IA 718 Tópicos em Sistemas Inteligentes

2-Introdução à Programação Dinâmica

Conteúdo

1. Introdução
2. Problema do caminho mínimo
3. Solução com programação dinâmica
4. Análise de complexidade
5. Programação dinâmica forward
6. Exemplos

1-Introdução

- Programação dinâmica
  - metodologia de otimização
  - problemas que requerem decisões sequenciais interelacionadas
  - decisão tem um custo imediato e afeta contexto decisões futuras

- Objetivo
  - como obter a sequência de decisões
  - minimização custo total em um número de estágios
  - compromisso entre custo imediato e futuro

Processos de decisão multiestágios

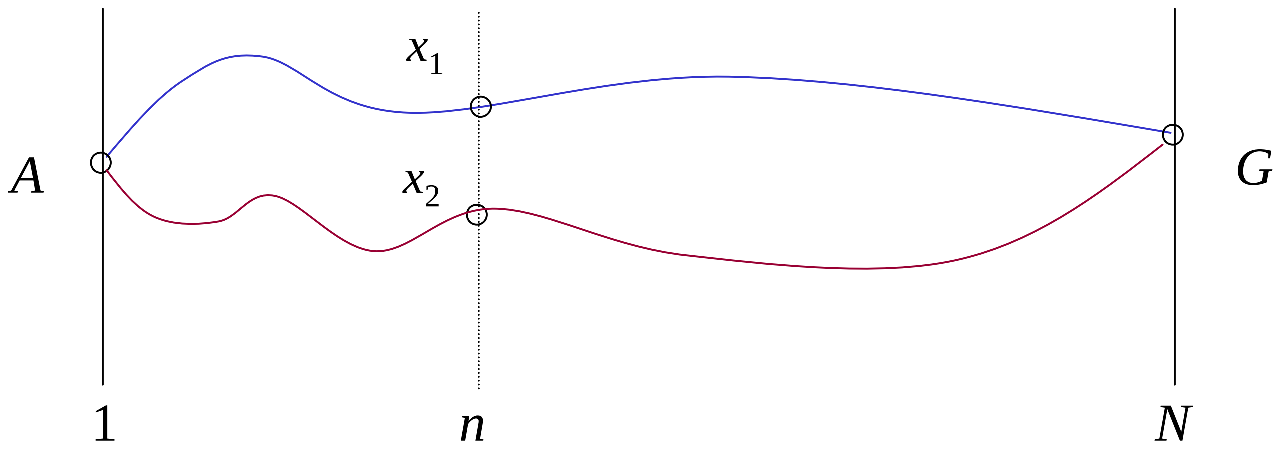
- Decisão multiestágios
  - processo que pode ser desdobrado em um número de etapas seqüenciais, ou estágios
- Estado
  - condição do processo num dado estágio é o estado neste estágio
- Decisões
  - opções que se tem em cada estágio
  - cada decisão causa uma transição do estado

- Estratégia (política)
  - uma seqüência de decisões
  - uma decisão para cada estado do processo
- Retorno
  - custo, benefício, associado a cada estágio de decisão
  - pode variar com o estágio e o estado
- Questão
  - determinar a política ótima (aquela que resulta no melhor retorno)

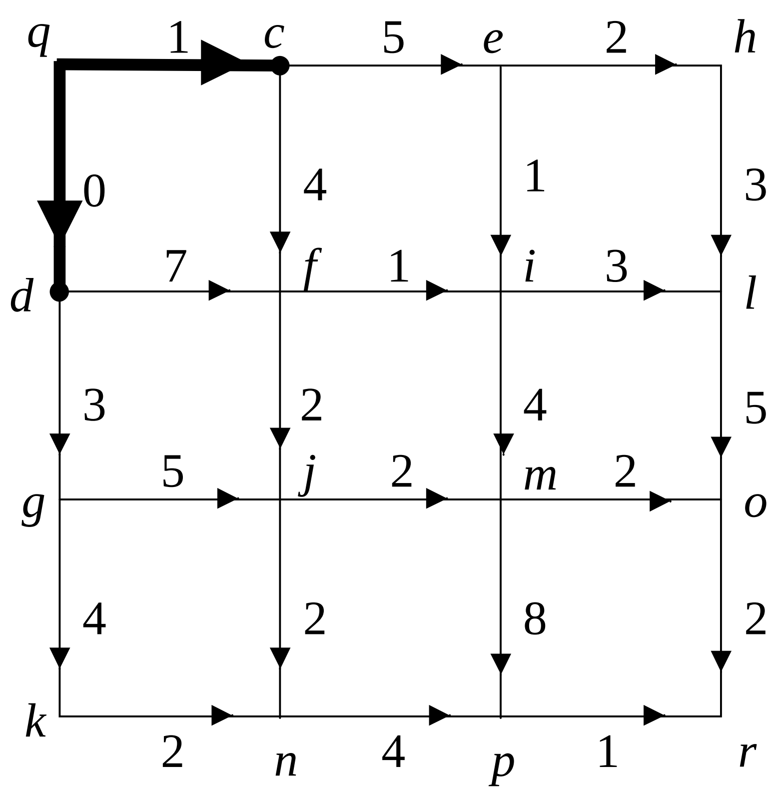
Princípio de otimalidade de Bellman

Uma estratégia ótima apresenta a propriedade segundo a qual, a despeito das decisões tomadas para se atingir um estado particular num certo estágio, as decisões restantes a partir deste estado devem constituir uma estratégia ótima.

[Richard Bellman, 1957]



2-Problema do caminho mínimo

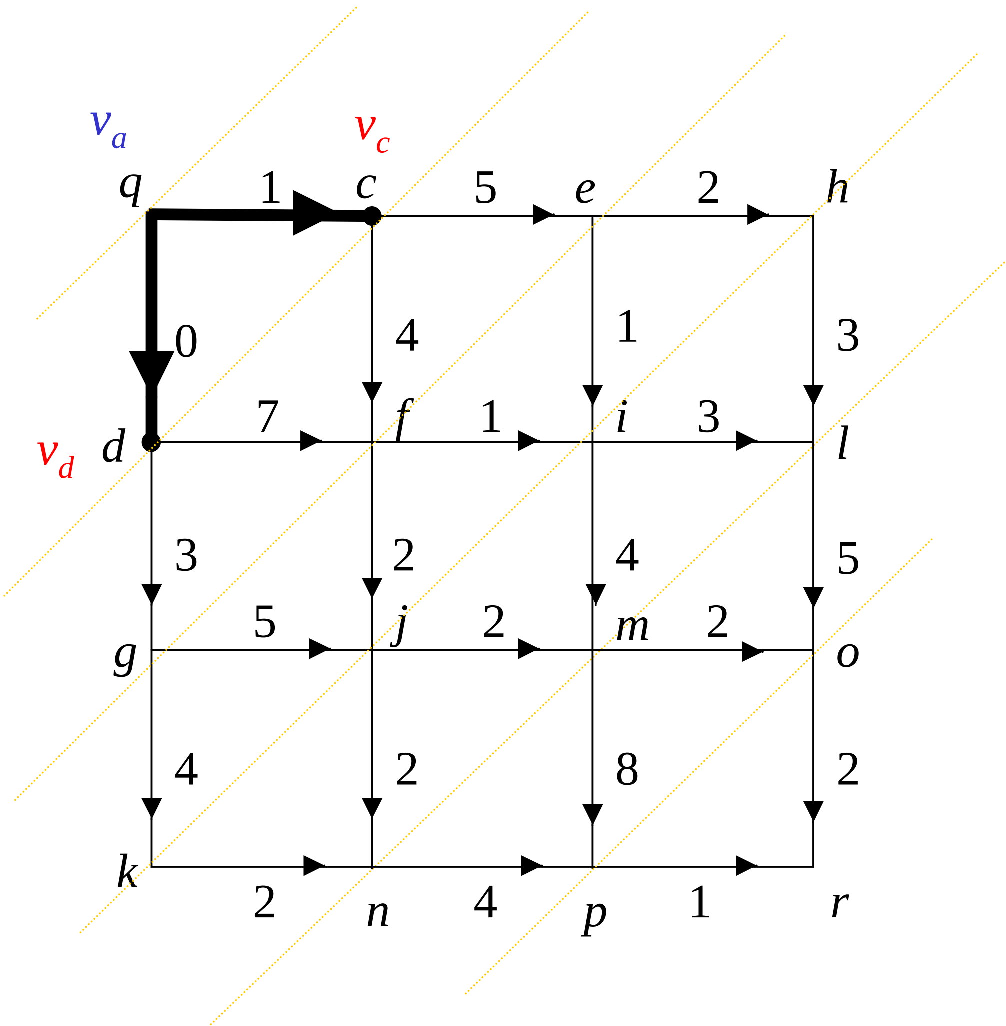


Qual é o caminho de menor esfôrço (tempo, custo, dis - tância, etc.) entre *q* e *r* ?

- 20 caminhos distinos
- 5 adições por caminho
- 19 comparações

3-Solução com programação dinâmica

v_i melhor caminho de i até r



$v_q = \min\{1 + v_c\ ,\ 0 + v_d\}$

$v_c = \min\{5 + v_e\ ,\ 4 + v_f\}$
$v_d = \min\{7 + v_f\ ,\ 3 + v_g\}$
……………………….

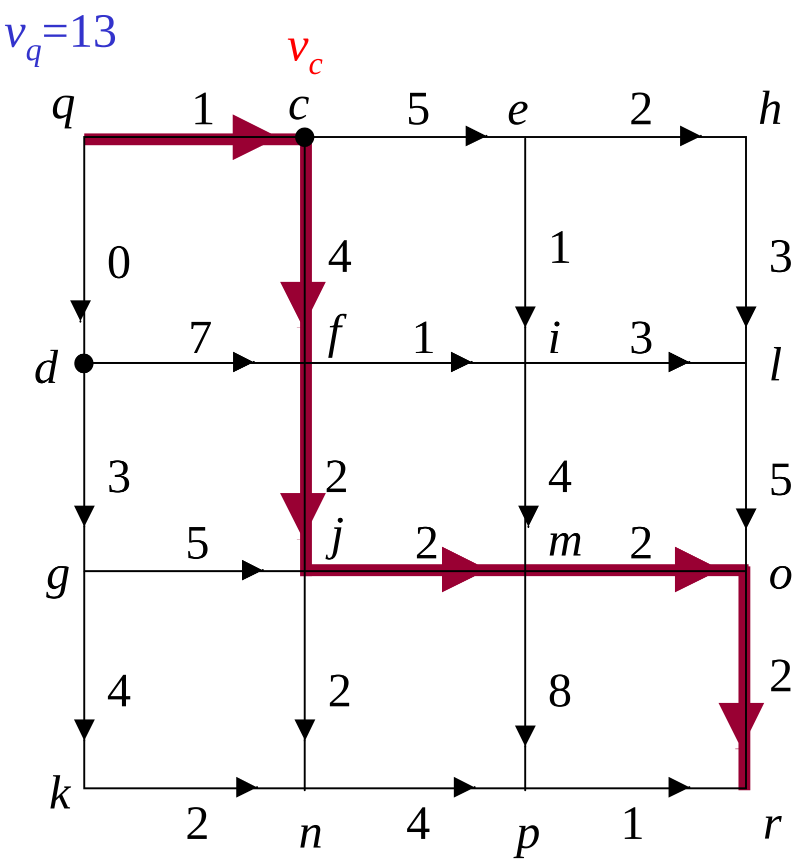
$v_l = 5 + v_o$
$v_m = \min\{2 + v_o\ ,\ 8 + v_p\}$
$v_n = 4 + v_p$

$v_o = 2$
$v_p = 1$

$v_r = 0$

S estado

P_S sucessor de S no caminho ótimo de S até r



Estratégia ótima

$P_o = r$ $P_p = r$

$P_l = o$ $P_m = o$ $P_n = p$

$P_h = l$ $P_i = m$ $P_j = m$ $P_k = n$

$P_e = i$ $P_f = j$ $P_g = j$ ou k

$P_c = f$ $P_d = g$

$P_q = c$

24 adições

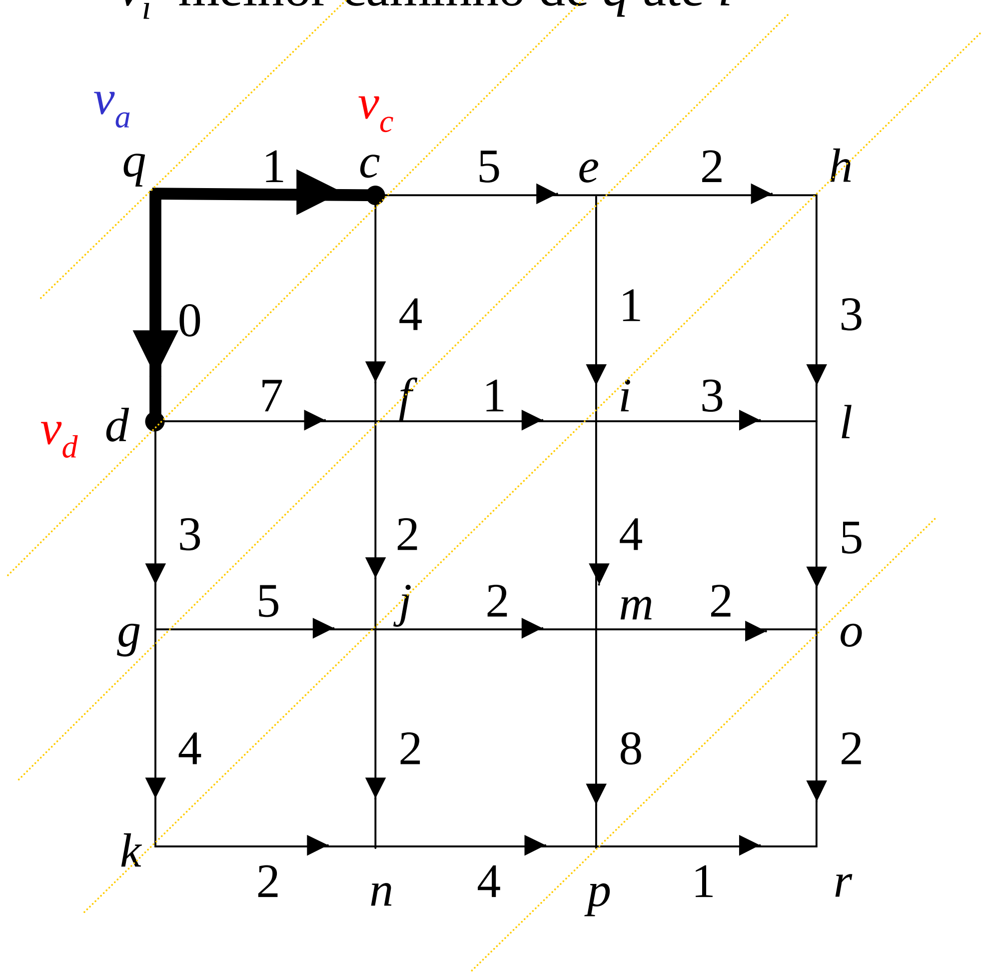
9 comparações

4-Análise de complexidade

- Função objetivo (custo, utilidade, etc.): aditivamente separável
- Ambientes estocásticos: sistemas Markovianos
- Enumeração exaustiva: $O(|A|^n)$
  - $|A|$ número decisões (ações) em cada estágio (passo)
- Programação dinâmica: $O(n|A||S|)$
  - $|S|$ número de estados possíveis
  - n: número de estágios

5-Programação dinâmica forward

v_i melhor caminho de q até i



$v_r = \min\{2 + v_o, 1 + v_p\}$
$v_o = \min\{5 + v_l, 2 + v_m\}$
$v_m = \min\{4 + v_i, 2 + v_j\}$
$v_l = \min\{3 + v_h, 3 + v_i\}$
$v_n = \min\{2 + v_j, 2 + v_k\}$

....

$v_e = \min\{5 + v_c\}$
$v_f = \min\{4 + v_c, 7 + v_d\}$
$v_g = \min\{3 + v_d\}$

$v_d = 0$
$v_c = 1$

$v_q = 0$

6-Exemplos

- Determinístico: caminho mínimo

$I =$ conjunto de nós

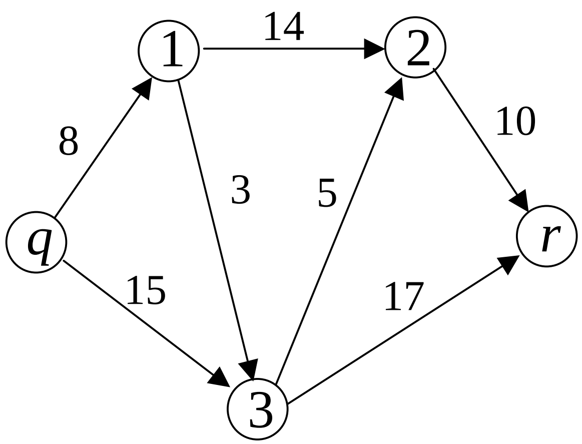
$L =$ conjunto de arcos (i, j)

$c_{ij} =$ custo de i para j

$I_i^+ =$ conjnto de nós j tal que $\exists\ (i, j) \in L$

$I_j^- =$ conjnto de nós i tal que $\exists\ (i, j) \in L$

$v_j =$ customínimo de j para r



- Equação de Bellman

$$v_i \leftarrow min\{v_i, \min_{j \in I_j^+}(c_{ij} + v_j)\}, \ \forall i \in I$$

- Solução iterativa

Iteração	custo do caminho a partir do nó				
	q	1	2	3	r
	100	100	100	100	0
1	100	100	10	15	0
2	30	18	10	15	0
3	26	18	10	15	0
4	26	18	10	15	0

- Algoritmo de Pape

$$v_j = \begin{cases} M & j \neq r \\ 0 & j = r \end{cases}$$

$C = \{q\}$ lista de candidatos

1. remover nó $j \in C$ do topo de C
2. $\forall j \in I_j^+$

   $\hat{v}_i = c_{ij} + v_j$

   se $\hat{v}_i < v_i$ então $v_i = \hat{v}_i$

   se $i \notin C$ então $C = C \cup \{i\}; i$ no fim de C
3. remover j de C. Se $C \neq \phi$ então passo 1 senão fim

- Pape (e Djkstra) são instâncias do algoritmo geral

- Algoritmo geral caminho mínimo

remover nó i da lista de candidatos C

para cada arco $(i, j) \in L$

se $v_j > v_i + c_{ij}$ então

$v_j = v_i + c_{ij}$

adicionar j à V se $j \notin C$

Proposição: sejam $v_1, v_2, ..., v_N$ escalares satisfazendo

$v_j \le v_i + c_{ij}\ \forall (i, j) \in L$

e seja P um caminho iniciando em um nó i_1 e terminando em um nó i_k. Se

$v_j = v_i + c_{ij}$ para todos arcos (i, j) de P

então P é menor caminho de i_1 para i_k.

Prova

Somando $v_j = v_i + c_{ij}$ para arcos de P $\rightarrow$ valor de $P = v_{ik} - v_{i1}$
Somando $v_j \le v_i + c_{ij}$ para arcos de P' $\rightarrow$ valor de $P' \ge P$
Logo, P é o menor caminho.

- Estocástico: atribuição dinâmica

  – atributos de um técnico

$$a_t = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} localzação\ técnico \\ tipo\ equipamento \\ \#\, dias\, no\ trabalho \end{pmatrix}$$

  – conjunto de todos técnicos

$$R_{ta} = \#\ t\text{écnicos com atributo } a$$
$$A = \text{conjnto dos valores de } a$$
$$R_t = (R_{ta})_{a \in A}$$

– demanda serviços técnicos

$b =$ atributos de um equipamento (localização, tipo)

$B =$ conjunto dos valores de b

$\hat{D}_{tb} = \#$ equipamentos tipo b instalados entre t e $t-1$(necessitaserviço)

$\hat{D}_t = (\hat{D}_{tb})_{b \in B}$

$D_{tb} =$ total equipamento tipo b a ser instalado no instante t

$D_t = (D_{tb})_{b \in B}$

– decisões

$$D^{H} = \text{conjunto decisões enviando técnico p/ casa}$$

$$d \in D^{H} = \text{representa um local particular}$$

$$D^{D} = \text{conjnto decisões enviando técncio p/ demanda}$$

$$d^{\phi} = \text{decisão "fazer nada" com um t'ecnico}$$

$$D = D^{H} \cup D^{D} \cup d^{\phi}$$

– impacto decisões nos atributos: função transição

$$a_{t+1} = a^M(a_t, d)$$

$$\delta_{a'}(a_t, d) = \begin{cases} 1 & \text{se } a^M(a_t, d) = a' \\ 0 & \text{caso contrário} \end{cases}$$

– indicação das decisões tomadas

$$x_{tad} = \text{número vezes decisão } d \text{ é aplicada técnico atributo } a$$

$$x_t = (x_{tad})_{a \in A, d \in D}$$

– indicação das decisões tomadas

$$c_{tda} = \text{custo decisão } d \text{ é aplicada técnico atributo } a$$

$$c_t = (c_{tda})_{a \in A, d \in D}$$

- Modelo míope

$$\min_{x_t} \sum_{a \in A} \sum_{d \in D} c_{atd} x_{atd}$$

$$s.a. \quad \sum_{d \in D} x_{tad} = R_{ta}$$

$$\sum_{a \in A} x_{tad} \leq D_{tb_d} \, , \, d \in D^D$$

$$x_{tad} \geq 0$$

– Dinâmica do sistema

$$R_{t+1,a} = \sum_{a' \in A} \sum_{d \in D} x_{ta'd} \delta_a(a', d)$$

$$D_{t+1,b_d} = D_{t,b_d} - \sum_{a \in A} x_{tad} + \hat{D}_{t+1,b_d}, \; d \in D^D$$

– Estado do sistema

$$S_t(R_t, D_t)$$

- Modelo atribuição dinâmica

$$V_t = \min_{x_t \in X_t} (C_t(S_t, x_t) + \gamma E V_{t+1}(S_{t+1}))$$

$$X_t = \{x_t \mid \sum_{d \in D} x_{tad} = R_{ta};\ \sum_{a \in A} x_{tad} \le D_{tb_d},\ d \in D^D;\ x_{tad} \ge 0\}$$

Observação

Este material refere-se às notas de aula do curso IA 718 Tópicos em Sistemas Inteligentes da Faculdade de Engenharia Elétrica e de Computação da Unicamp. Não substitui o livro texto, as referências recomendadas e nem as aulas expositivas. Este material não pode ser reproduzido sem autorização prévia dos autores. Quando autorizado, seu uso é exclusivo para atividades de ensino e pesquisa em instituições sem fins lucrativos.