O problema dos K trajectos mais curtos
e
o problema dos K caminhos mais curtos

Marta Pascoal†
Deolinda Rasteiro†
José Luis Santos†

{*marta, dml, zeluis*}*@mat.uc.pt*

Centro de Informática e Sistemas da Universidade de Coimbra
Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Braga, Abril de 1999

Os autores pertencem à Linha *Ciências da Computação* do Centro de Informática e Sistemas da Universidade de Coimbra, parcialmente subsidiado pelo Ministério das Ciências e Tecnologia através do Projecto PRAXIS XXI da Junta Nacional de Investigação Científica e Tecnológica.

http://www.mat.uc.pt/~marta/Publicacoes/tese.ps.gz
http://www.mat.uc.pt/~eqvm/CISUC/cisuc.html

† - Sob a orientação do Professor Doutor Ernesto de Queirós Vieira Martins.

Resumo

O problema dos K trajectos mais curtos e o problema dos K caminhos mais curtos

Apresenta-se o Problema da determinação ordenada de K trajectos mais curtos entre dois nós de uma rede, alguns subproblemas deste e possíveis aplicações práticas. Após uma breve referência a algumas das características são descritos alguns dos processos conhecidos para a resolução deste problema.
Apresenta-se em seguida o Problema dos K caminhos mais curtos, subproblema do anterior onde os trajectos determinados não contêm nós repetidos. Devido à restrição imposta neste problema é requerida uma abordagem diferente da anterior. É no entanto apresentada uma adaptação de uma classe de algoritmos para o Problema dos K trajectos mais curtos.

O problema dos K trajectos mais curtos

Descrição

Dados:

- Rede $(\mathcal{N}, \mathcal{A})$;
- Nós inicial e terminal $s, t \in \mathcal{N}$;
- Custo de $(i, j) \in \mathcal{A}$, c_{ij};
- Custo de $p \in \mathcal{P}$, $c(p) = \sum_p c_{ij}$;
- K inteiro positivo.

Determinar:

1. p_1, trajecto mais curto de s para t;
2. p_2, 2º trajecto mais curto de s para t;

...

K. p_K, $K^{\underline{\text{ésimo}}}$ trajecto mais curto de s para t.

Subproblemas

Considerando restrições nos trajectos a determinar:

- Problema dos K trajectos mais curtos nó-disjuntos;
- Problema dos K trajectos mais curtos arco-disjuntos;
- Problema dos K caminhos (trajectos sem ciclos) mais curtos;

....

Aplicações

Problemas que exijam:

1. A determinação de um trajecto.

   Caso de uma rede de estradas ou os movimentos de um *robot*.

2. O planeamento da execução de sucessivas tarefas.

   Problemas PERT/CPM.

3. Programação dinâmica ou técnicas de pesquisa em matrizes e que possam ser descritos como problemas dos trajectos mais curtos.

   Problema da mochila, biologia molecular.

A aplicação nestes campos deve-se a diversas razões:

1. Restrições adicionais.
2. Avaliação de modelos.
3. Análise de sensibilidade.
4. Geração de alternativas.

Princípio de optimalidade

Generalização do Princípio de Optimalidade para $K = 1$:

> p_k, é constituído por subtrajectos j-ésimos mais curtos, para algum $1 \leq j \leq k$.

Este problema verifica o Princípio de Optimalidade desde que:
$(\mathcal{N}, \mathcal{A})$ não contenha ciclos de custo negativo.

Algoritmos de rotulação

- Dreyfus, 1969:
  - Determina os K trajectos mais curtos de $i \in \mathcal{N}$ para t.
  - Necessita de 3 matrizes $K \times n$.
  - Associa a cada $i \in \mathcal{N}$ K rótulos.
- Shier, 1974:
  - Determina os K trajectos mais curtos de s para $i \in \mathcal{N}$.
  - Generaliza as formas do algoritmo de rotulação para $K = 1$.
  - Associa a cada $i \in \mathcal{N}$ um K-rótulo, $(\pi_i^1, \ldots, \pi_i^K)$.

Algoritmos de apagamento

- Original de Martins, 1984;
  - Ideia base:
    1. Determinar p_1.
    2. $(\mathcal{N}_1, \mathcal{A}_1)$ é a rede $(\mathcal{N}, \mathcal{A})$.
    3. Para $k \in \{2, \ldots, K\}$:
       (a) $(\mathcal{N}_k, \mathcal{A}_k)$ é a rede obtida da anterior onde é apagado p_{k-1}.
       (b) p' é o trajecto mais curto em $(\mathcal{N}_k, \mathcal{A}_k)$.
       (c) p_k é o trajecto correspondente a p' em $(\mathcal{N}, \mathcal{A})$.

- ...sujeito a sucessivos melhoramentos...

- Último de Martins e Santos, 1995, conhecido por MS.
  - Evita o aumento do número de arcos da rede, reduzindo assim o espaço de memória necessário.

Árvore dos K trajectos mais curtos

Os K trajectos, $p_1, \ldots, p_K$, formam uma árvore de trajectos:

- $\mathcal{T}_1$ é p_1;
- $\mathcal{T}_k$ é $\mathcal{T}_{k-1}$ "+" p_k, onde p_k coincide com p_1 de s até um certo nó,
  $\ldots,$
  com p_{k-1} de s até um certo nó.

Então:

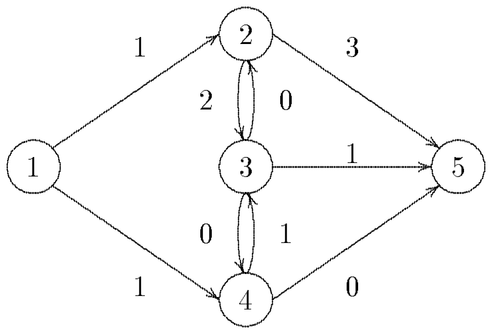
$$p_k = \text{sub}_k(s, \alpha_k) \diamond q_k,$$

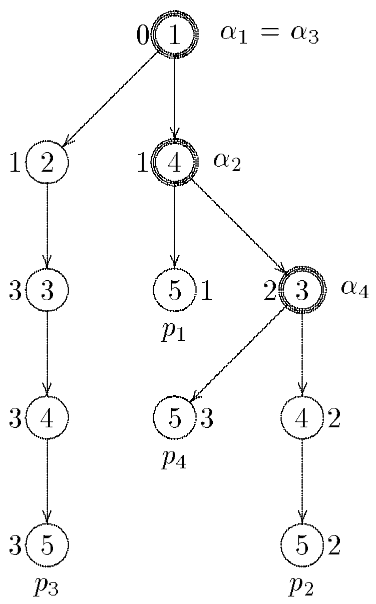
onde $\text{sub}_k(s, \alpha_k)$ é o "maior" subtrajecto que p_k tem em comum com $p_1, \ldots, p_{k-1}$.

Definições:

- α_k é nó desvio.
- $(\alpha_k, x) \in p_k$ é arco desvio.

Exemplo:





$p_1 = \langle \mathbf{1}, 4, 5 \rangle$

$p_2 = \langle 1, \mathbf{4}, 3, 4, 5 \rangle$

$p_3 = \langle \mathbf{1}, 2, 3, 4, 5 \rangle$

$p_4 = \langle 1, 4, \mathbf{3}, 5 \rangle$

Objectivo do problema:

- Construir uma árvore que contenha os K trajectos mais curtos.

Observações:

1. Esta determinação implica (a maior parte das vezes) a construção de uma sobre-árvore.
2. Então, tenta-se construir uma árvore que contenha $p_1, \ldots, p_K$ e com o menor número possível de trajectos.

Algoritmos dos desvios

1. Usar $X = \{\text{candidatos a } p_k\}$.
   (Inicialmente $X = \{p_1\}$.)
2. No passo k do algoritmo escolher o trajecto mais curto em X, p_k.
3. A partir de p_k gerar novos trajectos, candidatos a p_j, com $j > k$.

Notação:

1. $\mathcal{T}_t$ a árvore dos trajectos mais curtos de todos os nós para t.
2. $\mathcal{T}_t(i)$ o trajecto de i para t em $\mathcal{T}_t$.

Algoritmo de Hoffman e Pavley

A partir de cada p_k gerar todos os desvios possíveis:

1. Para cada $x \in p_k$ **seguinte** a α_k gerar **todos os trajectos novos** mais curtos possível.

   Novos trajectos:

   $$\mathrm{sub}_k(s, x) \diamond (x, y) \diamond \mathcal{T}_t(y),$$

   para todo o $(x, y) \in \mathcal{A} - \mathcal{T}_t$.

Algoritmo Elementar

A partir de cada p_k gerar o menor número possível de desvios:

1. Para cada $x \in p_k$ **coincidente ou seguinte** a α_k gerar **um trajecto novo** o mais curto possível.

   Novo trajecto:

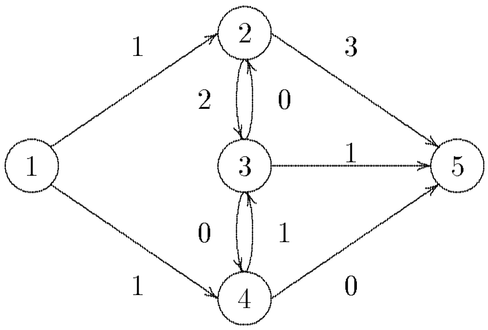
   $$\mathrm{sub}_k(s, x) \diamond q(x),$$

   com $q(x)$ trajecto mais curto de x para t, cujo primeiro arco não está na árvore já determinada.

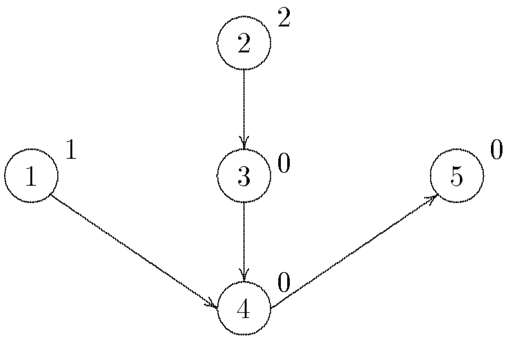
   Isto é:

   $$q(x) = (x, y) \diamond \mathcal{T}_t(y)$$

   trajecto mais curto com (x, y) não pertencente à árvore construída.

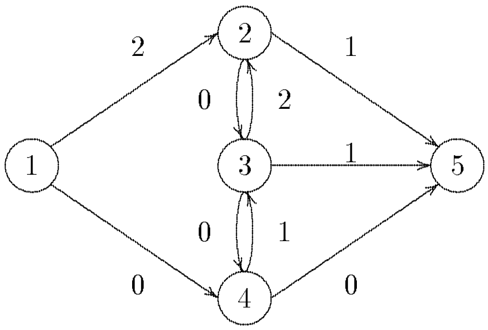


Rede $(\mathcal{N}, \mathcal{A})$

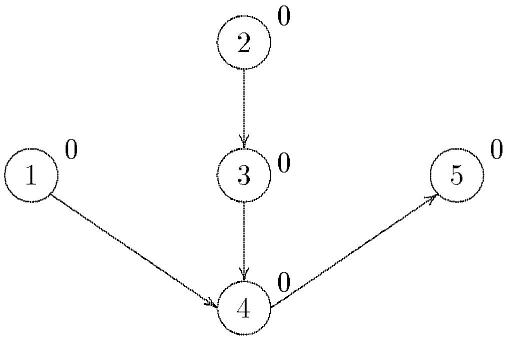


Árvore $\mathcal{T}_t$

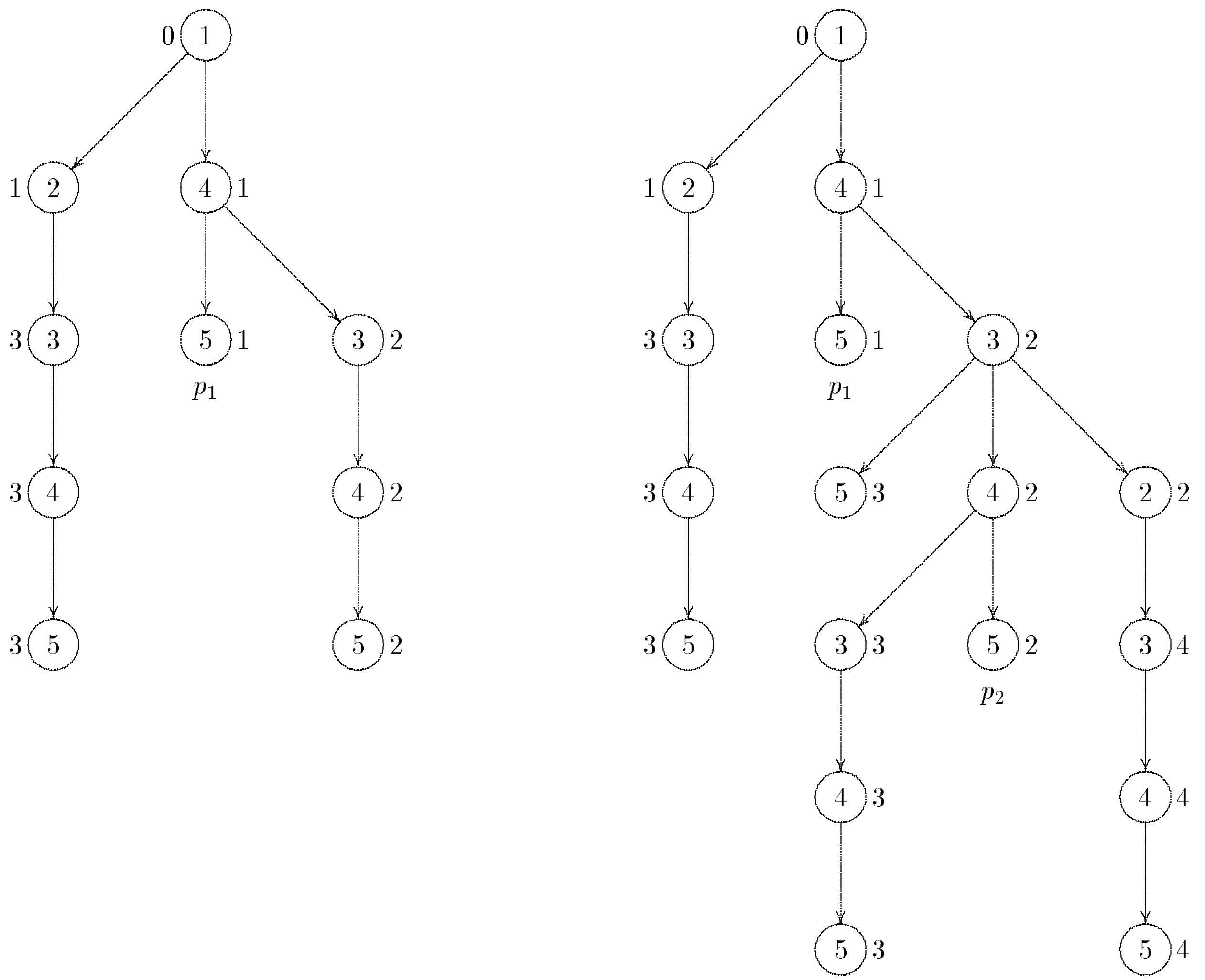
(i,j)	$(1,4)$	$(1,2)$	$(2,3)$	$(2,5)$	$(3,4)$	$(3,5)$	$(3,2)$	$(4,5)$	$(4,3)$
$\bar{c}_{ij}$	0	2	0	1	0	1	2	0	1



Rede $(\mathcal{N}, \mathcal{A})$

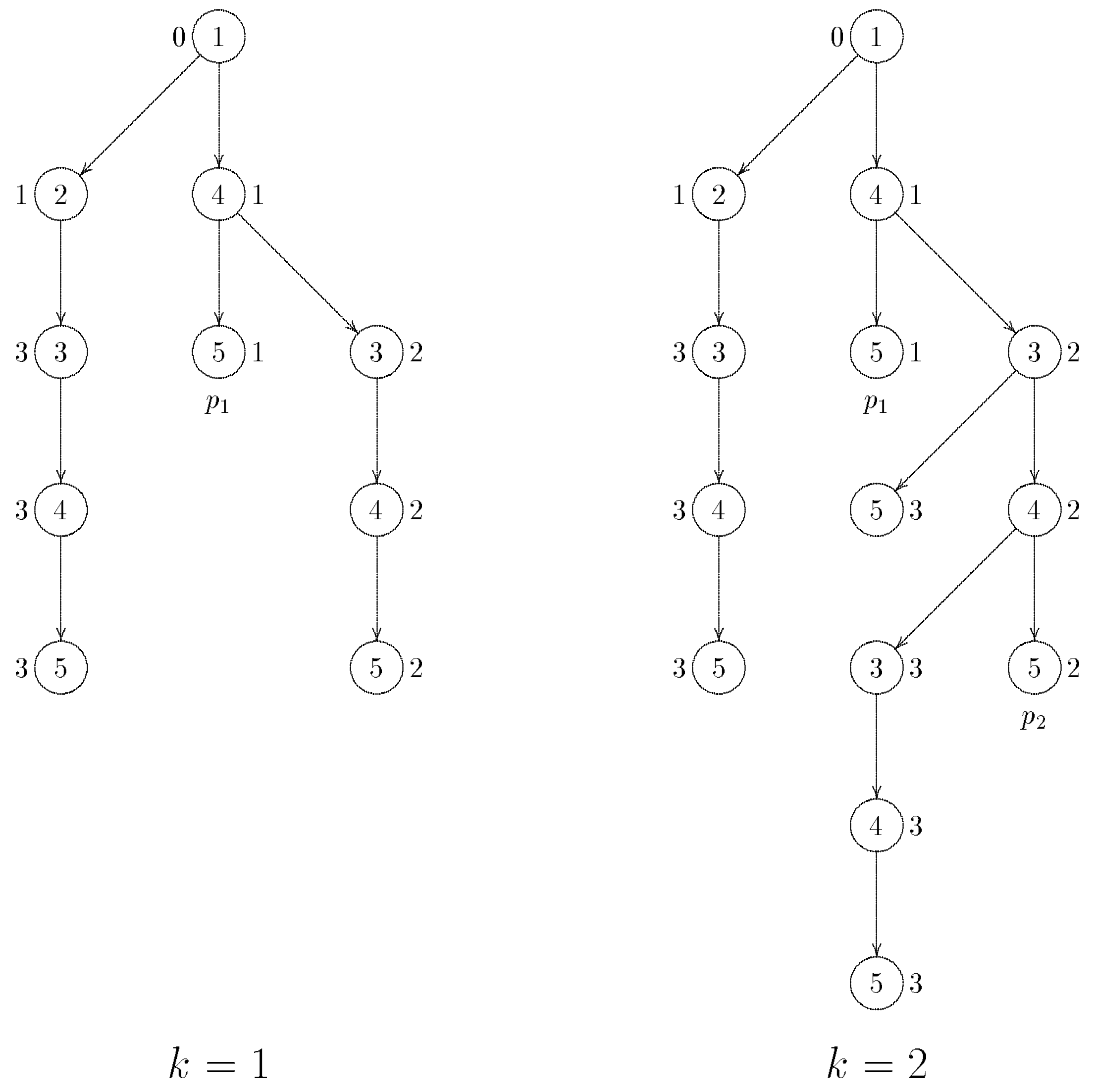


Árvore $\mathcal{T}_t$



$k = 1$ $k = 2$

Algoritmo de Hoffman e Pavley



Algoritmo Elementar

Custos reduzidos

Notação:

1. π_i o custo do trajecto de $i \in \mathcal{N}$ para t em $\mathcal{T}_t$.

Custo reduzido de $(i,j) \in (\mathcal{N}, \mathcal{A})$ associado a $\mathcal{T}_t$:

$$\bar{c}_{ij} = \pi_j - \pi_i + c_{ij}.$$

Custo reduzido de um trajecto p:

$$\bar{c}(p) = \sum_p \bar{c}_{ij}.$$

Propriedades:

1. $c(p) < c(q) \Leftrightarrow \bar{c}(p) < \bar{c}(q)$
   e
   $c(p) = c(q) \Leftrightarrow \bar{c}(p) = \bar{c}(q)$.
2. $\bar{c}_{ij} = 0, \forall (i,j) \in \mathcal{T}_t$.

Algoritmos de Hoffman e Pavley e Elementar com custos reduzidos

Substituir c_{ij} por $\bar{c}_{ij}, \forall (i,j) \in \mathcal{A}$.

Vantagens:

1. Diminui o número de operações aritméticas realizadas, uma vez que se q for subtrajecto de $\mathcal{T}_t$, $\bar{c}(q) = 0$ e então
$$\bar{c}(p \diamond q) = \bar{c}(p).$$
2. No algoritmo elementar, ao analisar $x \in p_k$, escolher (x,y) tal que:
   (a) (x,y) não está na árvore construída e
   (b) (x,y) tem o menor custo reduzido nestas condições.

Sorted forward star form

O custo reduzido $\bar{c}_{ij}$ representa:

1. O custo a acrescentar ao trajecto mais curto de i para t quando se obriga a que seja utilizado o arco (i,j).

Assim, devem usar-se primeiro os arcos com menor custo reduzido.

Ordena-se $\mathcal{A}$ da seguinte forma:

$$\begin{aligned}(i,j) \leq (x,y) \;\Leftrightarrow\; & i < x \\ & \vee \\ & i = x \wedge \bar{c}_{ij} \leq \bar{c}_{xy}\end{aligned}$$

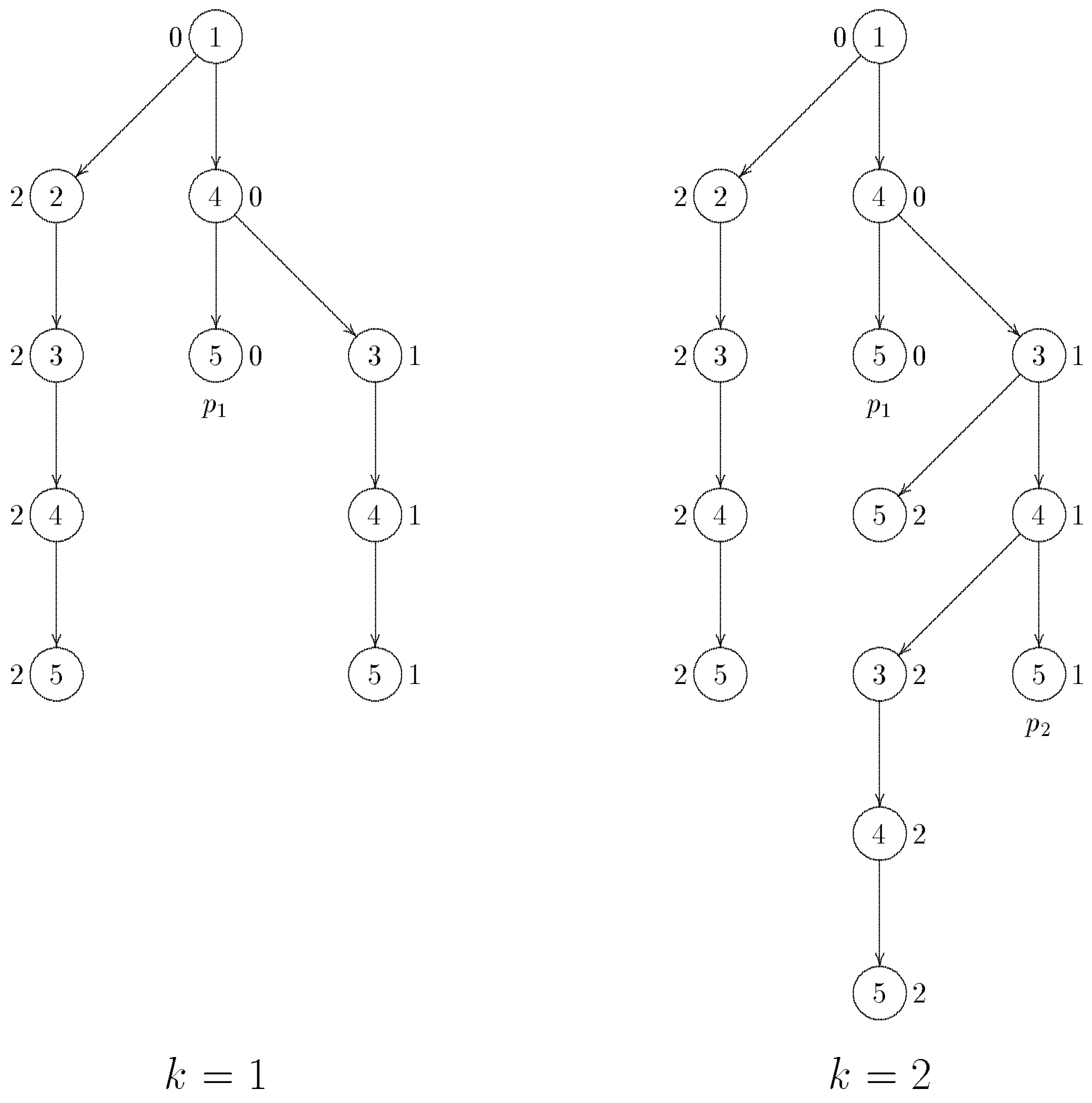
Algoritmo MPS

No início do algoritmo elementar com custos reduzidos $(\mathcal{N}, \mathcal{A})$ é representada na *sorted forward star form*.

Vantagens:

1. Ao analisar o nó $x \in p_k$, basta escolher (x,y) tal que:

   (a) (x,y) é o arco seguinte ao arco com início em x de p_k,



Algoritmo MPS

O problema dos K caminhos mais curtos

Descrição

Caso particular do anterior onde os trajectos são caminhos.

Determinar:

1. p_1, caminho mais curto de s para t;
2. p_2, 2º caminho mais curto de s para t;

   ...

K. p_K, $K^{\underline{\text{ésimo}}}$ caminho mais curto de s para t.

Consequência

A restrição imposta (não determinação de ciclos) resulta:

1. Na não verificação do Princípio de Optimalidade.
2. Na impossibilidade de aplicação, dos algoritmos de rotulação e de apagamento.

Árvore dos K caminhos mais curtos

Considerar (analogamente ao problema geral) uma estrutura de árvore dos K caminhos mais curtos.

Pode então pensar-se em resolver este problema de um modo semelhante.

Algoritmos dos desvios

Algoritmo de Yen

Observação:

1. Algoritmo elementar "é" generalização de Yen para trajectos quaisquer.

Algoritmo:

1′. Usar $X = \{\text{candidatos a } p_k\}$.
(Inicialmente $X = \{p_1\}$.)

2′. No passo k escolher o **caminho** mais curto em X, p_k.

3′. A partir de p_k gerar **novos caminhos**, candidatos a p_j, com $j > k$.

Para cada $x \in p_k$ **coincidente ou seguinte** a α_k gerar **um caminho novo** o mais curto possível.

Novo trajecto:

$$\mathrm{sub}_k(s,x) \diamond q(x),$$

com $q(x)$ caminho mais curto de x para t e $\mathrm{sub}_k(s,x) \diamond q(x)$ também é caminho.
Então $q(x)$ é o caminho mais curto de x para t numa rede obtida de $(\mathcal{N}, \mathcal{A})$ onde se removeram:

1. Os nós (e arcos) de $\mathrm{sub}_k(s,x)$, à excepção de x.

2. Os arcos (x, y) da árvore já construída.

Adaptações dos algoritmos para o problema geral

Algoritmo:

1'. Usar $X = \{\text{candidatos a } p_k\}$.
(Inicialmente $X = \{p_1\}$.)

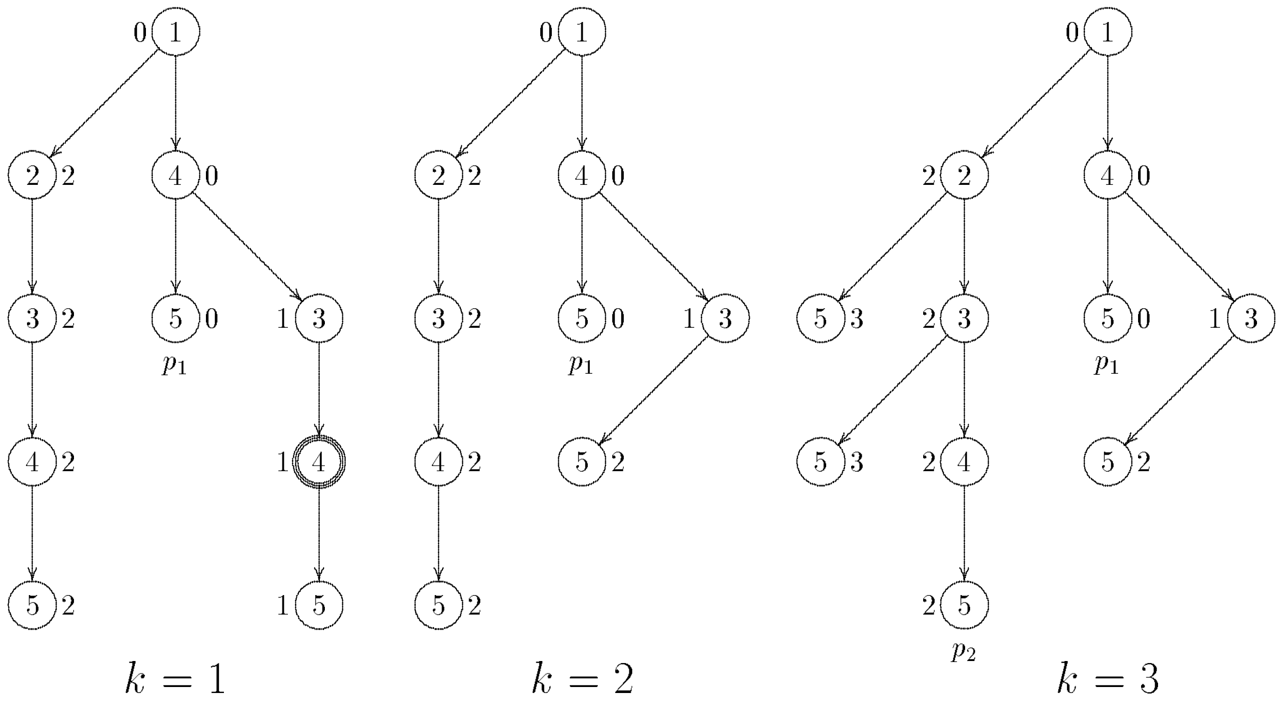
2'. No passo k do algoritmo escolher o **trajecto** mais curto em X, p.
Se p é caminho então $p = p_i$ para algum i.

3'. A partir de p_k gerar **novos trajectos**, minimizando a probabilidade de conterem ciclos.

Ao analisar o nó $x \in p$, gerar

$$\mathrm{sub}_p(s,x) \diamond (x,y) \diamond \mathcal{T}_t(y)$$

onde (x,y) é o arco que se segue ao arco de p com início em x com $y \notin \mathrm{sub}_p(s,x)$.



Particularização do algoritmo MPS para caminhos