UM NOVO MELHORAMENTO DE UM ALGORITMO PARA O PROBLEMA DOS K TRAJECTOS MAIS CURTOS

Marta Pascoal†
e
José Luis Santos†

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

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

IO2000, Setúbal, Abril de 2000

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/io2000_1.ps.gz

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

Resumo

Um novo melhoramento de um algoritmo para o problema dos K trajectos mais curtos

No problema dos K trajectos mais curtos pretende-se enumerar por ordem não decrescente de custos, K trajectos entre um dado par de nós de uma rede. Os algoritmos conhecidos para a resolução deste problema podem dividir-se em duas classes: os algoritmos baseados no Princípio de Optimalidade e os dos desvios, baseados na determinação de uma árvore de trajectos. A primeira destas classes de algoritmos pode ainda dividir-se em algoritmos de rotulação e algoritmos de apagamento, ou das "alternativas".

Nesta comunicação é feita um breve análise da evolução dos algoritmos de apagamento, sendo apresentado um melhoramento relativamente ao mais recente (conhecido por algoritmo MS). Este melhoramento permite uma diminuição da ordem de complexidade teórica aquando da análise do pior dos casos, que se reflecte posteriormente no tempo de execução.

Índice

Introdução 1
Problema 1

Princípio de Optimalidade 2

Algoritmos dos apagamentos ou das alternativas 3
Algoritmo dos "apagamentos" – Martins (1984) 4
Algoritmos das "alternativas" – Azevedo *et al.* (1990) 6
Algoritmo MS – (1995) 9
Melhoramento do algoritmo MS – (1999) 11
Forma incidente ordenada dinâmica (*Dynamic sorted reverse star form*) 11

Estudo comparativo 13
Ordem de complexidade teórica (pior caso) 13
Experiência computacional 14
Tipos de redes 14
Condições das experiências 15
Tempos de execução 15
Conclusões 21

Referências 22

Introdução

Problema

Dados:

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

Observações:

- Trajecto de $\mathcal{P}$, $p = \langle s = v_1, v_2, \dots, v_\ell = t \rangle$, onde $v_1, \dots, v_\ell \in \mathcal{N}$.
- Acrescentando S e T, super-nós inicial e terminal, e arcos (S, s), (t, T) de custo 0, pode admitir-se que:
  - Todo o trajecto tem pelo menos 3 nós;
  - Nenhum arco se inicia em T e nenhum termina em S.

Pretende-se determinar:

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

$\dots$

K. p_K, K-ésimo trajecto mais curto de s para t.

Princípio de Optimalidade

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

> p_k, é constituído por j-ésimos subtrajectos mais curtos de s para cada nó intermédio de p_k, para $1 \leq j \leq k$.

O problema verifica o Princípio de Optimalidade se e só se $(\mathcal{N}, \mathcal{A})$ não contém ciclos negativos.

Baseados neste princípio surgiram dois tipos de algoritmos:

- **Algoritmos de rotulação:**

  Generalizações dos algoritmos de rotulação para o problema do trajecto mais curto.

  – Dreyfus (1969),

  – Shier (1976).

- **Algoritmos dos apagamentos ou das alternativas:**

  Determinam os sucessivos trajectos mais curtos a partir da remoção de trajectos bem determinados.

  – Martins (1984),

  – Azevedo *et al.* (1990),

  – Martins e Santos – MS (1995),

  – Martins *et al.* – Melhoramento de MS (1999).

Algoritmos dos apagamentos ou das alternativas

Forma geral:

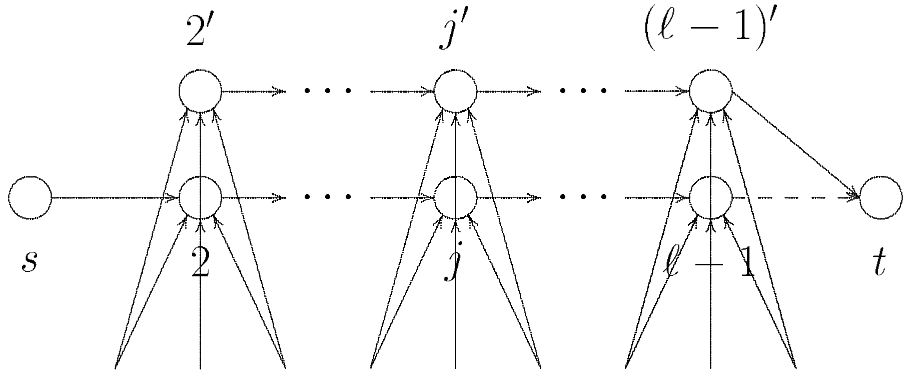
$(\mathcal{N}_1, \mathcal{A}_1) \longleftarrow (\mathcal{N}, \mathcal{A})$
$p'_1 \longleftarrow$ trajecto mais curto de s para t em $(\mathcal{N}_1, \mathcal{A}_1)$
$p_1 \longleftarrow$ trajecto de $(\mathcal{N}, \mathcal{A})$ correspondente a p'_1
`Para` $(i \in \{2, \ldots, K\})$ `Fazer`
  $(\mathcal{N}_i, \mathcal{A}_i) \longleftarrow$ rede obtida de $(\mathcal{N}_{i-1}, \mathcal{A}_{i-1})$ onde se **apagou** p'_{i-1}
  $p'_i \longleftarrow$ trajecto mais curto de s para t em $(\mathcal{N}_i, \mathcal{A}_i)$
  $p_i \longleftarrow$ trajecto de $(\mathcal{N}, \mathcal{A})$ correspondente a p'_i

Como **apagar** um trajecto p de s para t na rede $(\mathcal{N}, \mathcal{A})$?

Pretende-se construir $(\mathcal{N}', \mathcal{A}')$ tal que:

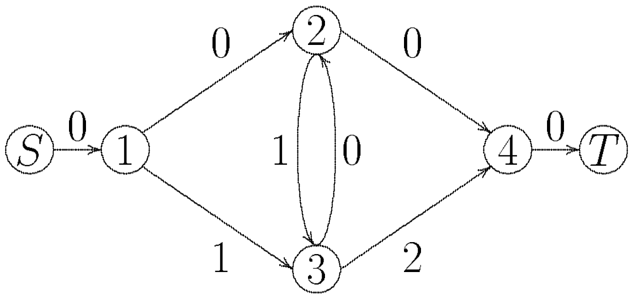
1. Todo o trajecto q' de $(\mathcal{N}', \mathcal{A}')$ corresponde a um único trajecto q de $(\mathcal{N}, \mathcal{A})$.
2. Nenhum trajecto em $(\mathcal{N}', \mathcal{A}')$ tem p por correspondente em $(\mathcal{N}, \mathcal{A})$.

Algoritmo dos "apagamentos" – Martins (1984)

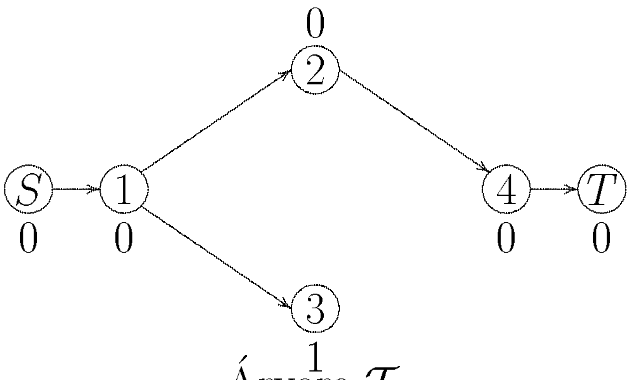


Apagamento de $p = \langle s = v_1, \ldots, v_\ell = t\rangle$ de $(\mathcal{N}, \mathcal{A})$

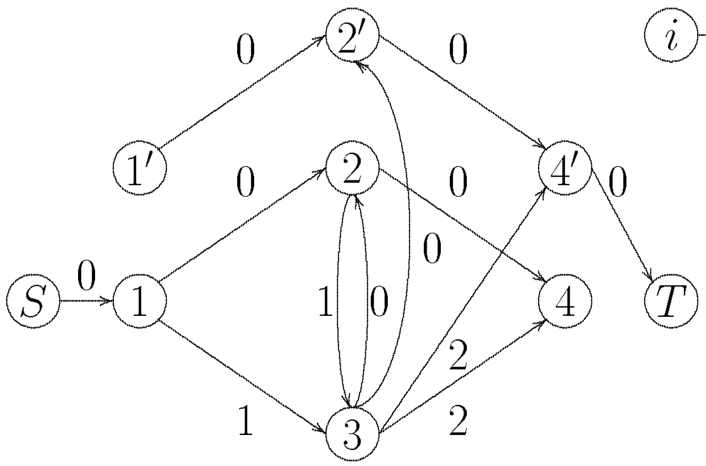
$(\mathcal{N}', \mathcal{A}') \longleftarrow (\mathcal{N}, \mathcal{A})$
`Para` $(j \in \{2, \ldots, \ell-1\})$ `Fazer`
    Criar alternativa, v_j', de v_j
    Criar alternativas dos arcos que incidem em v_j, excepto (v_{j-1}, v_j)
    `Se` $(j \neq 2)$ `Então`
        Criar alternativa de (v_{j-1}, v_j)



Rede $(\mathcal{N}, \mathcal{A}) = (\mathcal{N}_1, \mathcal{A}_1)$



Árvore $\mathcal{T}_s$



Martins – Rede $(\mathcal{N}_2, \mathcal{A}_2)$

Algoritmo das "alternativas" – Azevedo *et al.* (1990)

Teorema: Seja v_α o 1^o nó de $p \in \mathcal{P}$ ao qual chega mais que 1 arco:

1. v_α é o 1^o nó de p para o qual $\mathcal{P}_{sv_\alpha}$ tem mais que 1 trajecto.
2. Existe alternativa para os nós de p que se seguem a v_α. Para os nós de p seguintes a v_α existe mais que 1 trajecto em $\mathcal{P}_{sv_j}$.

Lema: Se foi calculada alternativa para o nó v_j de p_i, então o mesmo acontece com os que o precedem nesse trajecto.

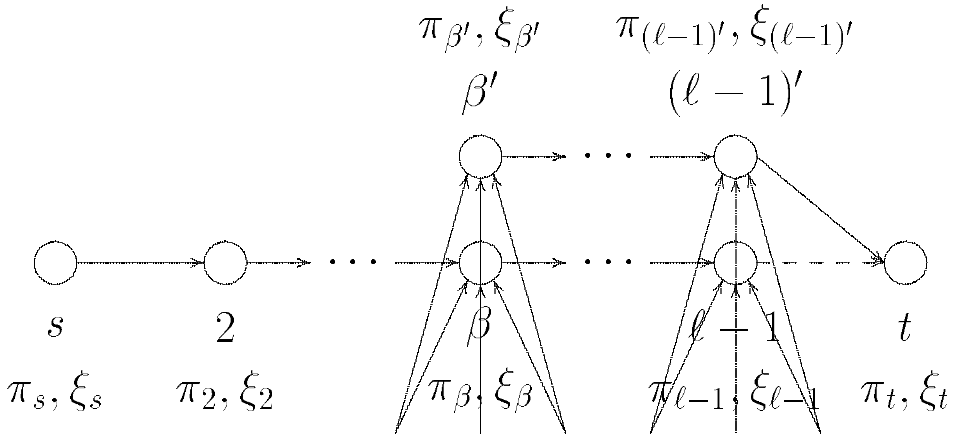
Criam-se alternativas dos nós que seguem o 1^o ao qual chega mais que 1 arco e que não tem alternativa criada.

Teorema: Ao construir $(\mathcal{N}_i, \mathcal{A}_i)$, $i \geq 1$, os rótulos dos nós acrescentados são permanentes.

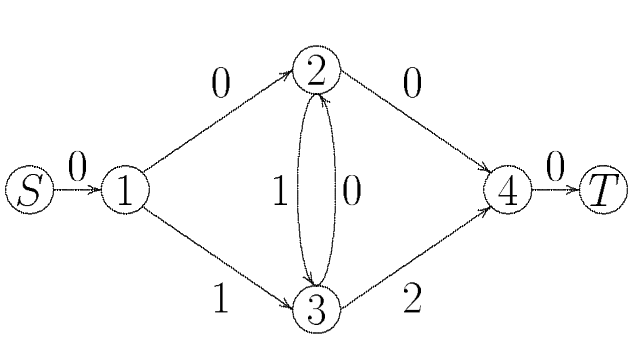
A cada nó x associa-se um rótulo:

1. π_x, custo do trajecto de s para x
2. ξ_x, nó a partir do qual x foi rotulado

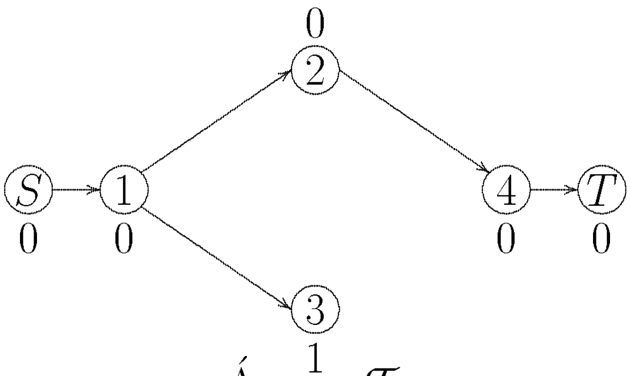
Substitui-se a resolução de um problema do trajecto mais curto apenas pela rotulação dos novos nós criados.



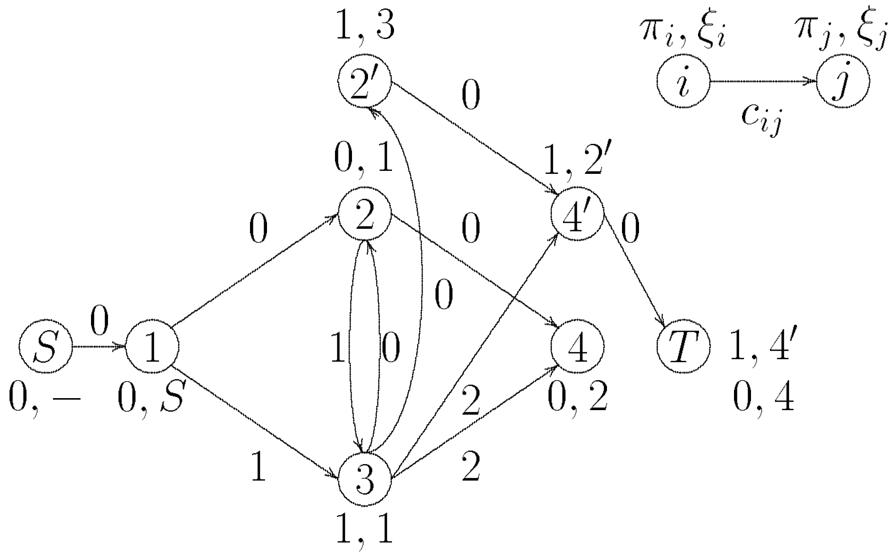
$(\mathcal{N}', \mathcal{A}') \longleftarrow (\mathcal{N}, \mathcal{A})$
$v_\alpha \longleftarrow 1^o$ nó de p ao qual chega mais que 1 arco
$v_\beta \longleftarrow 1^o$ nó a partir de v_α cuja alternativa não foi determinada
`Para` $(j \in \{\beta, \ldots, \ell-1\})$ `Fazer`
    Criar alternativa, v'_j, de v_j
    Criar alternativas dos arcos que incidem em v_j excepto (v_{j-1}, v_j)
    `Se` $(j \neq \beta)$ `Então`
        Criar alternativa de (v_{j-1}, v_j)
    Rotular v'_j



Rede $(\mathcal{N}, \mathcal{A}) = (\mathcal{N}_1, \mathcal{A}_1)$



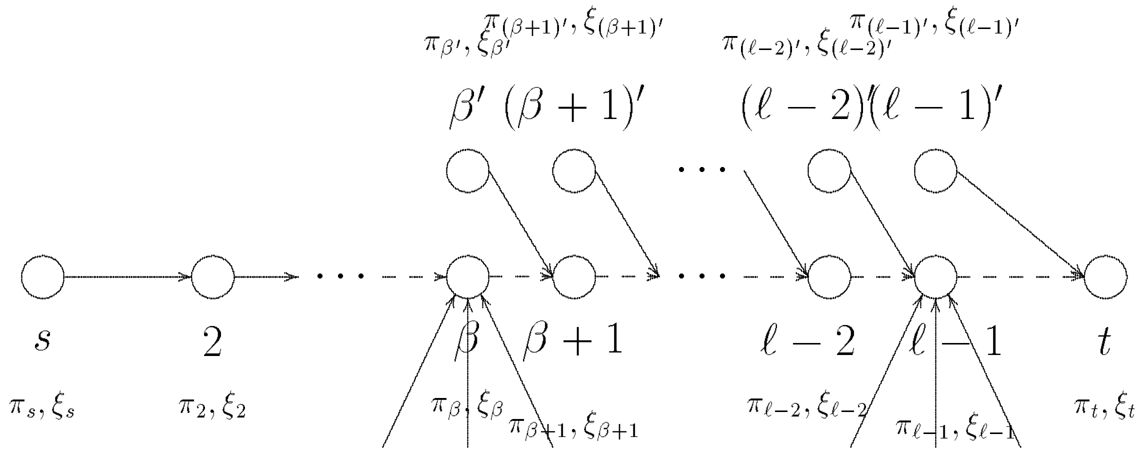
Árvore $\mathcal{T}_s$



Azevedo *et al.* – Rede $(\mathcal{N}_2, \mathcal{A}_2)$

Algoritmo MS – (1995)

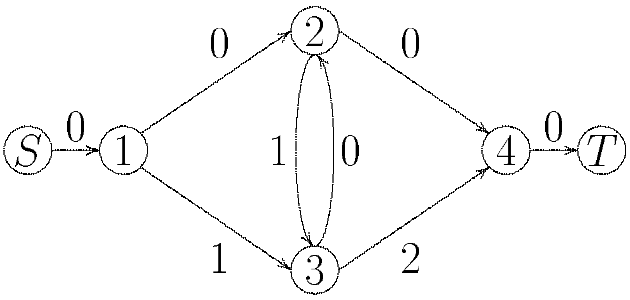
Os conjuntos dos arcos que incidem em v_j e v'_j diferem apenas em (v_{j-1}, v_j), que pode:
1. Não ter alternativa; ou
2. Ter (v'_{j-1}, v'_j) como alternativa.



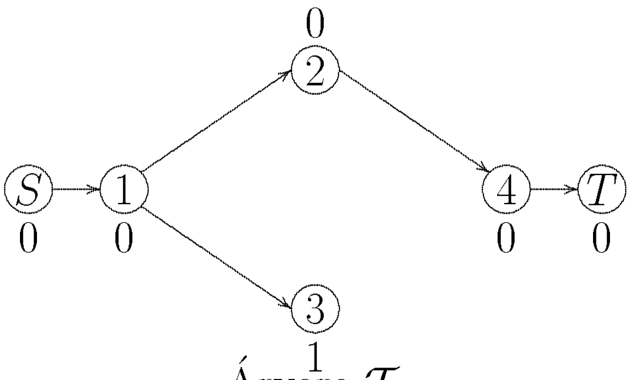
$(\mathcal{N}', \mathcal{A}') \longleftarrow (\mathcal{N}, \mathcal{A})$
$v_\alpha \longleftarrow 1^o$ nó de p ao qual chega mais que 1 arco
$v_\beta \longleftarrow 1^o$ nó a partir de v_α cuja alternativa não foi determinada
`Para` $(j \in \{\beta, \ldots, \ell-1\})$ `Fazer`
  Criar alternativa, v'_j, de v_j
  `Se` $(j \neq \beta)$ `Então` Substituir (v_{j-1}, v_j) por (v'_{j-1}, v_j)
  `Senão` Remover (v_{j-1}, v_j)
  Rotular v'_j

Seja v_j o nó correspondente a v'_j, rotulação de v'_j:

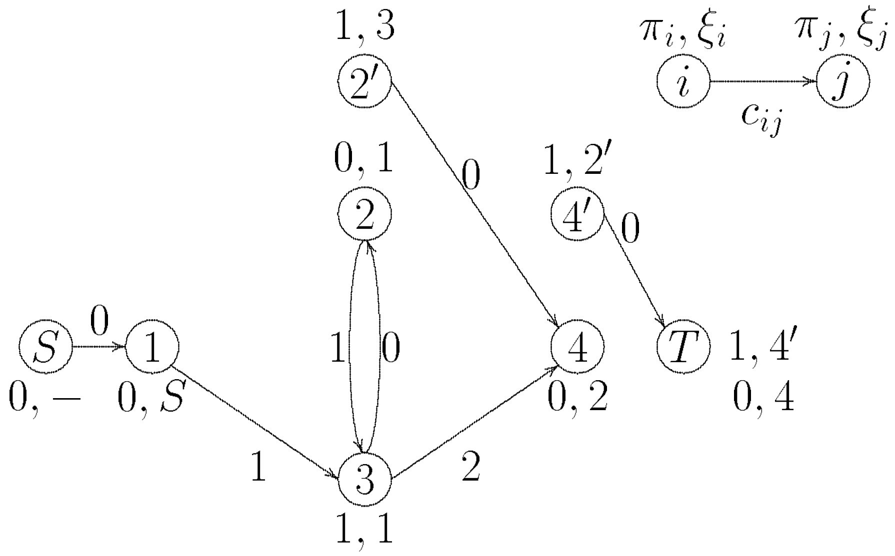
$$\pi_{v'_j} = \min\{\pi_i + c_{iv_j} : (i, v_j) \in \mathcal{A}_k\}$$



Rede $(\mathcal{N}, \mathcal{A}) = (\mathcal{N}_1, \mathcal{A}_1)$



Árvore $\mathcal{T}_s$



MS – Rede $(\mathcal{N}_2, \mathcal{A}_2)$

Melhoramento do algoritmo MS – (1999)

Manter conjunto dos arcos que incidem em v_j por ordem não-decrescente dos "possíveis" rótulos.

Forma incidente ordenada dinâmica

(*Dynamic sorted reverse star form*)

Mantendo $\mathcal{A}_k$ de modo que, $(i_1, j_1), (i_2, j_2) \in \mathcal{A}_k$,

$$\left.\begin{array}{l} j_1 < j_2 \\ \\ \quad \text{ou} \\ \\ j_1 = j_2 \ \text{ e } \left\{\begin{array}{c} \pi_{i_1} + c_{i_1 j_1} < \pi_{i_2} + c_{i_2 j_2} \\ \text{ou} \\ \pi_{i_1} + c_{i_1 j_1} = \pi_{i_2} + c_{i_2 j_2} \ \text{ e } \ (i_1, j_1) \in \mathcal{T}_s \end{array}\right. \end{array}\right\} \Rightarrow (i_1, j_1) \text{"}{<}\text{"} (i_2, j_2)$$

escolhemos o primeiro arco terminando em v_j.

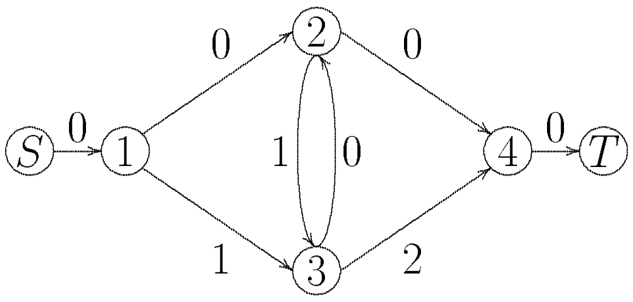
Observação:

Como a rede é alterada várias vezes por remoção/substituição de arcos e criação de nós, cada inserção de arcos tem que ser feita em "ordem".

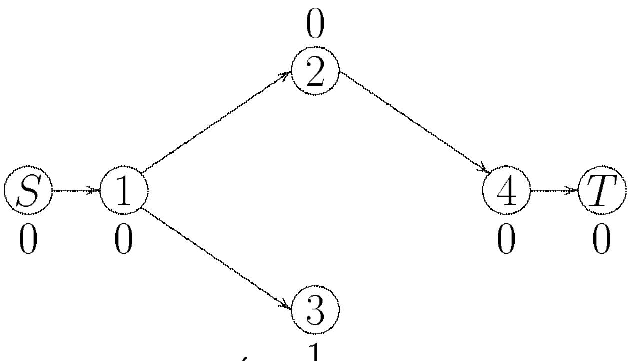
Seja v_j o nó correspondente a v'_j, rotulação de v'_j:

$$\pi_{v'_j} = \pi_i + c_{i v_j}$$

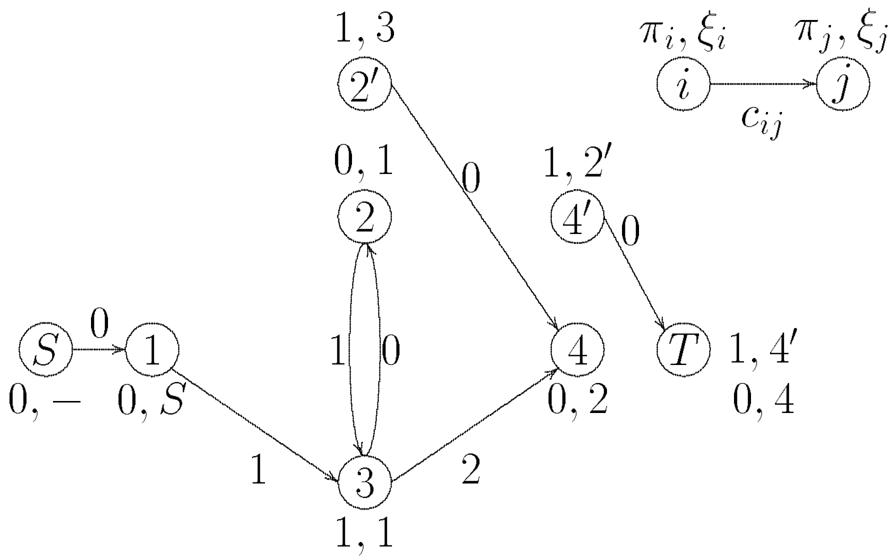
onde i é o 1^o nó de entre os que incidem em v_j.



Rede $(\mathcal{N}, \mathcal{A}) = (\mathcal{N}_1, \mathcal{A}_1)$



Árvore $\mathcal{T}_s$



Melhoramento de MS – Rede $(\mathcal{N}_2, \mathcal{A}_2)$

(i,j)	$(S,1)$	$(1,2)$	$(3,2)$	$(1,3)$	$(2,3)$	$(2,4)$	$(3,4)$	$(4,T)$
$\pi_i + c_{ij}$	0	0	1	1	1	0	3	0

$\mathcal{A} = \mathcal{A}_1$ na forma incidente ordenada dinâmica

(i,j)	$(S,1)$	$(1,2)$	$(3,2)$	$(1,3)$	$(2,3)$	$(2',4)$	$(3,4)$	$(4',T)$
$\pi_i + c_{ij}$	0		1	1	1	1	3	0

$\mathcal{A}_2$ na forma incidente ordenada dinâmica

Estudo comparativo

Ordem de complexidade teórica (pior caso)

Pior caso:
p_k passa k vezes por todos os nós de $(\mathcal{N}, \mathcal{A})$, $k \geq 1$.

Tempo de execução

	MS	Melhoramento MS
Determinar p_1		
(Dijkstra com *heaps*, $c_{ij} \geq 0$)	$\mathcal{O}(m + n \log n)$	$\mathcal{O}(m + n \log n)$
Forma inc. ordenada dinâmica		
(Usando *heaps* binários)		$\mathcal{O}(m \log n)$
Geração de alternativas	$\mathcal{O}(Km)$	$\mathcal{O}(Kn \log n)$
	$\mathcal{O}(Km + n \log n)$	$\mathcal{O}(Kn \log n)$ $(Kn > m)$

Espaço de memória

	MS e melhoramento
Arcos	$\mathcal{O}(m)$
Nós	$\mathcal{O}(Kn)$
	$\mathcal{O}(m + Kn)$

Experiência computacional

Tipos de redes

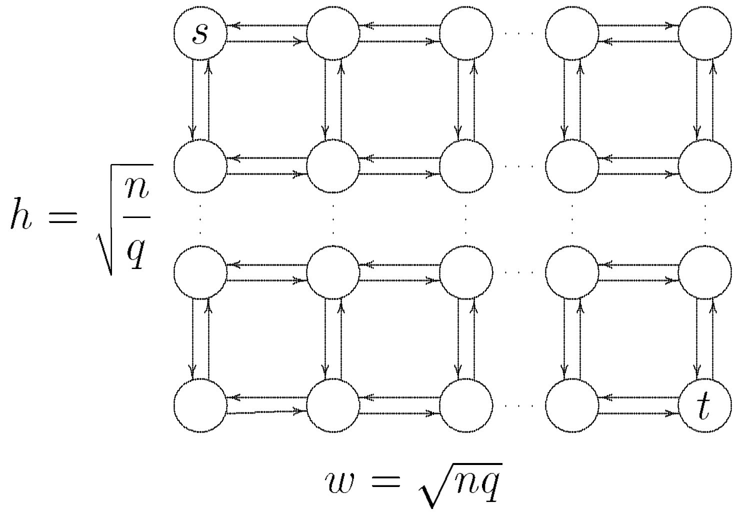
Redes aleatórias

Nós (n)	5000	10000
Arcos (m)	7500, 10000, 75000, 100000	15000, 20000, 150000, 200000
Densidades (m/n)	1.5, 2, 15, 20	

Redes completas

Nós (n)	100, 250, 500, 1000

Grelhas



Grelha com largura w e altura h

Nós (n)	500, 2500, 4500
($q = w^2/n = n/h^2$)	1, 2, 5, 10

Condições das experiências

Características da máquina utilizada (`rena4`):

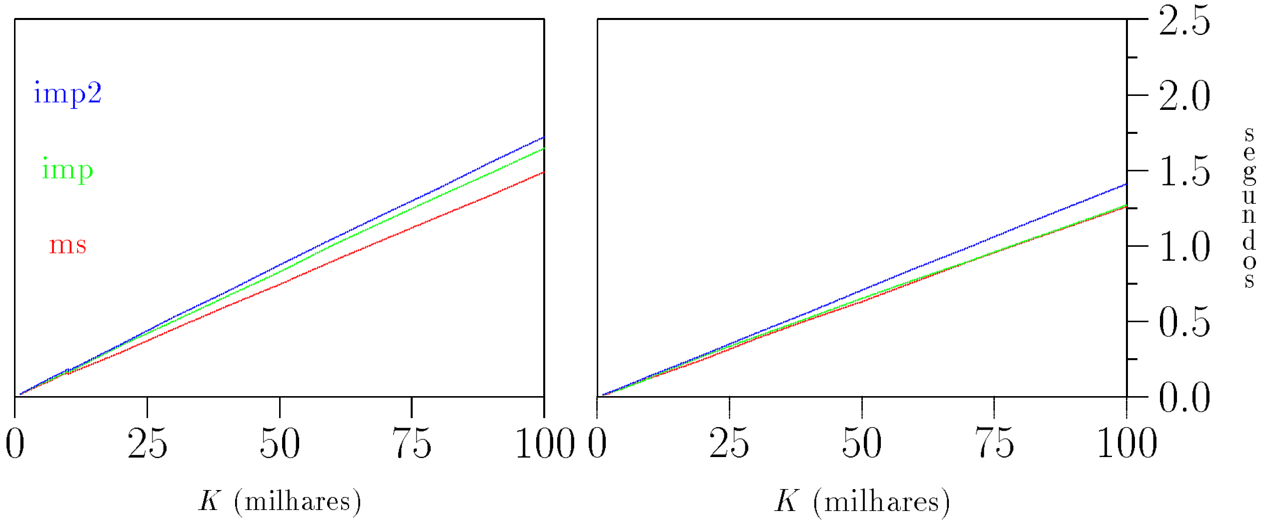
- Pentium II
- 266 MHz
- Sistema Linux

Tempos médios obtidos ao resolver:

- 15 problemas de cada tipo;
- 3 tipos de redes de diferentes dimensões.

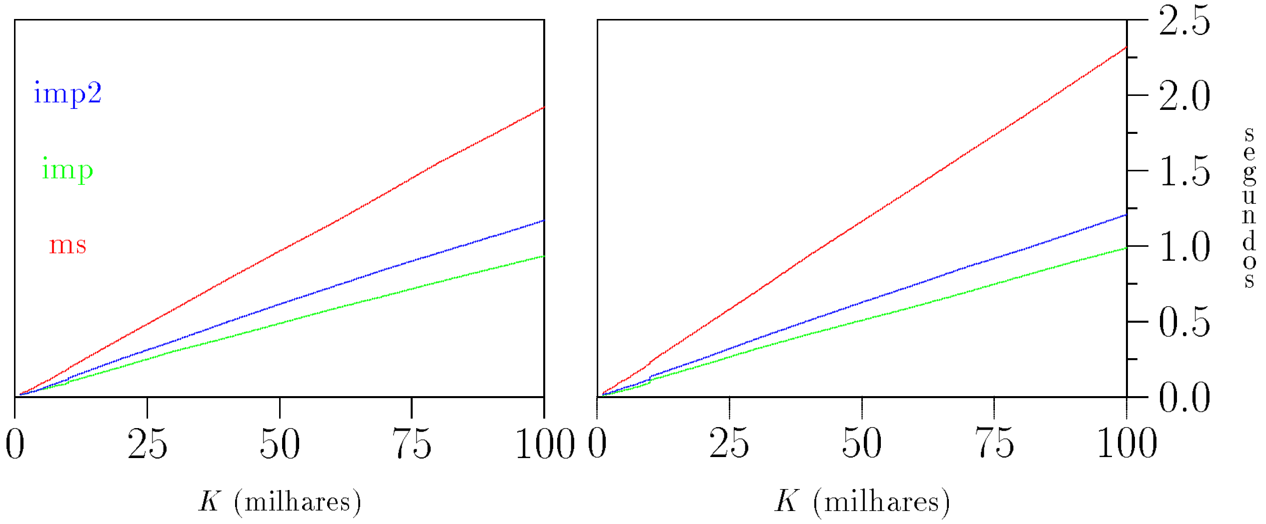
Testadas implementações, em linguagem C, de:

- Algoritmo MS, `ms`.
- Melhoramento do algoritmo MS usando inserção linear, `imp`.
- Melhoramento do algoritmo MS usando *heaps* binários, `imp2`.



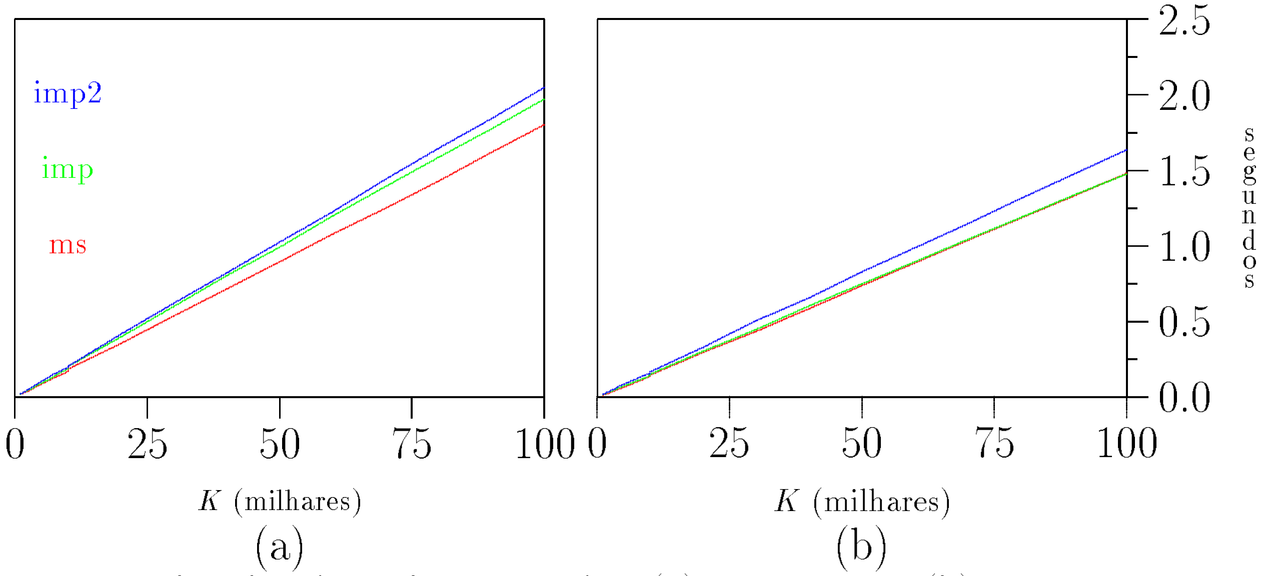
(a) (b)

Redes aleatórias de 5000 nós e (a) 7500 arcos, (b) 10000 arcos

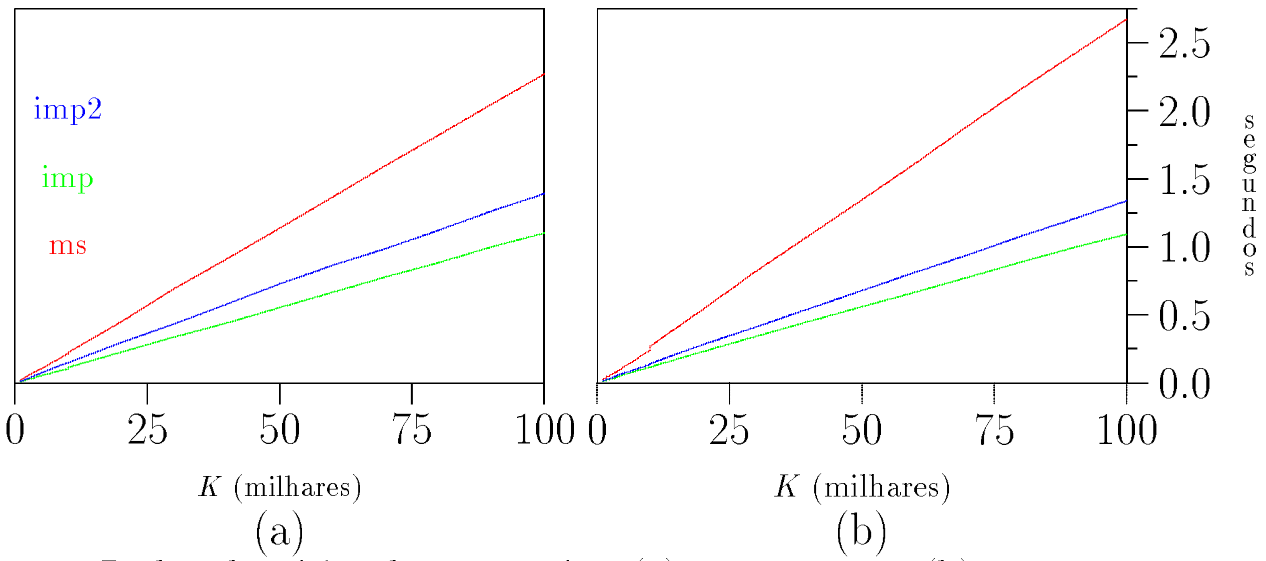


(a) (b)

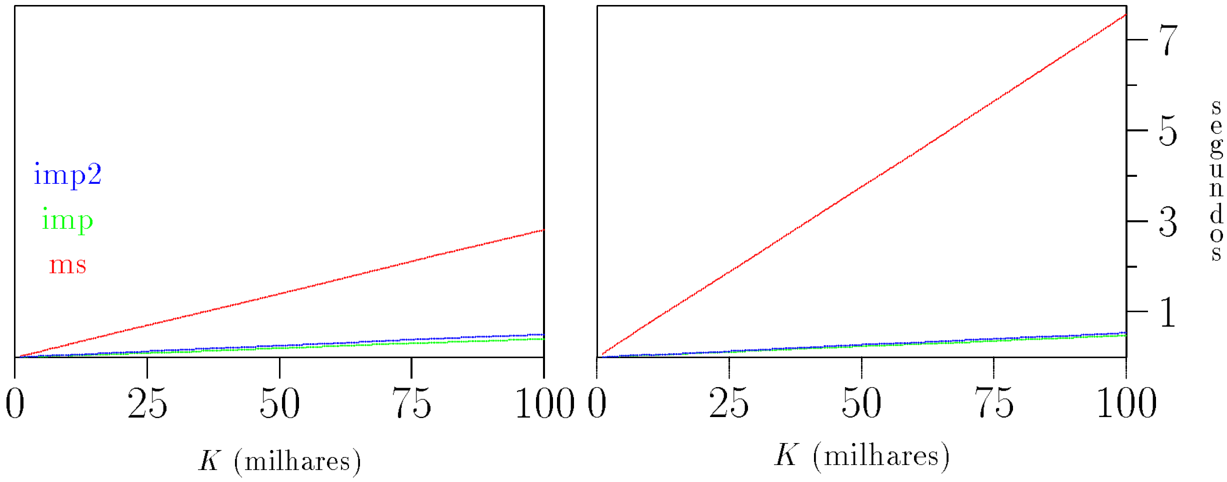
Redes aleatórias de 5000 nós e (a) 75000 arcos, (b) 100000 arcos



Redes aleatórias de 10000 nós e (a) 15000 arcos, (b) 20000 arcos

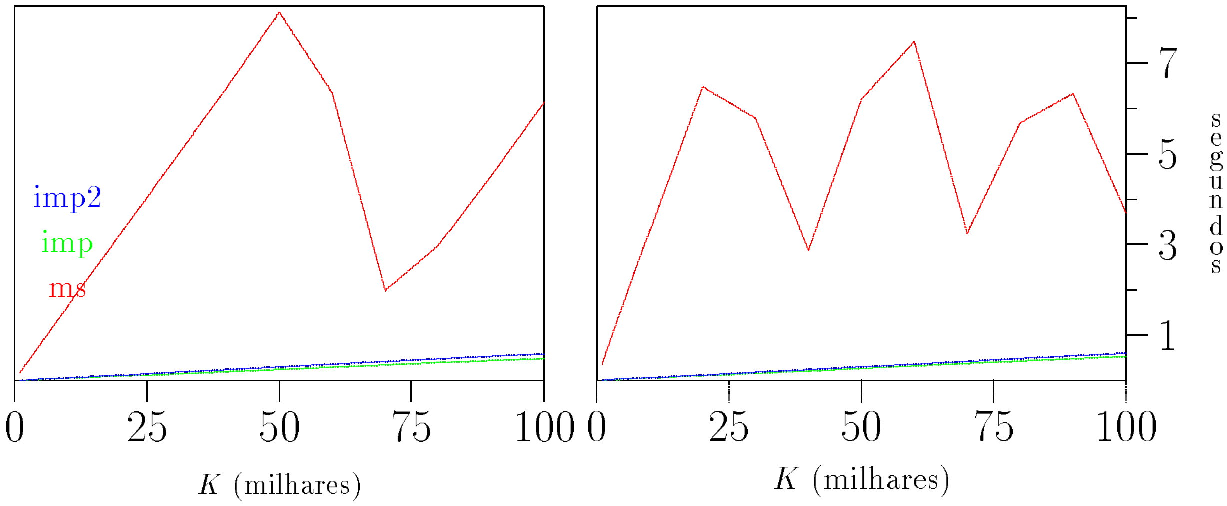


Redes aleatórias de 10000 nós e (a) 150000 arcos, (b) 200000 arcos



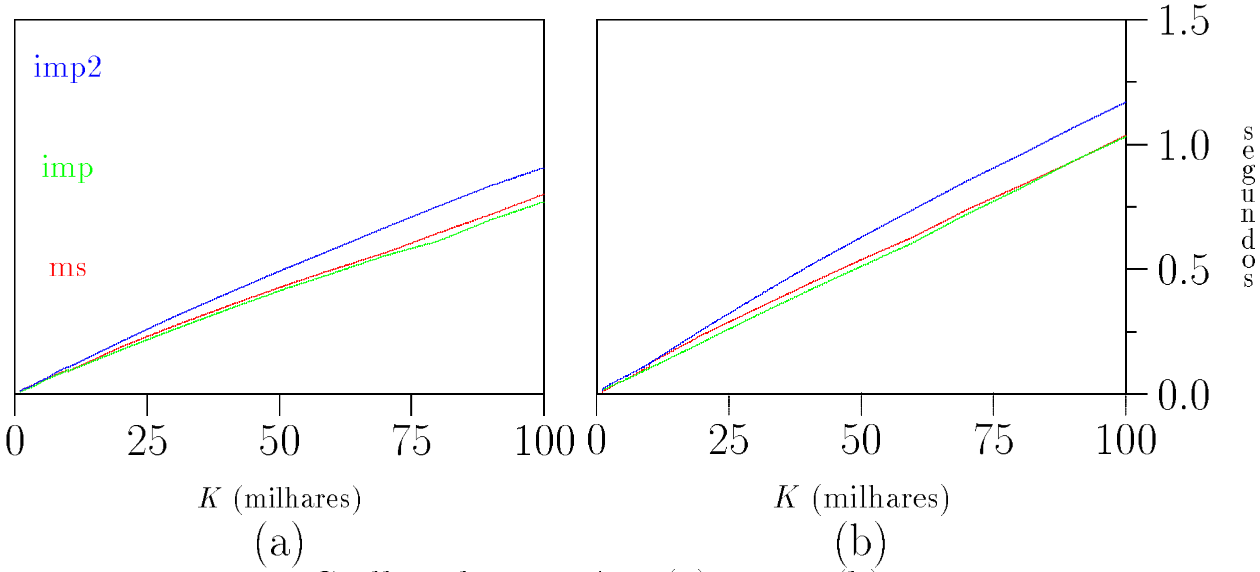
(a) (b)

Redes completas de (a) 100 nós e (b) 250 nós

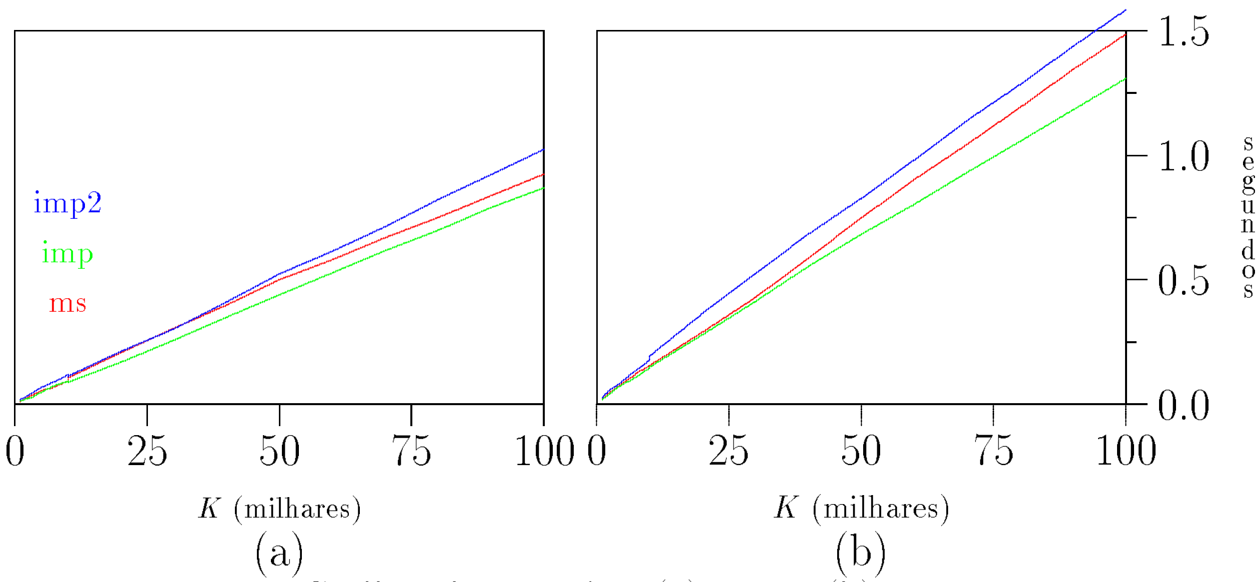


(a) (b)

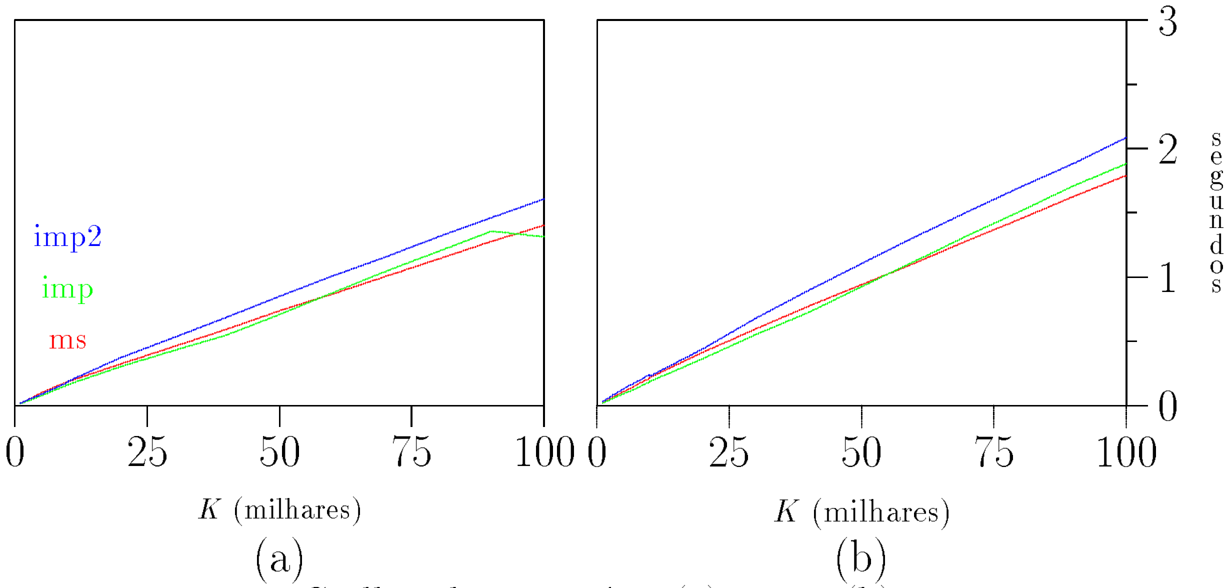
Redes completas de (a) 500 nós e (b) 1000 nós



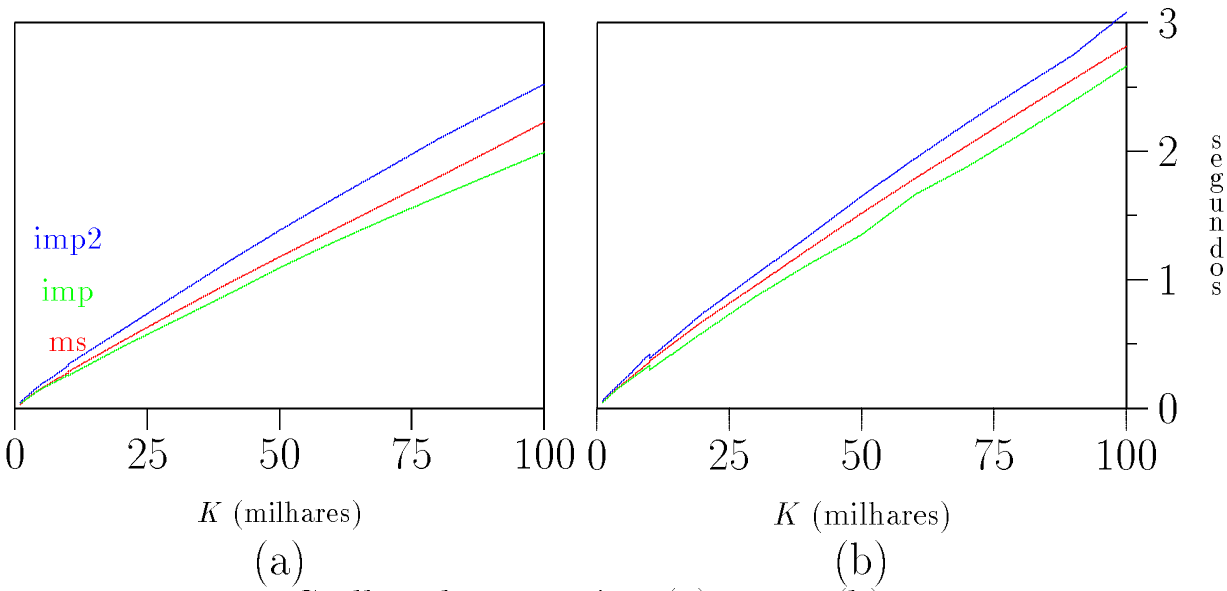
Grelhas de 500 nós e (a) $q = 1$, (b) $q = 2$



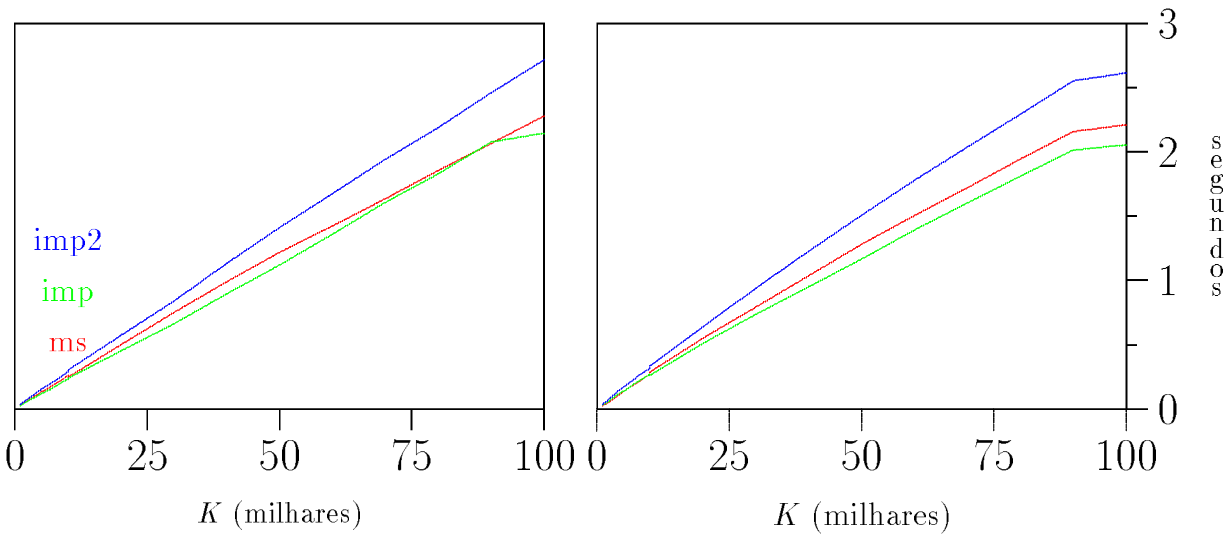
Grelhas de 500 nós e (a) $q = 5$, (b) $q = 10$



Grelhas de 2500 nós e (a) $q = 1$, (b) $q = 2$



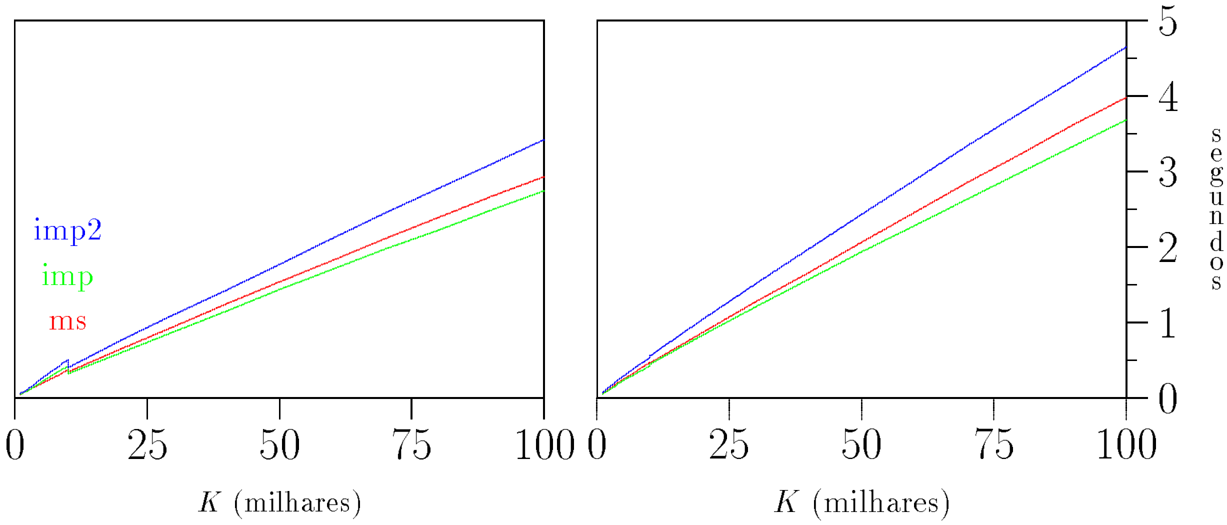
Grelhas de 2500 nós e (a) $q = 5$, (b) $q = 10$



Grelhas de 4500 nós e (a) $q = 1$, (b) $q = 2$

Conclusões

- Os algoritmos testados podem dividir-se em dois grupos:
  - `ms`, cujos tempos de execução aumentam "muito rapidamente" à medida que aumenta d e "ligeiramente" à medida que aumenta n;
  - `imp` e `imp2`, cujos tempos de execução aumentam à medida que aumenta d, e que são pouco sensíveis à variação de n.
- De entre os algoritmos do segundo grupo:
  - `imp`, apresenta, em geral, melhores tempos de execução que a implementação usando *heaps* binários, `imp2`.



(a) (b)

Grelhas de 4500 nós e (a) $q = 5$, (b) $q = 10$

Referências

- S. E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17:395–412, 1969.
- D. Shier. Interactive methods for determining the K shortest paths in a network. *Networks*, 6:151–159, 1976.
- E. Q. V. Martins. An algorithm for ranking paths that may contain cycles. *European Journal of Operational Research*, 18:123–130, 1984.
- J. A. Azevedo, J. J. E. R. S. Madeira, E. Q. V. Martins e F. M. A. Pires. A shortest paths ranking algorithm, 1990. Proceedings of the Annual Conference AIRO'90, Models and Methods for Decision Support, Operational Research Society of Italy, 1001–1011.
- J. A. Azevedo, M. E. O. S. Costa, J. J. E. R. S. Madeira e E. Q. V. Martins. An algorithm for the ranking of shortest paths. *European Journal of Operational Research*, 69:97–106, 1993.

- J.A. Azevedo, J. J. E. R. S. Madeira, E. Q. V. Martins e F. M. A. Pires. A computational improvement for a shortest paths ranking algorithm. *European Journal of Operational Research*, 73:188–191, 1994.
- E. Q. V. Martins e J. L. E. Santos. A new shortest paths ranking algorithm. Aceite para publicação em *Investigação Operacional* (http://www.mat.uc.pt/∼eqvm/cientificos/investigacao/Artigos/K.ps.gz).
- E. Q. V. Martins, M. M. B. Pascoal e J. L. E. Santos. Labeling algorithms for ranking shortest paths. Technical Report 00/001, CISUC, Janeiro 2000, (Submetido). (http://www.mat.uc.pt/∼eqvm/cientificos/investigacao/Artigos/LABELING.ps.gz).
- E. Q. V. Martins, M. M. B. Pascoal e J. L. E. Santos. A New Improvement for a K Shortest Paths Algorithm. Technical Report 99/006, CISUC, Novembro 1999. (http://www.mat.uc.pt/∼eqvm/cientificos/investigacao/Artigos/IMP_MS.ps.gz)