THE K SHORTEST LOOPLESS PATHS PROBLEM

Ernesto de Queirós Vieira Martins
Marta Margarida Braz Pascoal
José Luis Esteves dos Santos

{*eqvm,marta,zeluis*}*@mat.uc.pt*
Departamento de Matemática
Universidade de Coimbra
PORTUGAL

July 1998

Abstract: In this note we are concerned with the ranking of shortest loopless paths problem. It is shown that in general this problem does not satisfy the optimality principle and as a consequence only methods based on the computation of a super set of the set of the K shortest loopless paths are valid. It is shown how this methods for the unconstrained problem can be adapted for determining the K shortest loopless paths.

Keywords: network, path, loopless path, path distance, paths ranking, optimality principle.

1 Introduction

In a previous paper, [9], the unconstrained K shortest paths problem was studied. In the first part of this paper, the problem was viewed as a generalization of the classical shortest path problem, being the Optimality Principle and consequent *Labeling Algorithms* also generalized for $K > 1$. In the second part of the paper, the optimal set of the K shortest paths was determined with the use of algorithms that compute a pseudo tree which contains all the K shortest paths that are intended to be computed.

In this paper we intend to study the K shortest loopless paths problem. It is shown that the Optimality Principle is no more satisfied in general. However, the algorithms that compute the pseudo tree of the K shortest paths can be adapted to determine the K shortest loopless paths. An example of this adaptation is the algorithm proposed by Yen, [15], which was generalized for computing the K shortest paths – paths possibly with loops, [9, 11].

Notation and definitions have been introduced in the previous paper, [9].

2 The Optimality Principle

The Optimality Principle supports the well known labeling algorithms and asserts that there is a shortest path which is formed by shortest subpaths. As it is established in theorem 1 and in opposition with what happens in the K shortest paths problem, this principle can not be generalized for $K > 1$ when paths are required to be loopless.

Theorem 1 – *It may exist a* $k^{\underline{\text{th}}}$ *shortest loopless path containing a* $j^{\underline{\text{th}}}$ *loopless subpath with* $j > k$*.*

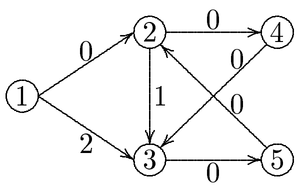

Figure 1: Network where the OP fails for the K shortest loopless paths problem

Proof – To prove theorem 1 we will use the example depicted from figure 1.

From $s = 1$ to $t = 2$ there are only two loopless paths in the network of figure 1: $p_1 = \langle 1, (1,2), 2\rangle$ and $p_2 = \langle 1, (1,3), 3, (3,5), 5, (5,2), 2\rangle$, with $c(p_1) = 0$ and $c(p_2) = 2$; that is, $c(p_1) < c(p_2)$. From $s = 1$ to node 3 only the following three loopless paths can be computed: $q_1 = \langle 1, (1,2), 2, (2,4), 4, (4,3), 3\rangle$, $q_2 = \langle 1, (1,2), 2, (2,3), 3\rangle$ and $q_3 = \langle 1, (1,3), 3\rangle$, with $c(q_1) = 0$, $c(q_2) = 1$ and $c(q_3) = 2$. So, q_3 is the third shortest loopless path from $s = 1$ to $t = 3$ and q_3 is a subpath of p_2, the second shortest loopless path from $s = 1$ to $t = 2$. ∎

From theorem 1 it can be concluded that the Optimality Principle does not hold in general for the ranking of loopless paths problem; as a consequence, labeling algorithms can not be generalized for $K > 1$ when only loopless paths have to be computed. In fact, since label setting algorithms determine the set $\mathcal{P}^K$ of the K shortest loopless paths during the execution of the algorithm, [9], they can lead to wrong solutions. Label correcting algorithms determine $\mathcal{P}^K$ only at the end of the algorithm, [9]; so, it would be necessary to determine the entire set of loopless paths in order to determine $\mathcal{P}^K$. That is, label correcting algorithms would be *brute force methods* with no practical utility.

From theorem 1 it results also that all the algorithms supported by the Optimality Principle, such as Dreyfus' algorithm, [4], all labeling algorithms due to Shier, [12, 13, 14], and all the versions of the path deletion algorithm of Martins *et al.*, [1, 2, 3, 7, 10], can not be adapted for determining only loopless paths.

3 The tree of the K shortest loopless paths

Such as in the unconstrained ranking path problem, the K shortest loopless paths form a pseudo-tree – the tree of the K shortest loopless paths. In this pseudo-tree all nodes can be repeated, except s – the initial node of each one of the K paths; despite, all the nodes in the tree will be considered as being different, [9].

Obviously, the goal of any ranking loopless paths algorithm is to compute the tree of the K shortest loopless paths. To achieve this, they usually determine a pseudo-tree containing the tree of the K shortest loopless paths. It is immediate that, the efficiency of the algorithm depends on the dimension of the computed pseudo-tree, that is, on its number of nodes.

The concepts of deviation node and deviation path of the $k^{\underline{\text{th}}}$ shortest loopless path are similar to the same concepts for the unconstrained problem.

Theorem 2 – *For $k \geq 1$, let $\mathcal{T}_k$ denote the tree of the k shortest loopless paths and let v_{k+1}, $p^{k+1}_{sv_{k+1}}$ and $p^{k+1}_{v_{k+1}t}$ denote the deviation node, the subpath of p_{k+1} from s to v_{k+1} and the deviation path of p_{k+1}, respectively. Then $p^{k+1}_{v_{k+1}t}$ is the shortest (loopless) path from v_{k+1} to t whose first arc is not an arc of $\mathcal{T}_k$ with v_{k+1} as tail node and such that $p^{k+1}_{sv_{k+1}} \diamond p^{k+1}_{v_{k+1}t}$ is a loopless path.*

Proof – Similar to the proof of theorem 5 in [9]. ∎

Such as in [9], theorem 2 is the support of a class of algorithm to de determine the K shortest loopless paths. The best known algorithm in this class is due to Yen, [15]; this algorithm will be briefly reviewed in the next paragraph.

3.1 Yen's algorithm

Like all the algorithms in this class, Yen's algorithm uses a set X of paths each of one candidate to the following loopless path. In the $k^{\underline{\text{th}}}$ step of the algorithm, the $k^{\underline{\text{th}}}$ shortest loopless path is picked from X – actually, it is the shortest path in X and it is a loopless path too – and some new candidates to the $(k+1)^{\underline{\text{th}}}$ shortest path are generated and added to X. Obviously, X is initialized with p_1, the shortest loopless path.

Let p_k be the $k^{\underline{\text{th}}}$ shortest loopless path, just picked from X; let v_k and p^k_{uv} denote the deviation node of p_k and the subpath of p_k from node u to node v, respectively. So, for each node v of path $p^k_{v_k t}$ the shortest loopless path from v to t, whose first arc is not an arc of $\mathcal{T}_k$ with v as tail node and such that $p^k_{sv} \diamond p^k_{vt}$ is a loopless path, will be determined. This is easily achieved by removing from the network all the nodes of p^k_{sv}, except node v, so as the arcs of $\mathcal{T}_k$ with v as tail node and determining the shortest path in the resulting network. Notice that the arcs whose tail node or/and head node had been removed, can be removed too.

The nodes of p^k_{sv} removal assures that the shortest path in the resulting network is a loopless path; the removal of arcs of $\mathcal{T}_k$ with v as tail node assures that the shortest path of the resulting network had never been determined before.

It must be noticed that for each k no more that n new candidate paths are generated. So, Yen's algorithm solves Kn shortest path problems at most. For undirected networks, Katoh, Ibaraki and Mine proposed a specialization of Yen's algorithm which allows the determination of, at most, three new candidate paths for each loopless path p_k, [6].

3.2 Adaptation for loopless paths of algorithms in a subclass

Some algorithms in the class of those that compute a super tree of the K shortest loopless paths can be grouped in a well defined subclass. This subclass comprises the algorithms adapted from those for the unconstrained problem for which $\mathcal{T}(t)$ – the tree of the shortest paths from $i \in \mathcal{N} - \{t\}$ to t – has to be computed. In fact, the usage of $\bar{c}_{ij}$ – the reduced cost associated with $\mathcal{T}(t)$ of the arc (i,j) – does not change the optimal solution and allows a significant improvement on the efficiency of the algorithms, [5, 9, 11]. In this subclass are included Eppstein's algorithm, [5], and algorithm MPS, [9, 11]. These algorithms can be adapted for determining the set of the K shortest loopless paths. Since the adaptation of all the algorithms is similar, it will only be briefly described the MPS adaptation.

The first step of MPS algorithm is to arrange the set $\mathcal{A}$ of arcs in the sorted forward star form, that is, in such a way that $\mathcal{A} = \{\bar{a}_1, \ldots, \bar{a}_m\}$ and $k < \ell$ if and only if $\bar{c}(\bar{a}_k) \leq \bar{c}(\bar{a}_\ell)$, where $\bar{c}(\bar{a}_i)$ stands for the reduced cost of arc $\bar{a}_i$. Moreover, $c(\bar{a}_i) = 0$ for all $\bar{a}_i \in \mathcal{T}(t)$, [5, 9]. Let us assume that p_k, the $k^{\underline{\text{th}}}$ shortest loopless path was just picked (and removed) from the set X of candidates to the $(k+1)^{\underline{\text{th}}}$ shortest loopless path. In general, some new candidates to p_{k+1} will be generated from p_k; to achieve this, for each node v of p^k_{vt} – the subpath of p_k from v to t – the shortest path $p^\star_{vt}$ from v to t, whose first arc is not an arc of $\mathcal{T}_k$ with v as tail node, will be computed if some path exists under this condition.

Let us assume that $\bar{a}_\ell$ is the p_k arc whose tail node is v. Let us assume too that $tail(\bar{a}_{\ell+1}) = v$ and $head(\bar{a}_{\ell+1}) = u$; that is, $\bar{a}_{\ell+1} = (v,u)$. Notice that there exists a path from v to t whose first arc is not an arc of $\mathcal{T}_k$ if and only if $tail(\bar{a}_{\ell+1}) = v$.

Once arcs are arranged in the sorted forward star form, $p^\star_{vt}$ is easily computed; in fact, if $p_{ut}(\mathcal{T}_t)$ stands for the path from u to t which is determined in $\mathcal{T}_t$, then $p^\star_{vt} = \langle v, (v,u), u \rangle \diamond p_{ut}(\mathcal{T}_t)$. This conclusion results from the fact of the reduced cost of path $p_{ut}(\mathcal{T}_t)$ being zero, [5, 9]. So, for each node v of $p^k_{v_k t}$, X is updated with $p^k_{sv} \diamond p^\star_{vt}$, if such a path exists.

However, the two loopless paths concatenation is not necessarily a loopless path. As a consequence, $p^\star_{vt} = \langle v, (v,u), u \rangle \diamond p_{ut}(\mathcal{T}_t)$ can be a path with loops and all the paths in the set X are not necessarily loopless. To reduce the probability of X containing some paths with loops, the arc $\bar{a}_{\ell+r}$ can be determined for each node v of $p^k_{v_k t}$, such as:

- $\bar{a}_\ell$ is the p_k arc whose tail node is v,
- $tail(\bar{a}_{\ell+r}) = v$, (and $head(\bar{a}_{\ell+r}) = u$)
- $p^k_{sv} \diamond \langle v, \bar{a}_{\ell+r}, u \rangle$ is a loopless path and $p^k_{sv} \diamond \langle v, \bar{a}_{\ell+i}, u \rangle$ is not a loopless path for $i \in \{1, \ldots, r-1\}$,

that is, $\bar{a}_{\ell+r}$ is the first arc following $\bar{a}_\ell$ in the sorted forward star form whose tail node is yet node v and its concatenation with p^k_{sv} is still a loopless path.

Furthermore, if a path p_k picked (and removed) from X is not a loopless path, for generating candidates to the $k^{\underline{\text{th}}}$ shortest loopless path (note that p_k is not loopless, so it is not the $k^{\underline{\text{th}}}$ shortest loopless path) the nodes v from its subpath $p^k_{v_k t}$ have to be analyzed while the two paths concatenation $p^k_{sv_k} \diamond p^k_{v_k v}$ is a loopless path, [8, 11].

Notice that an upper bound for the number of candidate paths in X can not be established. However, in practice the algorithm seems to perform well, [9, 11].

4 Conclusion

In this note the ranking of loopless paths problem is analized. It is shown that the Optimality Principle is not verified in general, the reason why labeling algorithms can not be used. However, algorithms in one different class can be adapted for computing loopless paths, being this adaptation described.

Some preliminary computational experiments, [8, 11], seem to indicate that the problem can be solved almost so easily as the unconstrained problem, in opposition with what has being usually believed.

Acknowledgments:
Research developed within **CISUC** ("Centro de Informática e Sistemas da Universidade de Coimbra") and partly supported by the portuguese Ministry of Science and Technology (MCT), under **PRAXIS XXI** Project of **JNICT** ("Junta Nacional de Investigação Científica e Tecnológica").

References

[1] J.A. Azevedo, M.E.O.S. Costa, J.J.E.R.S. Madeira, and E.Q.V. Martins. An algorithm for the ranking of shortest paths. *European Journal of Operational Research*, 69:97–106, 1993.

[2] J.A. Azevedo, J.J.E.R.S. Madeira, E.Q.V. Martins, and 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.

[3] J.A. Azevedo, J.J.E.R.S. Madeira, E.Q.V. Martins, and F.M.A. Pires. A computational improvement for a shortest paths ranking algorithm. *European Journal of Operational Research*, 73:188–191, 1994.

[4] S.E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17:395–412, 1969.

[5] D. Eppstein. Finding the k shortest paths. SIAM Journal on Computing (to appear).

[6] N. Katoh, T. Ibaraki, and H. Mine. An efficient algorithm for k shortest simple paths. *Networks*, 12:411–427, 1982.

[7] E.Q.V. Martins. An algorithm for ranking paths that may contain cycles. *European Journal of Operational Research*, 18:123–130, 1984.

[8] E.Q.V. Martins, M.M.B. Pascoal, and J.L.E. Santos. A new algorithm for ranking loopless paths. Submitted, 1997.

[9] E.Q.V. Martins, M.M.B. Pascoal, and J.L.E. Santos. The k shortest paths problem. Submitted, 1998.

[10] E.Q.V. Martins and J.L.E. Santos. A new shortest paths ranking algorithm. Submitted, 1996.

[11] M.M.B. Pascoal. Algoritmos para a enumeração dos k trajectos mais curtos, 1997. (Dissertação de Mestrado; Departamento de Matemática; Universidade de Coimbra).

[12] D. Shier. Computational experience with an algorithm for finding the k shortest paths in a network. *Journal of Research of the NBS*, 78:139–164, 1974.

[13] D. Shier. Interactive methods for determining the k shortest paths in a network. *Networks*, 6:151–159, 1976.

[14] D. Shier. On algorithms for finding the k shortest paths in a network. *Networks*, 9:195–214, 1979.

[15] J.Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17:712–716, 1971.