A NEW IMPLEMENTATION OF YEN'S RANKING LOOPLESS PATHS ALGORITHM*

Ernesto Martins
Marta Pascoal

marta@mat.uc.pt

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

Optimization 2001, Aveiro, July 2001

This work was partially supported by POSI — Programa Operacional Sociedade de Informação of Portuguese Fundação para a Ciência e Tecnologia and European Union FEDER.

http://www.mat.uc.pt/~marta/Publicacoes/opt2001.ps.gz

* Issued from: "A new implementation of Yen's ranking loopless paths algorithm", E. Martins and M. Pascoal, submitted for publication in October 2000.

Summary

A new implementation of Yen's ranking loopless paths algorithm

Yen's algorithm is a classical algorithm for ranking the K shortest loopless paths between a pair of nodes in a network. In this paper an implementation of Yen's algorithm is presented. Although with no consequences in what concerns the computational complexity order when considering a worst-case analysis, $\mathcal{O}(Kn(m + n \log n))$, in practice this new implementation outperforms a straightforward one, as the reported comparative computational experiments allow us to conclude.

Contents

Introduction

Given:

- Network $(\mathcal{N}, \mathcal{A})$: set of nodes $\mathcal{N}$ and set of arcs $\mathcal{A}$.
- Arc (i, j) cost, $c_{ij} \in \mathbb{R}$.
- Initial node s and terminal node t, with $s \neq t$.

Path from i to j in $(\mathcal{N}, \mathcal{A})$ is $\langle i = v'_1, a'_1, v'_2, \ldots, a'_{\ell_p-1}, v'_{\ell_p} = j \rangle$, where:

- $v'_k \in \mathcal{N}$ for any $k \in \{1, \ldots, \ell_p\}$;
- $a'_k = (v'_k, v'_{k+1}) \in \mathcal{A}$ for any $k \in \{1, \ldots, \ell_p - 1\}$.

Cycle is a path from i to i where all nodes are different.

Set of paths in $(\mathcal{N}, \mathcal{A})$ from i to j: $\mathcal{P}_{ij}$. $\mathcal{P} = \mathcal{P}_{st}$.

Path p cost, $c(p) = \sum_{(x,y) \in p} c_{xy}$.

The K shortest loopless paths problem

Given $K \in \mathbb{N}$, it is intended to determine $\mathcal{P}_K = \{p_1, \ldots, p_K\}$, where:

- $p_k \in \mathcal{P}$ is loopless, for any $k \in \{1, \ldots, K\}$;
- $c(p_k) \leq c(p_{k+1})$, for any $k \in \{1, \ldots, K-1\}$;
- $c(p_K) \leq c(p)$, for any loopless path $p \in \mathcal{P} - \mathcal{P}_K$;
- p_k is determined before p_{k+1}, for any $k \in \{1, \ldots, K-1\}$.

Tree of K loopless paths

Lemma 1 *Given $k \in \mathbb{N}$, k paths between a pair of nodes in a network form a "pseudo"-tree.*

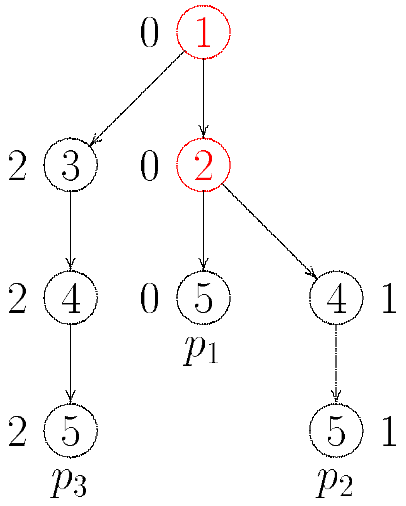
The deviation node of path p_k is $d(p_k) \in p_k$, the node such that $\mathrm{sub}_{p_k}(s, d(p_k))$ is the path with maximum number of nodes, contained in every $p_1, \ldots, p_{k-1}$.

By assumption, $d(p_1) = s$.

Given paths from $s = 1$ to $t = 5$:

$$p_1 = \langle 1, 2, 5 \rangle, p_2 = \langle 1, 2, 4, 5 \rangle \text{ and } p_3 = \langle 1, 3, 4, 5 \rangle .$$

Tree of paths (and tree of the 3 shortest loopless paths) p_1, p_2 and p_3.



$$d(p_1) = d(p_3) = s = 1$$

$$d(p_2) = 2$$

Fig. 1: Tree of paths

Deviation algorithms for ranking loopless paths

Construct the K shortest loopless paths tree.

$$\Downarrow$$

Determination of a "super"-tree of the K shortest loopless paths tree.

Initialization: $X = \{p_1\}$. /* X set of (loopless) paths, candidates */
For $k \in \{1, \ldots, K\}$: /* to next (loopless) paths. */
  p_k is the element with lower cost in X;
  p_k is picked out from X;
  p_k is analysed:
    New (loopless) paths are generated, candidates to p_j, $j > k$;
    X is updated with those (loopless) paths.

Generated candidates are deviations from p_k, that is, with the form:

$$p = \ \mathrm{sub}_{p_k}(s, v_i^k) \diamond q \ ,$$

where the first arc of q is not the first one in $\mathrm{sub}_{p_k}(v_i^k, t)$.

To avoid recalculation of paths:

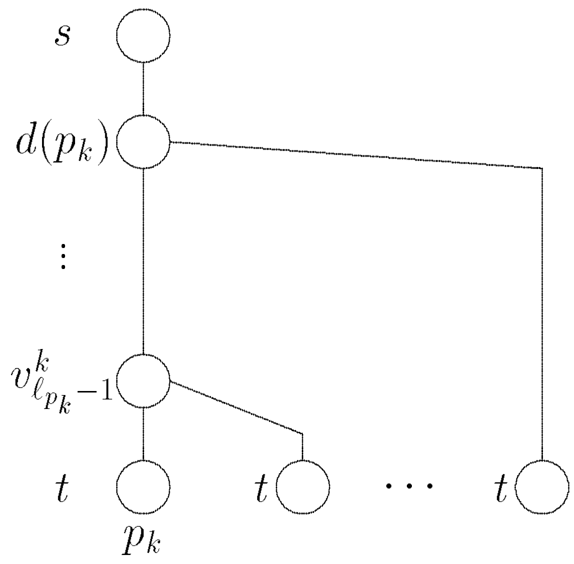
- Only nodes in $\mathrm{sub}_{p_k}(d(p_k), t)$ are analysed;
- When analysing $d(p_k)$, arcs $(d(p_k), j)$ belonging to (loopless) paths already determined should not be used.

Yen's ranking loopless paths algorithm

Deviation algorithm where only loopless paths are determined.

For every v_i^k to analyse:

- p is the shortest loopless path deviating from p_k in v_i^k.
- p_k is the parent of p.
- v_i^k is the deviation node of p.



To compute only loopless paths:

- Nodes of $\mathrm{sub}_{p_k}(s, v_{i-1}^k)$ cannot be repeated.

$$\Downarrow$$

Nodes in $\mathrm{sub}_{p_k}(s, v_{i-1}^k)$ are temporarily removed from $(\mathcal{N}, \mathcal{A})$.

$$+$$

The shortest loopless path from v_i^k to t in the modified network is determined.

Usual implementation of Yen's algorithm

Straightforward implementation of Yen's algorithm.

Analyse every node v_i^k of p_k from $d(p_k)$ to $v^k_{\ell_{p_k}-1}$.

For each analysed node v_i^k a shortest loopless path p is computed.

Loopless path p is such that:

- $p = \mathrm{sub}_{p_k}(s, v_i^k) \diamond q$,
- $q \in \mathcal{P}_{v_i^k t}$ is such that:
  - its first arc is not (v_i^k, v_{i+1}^k);
  - it has not been computed yet.

To determine p:

- $(\mathcal{N}, \mathcal{A})$ is modified by deleting:
  - Nodes in $\mathrm{sub}_{p_k}(s, v_{i-1}^k)$;
  - Arc (v_i^k, v_{i+1}^k);
  - Arcs (v_i^k, j), deleted when p_k was computed.
- q is the shortest path from v_i^k to t in the resulting network.

A new implementation of Yen's algorithm

Shortest tree rooted at t: $\mathcal{T}_t$.

Loopless path from $i \in \mathcal{N}$ to t in $\mathcal{T}$: $\mathcal{T}_t(i)$ and its cost: π_i.

Analyse every node v_i^k of p_k from $v^k_{\ell_{p_k}-1}$ to $d(p_k)$.

$$\Downarrow$$

Replace solving a shortest path problem, by:

1. Determining $\mathcal{T}_t$ in a "small" network.
2. Correcting $\mathcal{T}_t$ when a node or an arc is re-inserted in $(\mathcal{N}, \mathcal{A})$.

Remove $\mathrm{sub}_{p_k}(s, v^k_{\ell_{p_k}-1})$ from $(\mathcal{N}, \mathcal{A})$.
Remove $(d(p_k), j)$, $j \in \mathcal{N}$, of $p_1, \ldots, p_{k-1}$ from $(\mathcal{N}, \mathcal{A})$.
Determine $\mathcal{T}_t$.
For $(v_i^k \in \{v^k_{\ell_{p_k}-1}, \ldots, d(p_k)\})$:
    Restore v_i^k and (v_i^k, j), for $j \neq v^k_{i+1}$ in $(\mathcal{N}, \mathcal{A})$ /* Correct $\mathcal{T}_t$ */
    Add $p = \mathrm{sub}_{p_k}(s, v_i^k) \diamond \mathcal{T}_t(v_i^k)$ to X
    Restore (v_i^k, v^k_{i+1}) in $(\mathcal{N}, \mathcal{A})$ /* Correct $\mathcal{T}_t$ */

Re-inserting (v_i^k, v_{i+1}^k) in $(\mathcal{N}, \mathcal{A})$.

Some nodes' label might be improved.

Analyse $j \in \mathcal{N}$, from which there's at least one path until v_i^k.
/* $(\mathcal{N}, \mathcal{A})$ in the reverse star form. */

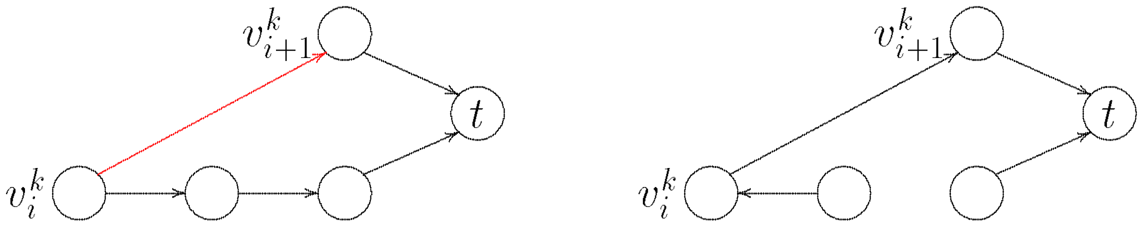


Fig. 2 (a): Inserting arc (v_i^k, v_{i+1}^k) in $\mathcal{T}_t$

Re-inserting v_i^k in $(\mathcal{N}, \mathcal{A})$.

Label v_i^k analysing $(v_i^k, j) \in \mathcal{A}$.
/* $(\mathcal{N}, \mathcal{A})$ in the forward star form. */

Other labels might be improved.
/* $(\mathcal{N}, \mathcal{A})$ in the reverse star form. */

If $\pi_{v_i^k} < \infty$, then $\mathcal{T}_t(v_i^k)$ is defined and $p = \mathrm{sub}_k(s, v_i^k) \diamond \mathcal{T}_t(v_i^k)$.

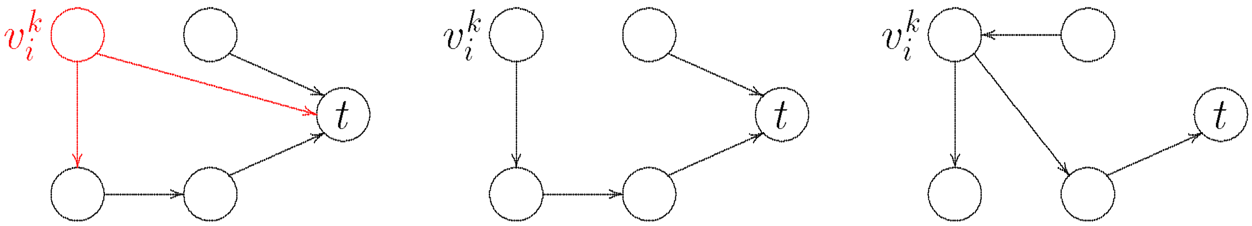


Fig. 2 (b): Inserting node v_i^k in $\mathcal{T}_t$

Algorithm 1 – *New implementation of Yen's algorithm*

```
Determine the shortest (loopless) path p from s to t in (N, A);
X ⟵ {p};
k ⟵ 0;
While (X ≠ ∅ and k < K) Do
    k ⟵ k + 1;
    Shortest loopless path p_k is picked out from X;

    Remove sub_{p_k}(s, v^k_{ℓ_{p_k}−1}) from (N, A);
    Remove arcs (d(p_k), i), i ∈ N, of p_1, ..., p_{k−1} from (N, A);

    T_t ⟵ shortest tree rooted at t in (N, A);

    For (v^k_i ∈ {v^k_{ℓ_{p_k}−2}, ..., d(p_k)}) Do
        Restore v^k_i in (N, A);
        Calculate π_{v^k_i} using forward star form;    /*  Label of v^k_i  */
        If (π_{v^k_i} is defined) Then
            Correct labels of v^k_i's successors using reverse star form;
            Add sub_{p_k}(s, v^k_i) ◇ T_t(v^k_i) to X;
        EndIf
        Restore (v^k_i, v^k_{i+1}) in (N, A);
        If (π^k_{v_i} > π_{v^k_{i+1}} + c_{v^k_i v^k_{i+1}}) Then
            π^k_{v_i} ⟵ π_{v^k_{i+1}} + c_{v^k_i v^k_{i+1}};
            Correct labels of v^k_i's successors using reverse star form;
        EndIf
    EndFor

    Restore sub_{p_k}(s, v^k_{ℓ_{p_k}−1}) in (N, A);
    Restore arcs (d(p_k), i), i ∈ N, of p_1, ..., p_{k−1} in (N, A);
EndWhile
```

Procedure 1.1 – *Calculate* π_i *using forward star form*

```
For ((i, j) ∈ A) Do
  If (π_i > π_j + c_ij) Then π_i ⟵ π_j + c_ij;
EndFor
```

Procedure 1.2 – *Correct labels of* x *successors using reverse star form*

```
list ⟵ {x};
Repeat
  i ⟵ element of list; list ⟵ list − {i};
  For ((j, i) ∈ A) Do
    If (π_j > π_i + c_ji) Then
      π_j ⟵ π_i + c_ji;
      If (j ∉ list) Then list ⟵ list ∪ {j};
    EndIf
  EndFor
Until (list = ∅);
```

Example

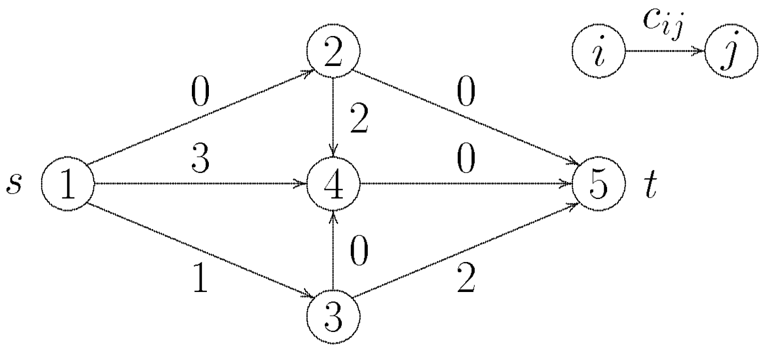


Fig. 4: Network $(\mathcal{N}, \mathcal{A})$

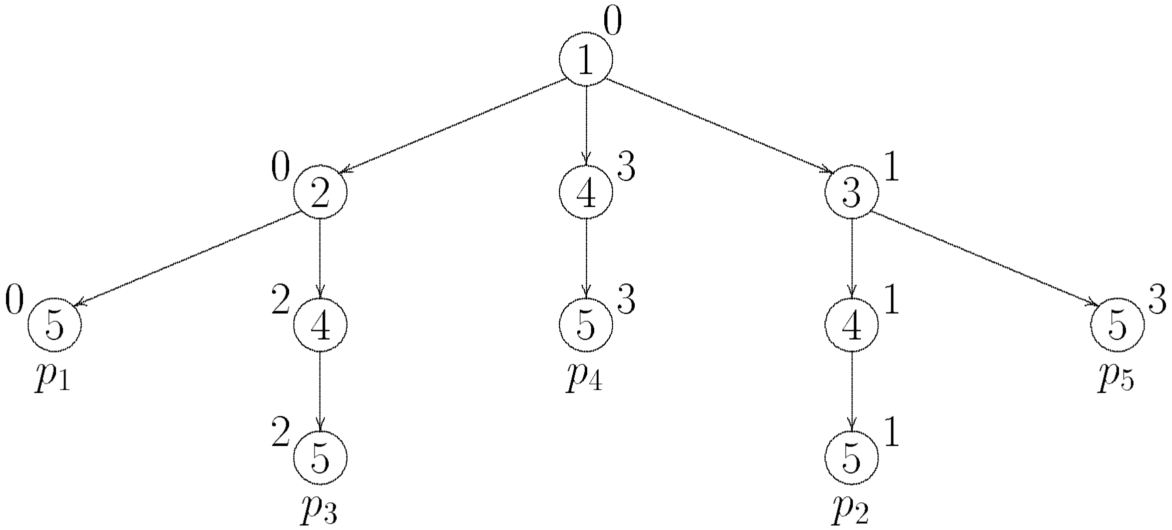


Fig. 5: Tree of the loopless paths from 1 to 5 in $(\mathcal{N}, \mathcal{A})$

$$X = \{\langle 1,2,5\rangle\} \quad p_1 = \langle 1,2,5\rangle$$

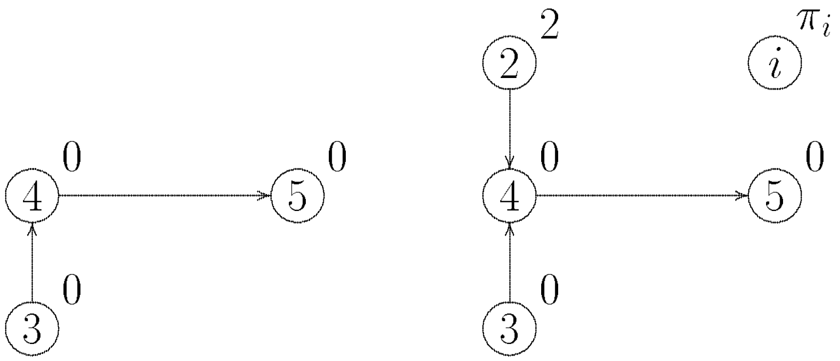


Fig. 6 (a): New implementation of Yen's algorithm – Analysing p_1

$$X = \{\langle 1,2,4,5\rangle\}$$

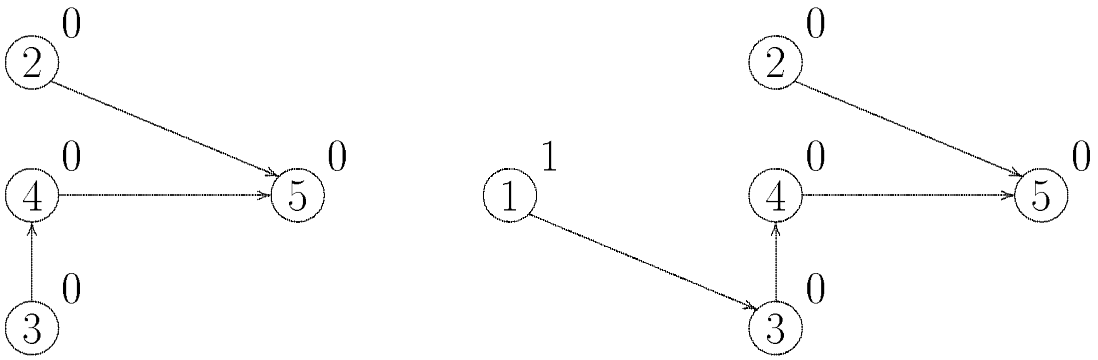


Fig. 6 (b): New implementation of Yen's algorithm – Analysing p_1

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

$$X = \{\langle 1, 2, 4, 5 \rangle\} \quad p_2 = \langle 1, 3, 4, 5 \rangle$$

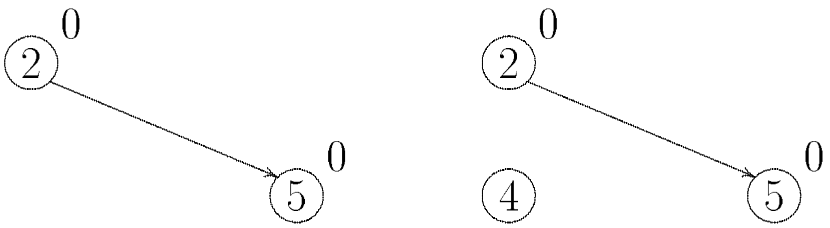


Fig. 7 (a): New implementation of Yen's algorithm – Analysing p_2

$$X = \{\langle 1, 2, 4, 5 \rangle\}$$

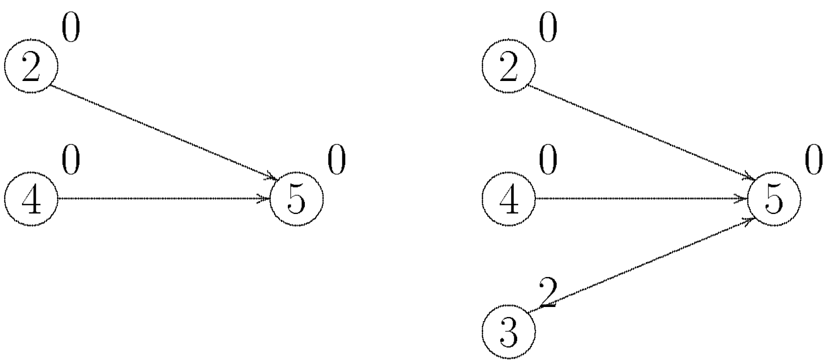


Fig. 7 (b): New implementation of Yen's algorithm – Analysing p_2

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

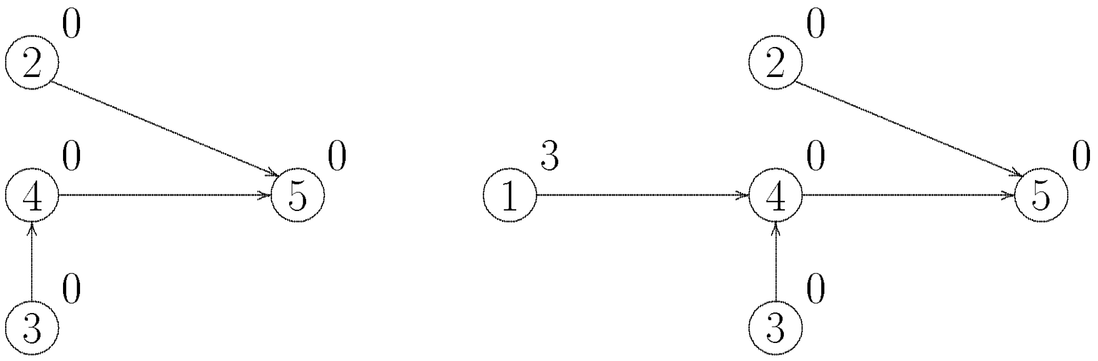


Fig. 7 (c): New implementation of Yen's algorithm – Analysing p_2

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

Theoretical complexity order (worst case analysis)

Usual implementation

Representing $(\mathcal{N}, \mathcal{A})$: Forward star form.

Complexity order:

> Each analysed node implies solving one shortest path problem:
> $\mathcal{O}(m + n \log n)$ (Dijkstra's algorithm).
> If each $p_1, \ldots, p_K$ demands the analysis of n nodes:
> $$\mathcal{O}(Kn(m + n \log n))$$

Deletion of nodes and arcs of p_k	$\leftrightarrow$	Re-insertion of nodes and arcs of p_k
Resolution of a shortest loopless path problem	$\leftrightarrow$	Labeling and correcting labels of some nodes

Proposed implementation

Representing $(\mathcal{N}, \mathcal{A})$: Forward star form and Reverse star form.

Complexity order:

> All nodes are relabeled, thus the operations performed are also
> $$\mathcal{O}(Kn(m + n \log n))$$

Computational experiments

Straightforward implementation (`SY`) versus new implementation (`NY`), coded in C.

Celeron 533 MHz, with 192 Mb of RAM, over Suse Linux 6.1.

Average times for solving 15 problems.

Networks	n	$d = m/n$
Random	5000, 10000	1.5, 2, 15, 20
Complete	100, 500	

Costs randomly calculated, uniformly generated in $[0, 1000]$.

	$d = 1.5$	$d = 2$	$d = 15$	$d = 20$
SY	0.932	1.147	4.181	5.055
NY	0.150	0.200	1.175	1.560
SY/NY	6.213	5.735	3.558	3.241

$K = 10$

Tab. 1 (a): Running times in seconds for random networks, $n = 5000$

	$d = 1.5$	$d = 2$	$d = 15$	$d = 20$
SY	106.425	131.763	545.447	631.419
NY	16.593	23.804	153.391	211.049
SY/NY	6.414	5.535	3.556	2.992

$K = 1000$

Tab. 1 (b): Running times in seconds for random networks, $n = 5000$

	$d = 1.5$	$d = 2$	$d = 15$	$d = 20$
SY	4.937	4.191	14.125	16.967
NY	0.452	0.610	3.037	3.873
SY/NY	10.923	6.871	4.650	4.381

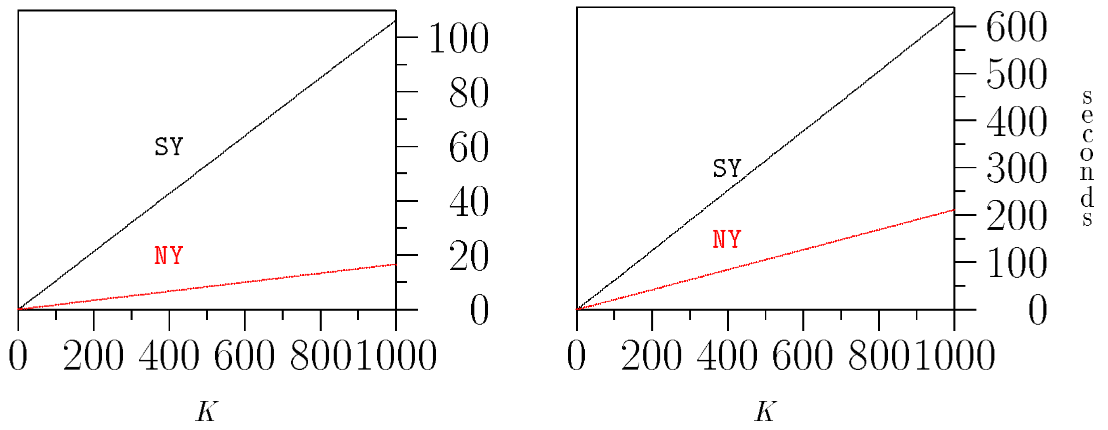
$K = 10$

Tab. 2 (a): Running times in seconds for random networks, $n = 10000$

	$d = 1.5$	$d = 2$	$d = 15$	$d = 20$
SY	476.909	463.924	670.430	557.182
NY	50.386	67.600	402.846	497.570
SY/NY	9.465	6.863	1.664	1.198

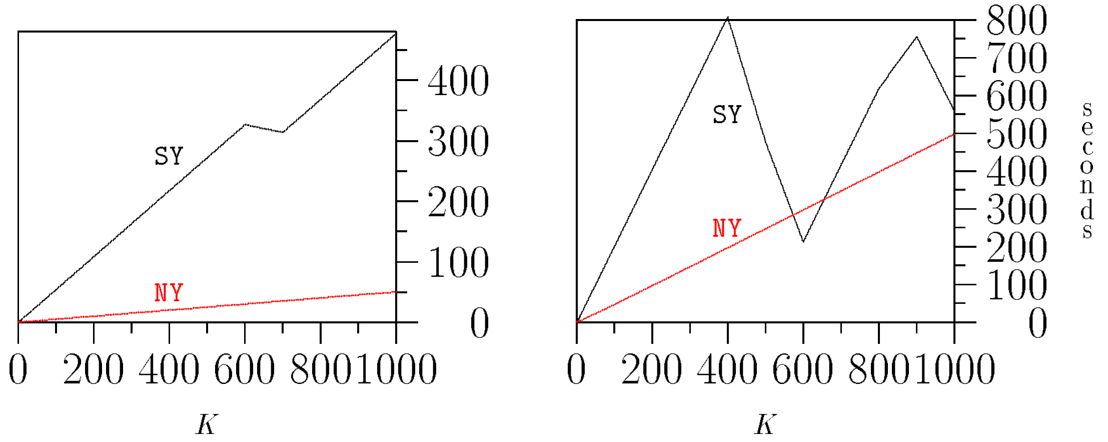
$K = 1000$

Tab. 2 (b): Running times in seconds for random networks, $n = 10000$



(a) $m = 7500$ (b) $m = 100000$

Fig. 8: Random networks, $n = 5000$



(a) $m = 15000$ (b) $m = 200000$

Fig. 9: Random networks, $n = 10000$

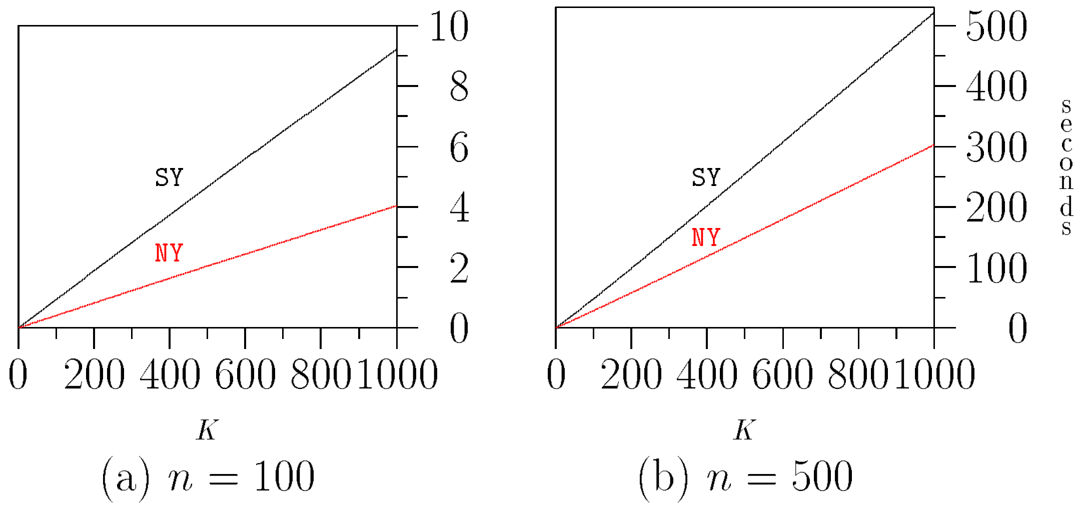


Fig. 10: Complete networks

References

- R. Dial, G. Glover, D. Karney, and D. Klingman. A computational analysis of alternative algorithms and labelling techniques for finding shortest path trees. *Networks*, 9:215–348, 1979.
- E. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, 1:395–412, 1959.
- S. E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17:395–412, 1969.
- D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28:652–673, 1998.
- E. Hadjiconstantinou and N. Christofides. An efficient implementation of an algorithm for finding k shortest simple paths. *Networks*, 34(2):88–101, 1999.
- R. Hoffman and R. R. Pavley. A method for the solution of the n th best path problem. *Journal of the Association for Computing Machinery*, 6:506–514, 1959.
- V. M. Jiménez and A. M. Varó. Computing the k shortest paths: a new algorithm and an experimental comparison. In *Proc. 3rd Workshop Algorithm Engineering*, July 1999.
- N. Katoh, T. Ibaraki, and H. Mine. An efficient algorithm for k shortest simple paths. *Networks*, 12:411–427, 1982.
- E. L. Lawler. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. *Management Science*, 18:401–405, 1972.

- E. Q. V. Martins. An algorithm for ranking paths that may contain cycles. *European Journal of Operational Research*, 18:123–130, 1984.
- E. Q. V. Martins, M. M. B. Pascoal e J. L. E. Santos. Deviation algorithms for ranking shortest paths. *The International Journal of Foundations of Computer Science*, 10:(3):247–263, 1999.
- E. Q. V. Martins, M. M. B. Pascoal, and J. L. E. Santos. An algorithm for ranking loopless paths. *CISUC*, Technical Report 99/007, 1999.
- E. Q. V. Martins, M. M. B. Pascoal, and J. L. E. Santos. A new improvement for a k shortest paths algorithm. To appear in *Investigação Operacional*, 1999.
- E. Q. V. Martins and J. L. E. Santos. A new shortest paths ranking algorithm. *Investigação Operacional*, 20:(1):47–62, 2000.
- A. Perko. Implementation of algorithms for k shortest loopless paths. *Networks*, 16:149–160, 1986.
- D. Shier. Interactive methods for determining the k shortest paths in a network. *Networks*, 6:151–159, 1976.
- J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17:712–716, 1971.
- J. Y. Yen. Shortest path network problems. *Mathematical Systems in Economics, Heft 18*, Hain, (1975), Meisennheim am Glan.