. Universidade de São Paulo. 2013. .

KNUTH, Donald E.; The Art of Computer Programming – Sorting and Searching. 2o Ed. Boston: Addison-Wesley. 1998.

LEISERSON, Charles E.; STEIN, Clifford; RIVEST, Ronald L.; CORMEN, Thomas H. Algoritmos - Trad. 2ª Ed. Americana, Editora Campus, 2002.

PREISS, Bruno. Estruturas de Dados e Algoritmos. Editora Campus, 2001.

SKIENA, Steven S. The Algorithm Design Manual. Springer-Verlag, 1997. Online: <http://www2.toki.or.id/book/AlgDesignManual/>

SKIENA, Steven S.; REVILLA, Miguel A. Programming Challenges. Springer, 2003.

SKIENA, Steven S.; REVILLA, Miguel A. Programming Challenges. Springer. 2003 Online: http://acm.cs.buap.mx/downloads/Programming_Challenges.pdf

SKIENA, Steven S. The Algorithm Design Manual. Springer, 2008. Online:
<https://www.inf.ufpr.br/andre/textos-CI1355-CI355/TheAlgorithmDesignManual.pdf>

SOKKAPPA, Padmini Rachel. The cost-constrained traveling salesman problem. University of California, 1990.