Ann Oper Res (2006) 147:5–21
DOI 10.1007/s10479-006-0068-x

A comprehensive survey on the quickest path problem

Marta M. B. Pascoal · M. Eugénia V. Captivo · João C. N. Clímaco

Published online: 22 August 2006

Abstract This work is a survey on a special minsum-maxmin bicriteria problem, known as the quickest path problem, that can model the transmission of data between two nodes of a network. Moreover, the authors review the problems of ranking the K quickest paths, and the K quickest loopless paths, and compare them in terms of the worst-case complexity order. The classification presented led to the proposal of a new variant of a known K quickest loopless paths algorithm. Finally, applications of quickest path algorithms are mentioned, as well as some comparative empirical results.

Keywords Bicriteria problem · Minsum-maxmin · Quickest path · Paths ranking

The quickest path problem is a particular case of optimal path problems, whose objective function evaluates the time taken to transmit a given amount of data through each path. This function depends on two parameters, an additive function that represents the path delay, and a bottleneck function that represents the path capacity or bandwidth. Thus, the resolution of

This work was partially supported by FEDER and OE, under the projects POCTI/MAT/139/2001, POCTI/ISFL-1/152, POSI/SRI/37346/2001, and POCTI/MAT/37707/2001.

M. M. B. Pascoal (✉)
Departamento de Matemática—Centro de Informática e Sistemas,
Apartado 3008, 3001-454 Coimbra, Portugal
e-mail: marta@mat.uc.pt

M. E. V. Captivo
Universidade de Lisboa, Faculdade de Ciências, Centro de Investigação Operacional, Bloco C6,
Campo Grande, 1749-016 Lisboa, Portugal
e-mail: mecaptivo@fc.ul.pt

J. C. N. Clímaco
Universidade de Coimbra, Faculdade de Economia, Avenida Dias da Silva, 165, 3004-512, Coimbra, Portugal

Instituto de Engenharia de Sistemas e Computadores—Coimbra, Rua Antero de Quental, 199,
3000-033 Coimbra, Portugal
e-mail: jclimaco@inescc.pt

Table 1 Worst-case complexities for quickest path algorithms in a network with n nodes, m arcs and r bandwidth values

	Time	Space
Quickest path algorithms		
Chen and Chin	$\mathcal{O}(r(m + n \log n))$	$\mathcal{O}(r(n + m))$
Rosen, Sun, and Xue	$\mathcal{O}(r(m + n \log n))$	$\mathcal{O}(n + m)$
Martins and Santos	$\mathcal{O}(r(m + n \log n))$	$\mathcal{O}(n + m)$
K-quickest path algorithms		
Chen (with Martins et al.)	$\mathcal{O}(m^2 + mn \log n + Kn + K^{3/2} \log K)$	$\mathcal{O}(Krn + m)$
K-quickest simple path algorithms		
Rosen, Sun, and Xue	$\mathcal{O}(Knr(m + n \log n))$	$\mathcal{O}(Kn + m)$
Chen (with Yen)	$\mathcal{O}(Knr(m + n \log n))$	$\mathcal{O}(Krn + m)$
Chen (with Katoh et al.)	$\mathcal{O}(Kr(m + n \log n))$	$\mathcal{O}(Krn + m)$
Pascoal, Captivo, and Clímaco	$\mathcal{O}(Kr(m + n \log n))$	$\mathcal{O}(Kn + m)$

a quickest path problem involves the minimisation of the first function and the maximisation of the second one, resulting in a minsum-maxmin problem. In the sequel some of the research papers on this subject are reviewed and compared. This paper also focuses on the determination of K paths and of K loopless paths according to the total transmission time. Potential applications are analysed and comparative studies of the algorithms summarised are referred to. Table 1 summarises the problems considered in this paper, the algorithms reviewed to solve them, besides the corresponding time and space bounds.

Section 1 is devoted to the introduction of the quickest path problem, whereas Section 2 addresses its generalisation, the ranking of K quickest paths and loopless paths. Section 2 also presents a new version of an algorithm for the K quickest loopless paths problem. Finally, Section 3 concerns the potential applications and the computational efficiency of quickest path algorithms.

1. The minsum-maxmin bicriteria problem

Let $(\mathcal{N}, \mathcal{A})$ be a network with n nodes and m arcs, and $s, t \in \mathcal{N}$ (with $s \neq t$) be the initial and terminal nodes of $(\mathcal{N}, \mathcal{A})$, respectively.

A path p from $i \in \mathcal{N}$ to $j \in \mathcal{N}$ in $(\mathcal{N}, \mathcal{A})$ is a sequence of the form $p = \langle i = v_1, v_2, \ldots, j = v_{\ell(p)} \rangle$, where $\ell(p)$ denotes the length of p, that is, its number of nodes, and $(v_k, v_{k+1}) \in \mathcal{A}$, for any $k \in \{1, \ldots, \ell(p) - 1\}$. Nodes i and j are called the initial and terminal nodes of path p, respectively. Let x and y be two nodes of p, then $\text{sub}_p(x, y)$ represents its subpath from x to y. A cycle (or loop) is a path with no repeated nodes, with the exception of the first one, which coincides with the last. Therefore, a path is said to be loopless (or simple) when it has no repeated nodes.

The set of paths (loopless paths) from i to j in $(\mathcal{N}, \mathcal{A})$ will be denoted by $\mathcal{P}_{ij}$ ($\bar{\mathcal{P}}_{ij}$), while $\mathcal{P}$ ($\bar{\mathcal{P}}$) will denote the set $\mathcal{P}_{st}$ ($\bar{\mathcal{P}}_{st}$). The concatenation of two paths, $p \in \mathcal{P}_{ij}$ and $q \in \mathcal{P}_{j\ell}$, is denoted by $p \diamond q$ and is the path from i to ℓ formed by path p followed by q. In the sequel each arc (i, j) will be associated with two values, $d_{ij} \in \mathbb{R}$ and $b_{ij} \in \mathbb{R}^+$.

One of the most well-known bicriteria problems is the minsum-maxmin problem, the purpose of which is to $\min_{p \in \mathcal{P}}\{d(p)\}$ and $\max_{p \in \mathcal{P}}\{b(p)\}$, with $d(p) = \sum_{(i,j) \in p} d_{ij}$ and $b(p) = \min_{(i,j) \in p}\{b_{ij}\}$, referred to in the classical paper by Hansen (1980). This problem considers that a path p dominates q iff $d(p) \leq d(q)$ and $b(p) \geq b(q)$, and at least one of the inequalities is strict. A path q is non-dominated if there exists no other path p dominating it. Hansen

proposes an algorithm that alternates the determination of shortest paths and the deletion of arcs with certain capacities, later adapted by Martins (1984) for another bicriteria problem, where one of the objective functions is of the maxmin type, and algorithms to compute the optimal path for the other function are known.

1.1. The quickest path problem

There are several applications of the minsum-maxmin problem and related problems as referred to by Hansen. One of those problems, first mentioned by Moore (1976), is related to the transmission of data or to road transportation, where each edge is simultaneously associated with a cost, delay or time value, $d_{ij} \in \mathbb{R}$, and an upper bound on the capacity or bandwidth of the arc, $b_{ij} \in \mathbb{R}^+$. If one considers the transmission of $\sigma \in \mathbb{R}^+$ data units of information (u.i.) from s to t, $T(p) = d(p) + \frac{\sigma}{b(p)}$ represents the total time for that transmission throughout $p \in \mathcal{P}$. The aim of the quickest path problem is to find a path p^*, such that $T(p^*)$ is minimum over $\mathcal{P}$. In the following d_{ij} will be called the delay of arc (i, j) and b_{ij} its bandwidth. Unlike the well-known shortest path problem, the quickest path problem does not satisfy the Optimality Principle. Therefore a labelling algorithm cannot be used to solve it. However, Martins and Santos (1997), and later Boffey et al. (2002), demonstrated that any optimal solution of the quickest path problem is a non-dominated solution of $(\min_{\mathcal{P}}\{d(p)\}, \max_{\mathcal{P}}\{b(p)\})$. Thus, although several algorithms for the quickest path problem are known in the literature, they are based on the computation of non-dominated paths for the above problem, followed by the selection of the one with the lowest total transmission time.

The algorithms described below are based on the transformation of the quickest path problem into the shortest path problem, by using three approaches: computation of the shortest path from s to any node in a new network, and computation of the shortest paths from s to t in a sequence of subnetworks of the original, without considering the pre-determined paths, or using this information.

Chen and Chin's algorithm. The first algorithm developed specifically for this problem was proposed by Chen and Chin (1990), who noted that $b(p) \in \{b_1, \ldots, b_r\}$ for any path $p \in \mathcal{P}$, where $b_1, \ldots, b_r$, with $r \leq m$, represent the distinct arc bandwidth values and are arranged so as to satisfy $b_1 \leq \cdots \leq b_r$. Furthermore, when the arcs bandwidth values are constant, the quickest path problem is reduced simply to the shortest path problem. Thus, Chen and Chin proposed enlarging the original network, by creating r levels, where each level corresponds to a subnetwork of the original, with a fixed bandwidth lower bound. The connection between levels is achieved by duplicating the arcs in $\mathcal{A}$, starting at a level whose bandwidth is higher than the arriving one. The new network will be denoted by $(\mathcal{N}', \mathcal{A}')$, and is defined as:

- $\mathcal{N}' = \{1^i, \ldots, n^i : 1 \leq i \leq r\}$,
- $\mathcal{A}' = \{(x^i, y^i) : (x, y) \in \mathcal{A} \wedge b_{xy} \geq b_i \wedge 1 \leq i \leq r\} \cup \{(x^i, y^j) : (x, y) \in \mathcal{A} \wedge b_{xy} = b_j \wedge 1 \leq j < i \leq r\}$.

The new initial node is s^r and $t^1, \ldots, t^r$ are all terminal nodes. In addition, the arcs' delay is maintained in the new network.

The paths of the augmented network, starting at s^r and ending at node t^i, correspond to the paths from s to t in $(\mathcal{N}, \mathcal{A})$, with a bandwidth of, at least, b_i, which leads to:

Theorem 1. *Let p_i be the shortest path (in terms of delay) from s^r to t^i in $(\mathcal{N}', \mathcal{A}')$, for any $i = 1, \ldots, r$. Then, the quickest path in $\mathcal{P}$ is $p^* \in \mathcal{P}$ such that $T(p^*) = \min_{1 \leq i \leq r}\{T(p_i)\}$.*

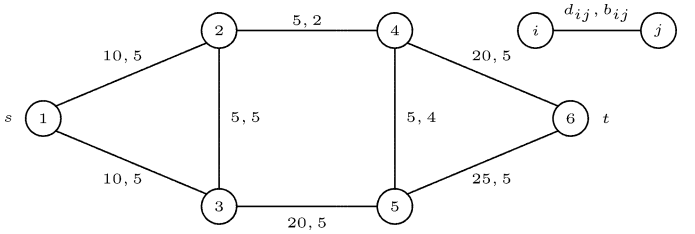

Fig. 1 Network $(\mathcal{N}, \mathcal{A})$, $\sigma = 100$

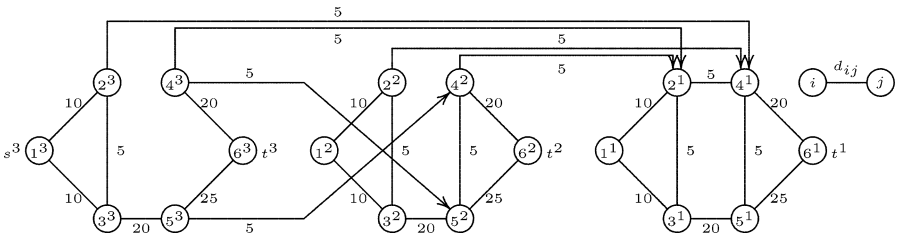

Fig. 2 Chen and Chin's network $(\mathcal{N}', \mathcal{A}')$

Chen and Chin's proposal consists of constructing $(\mathcal{N}', \mathcal{A}')$, computing the shortest path from s^r to any t^i, $i = 1, \ldots, r$, and choosing, from those paths, the one that corresponds to the quickest path from s to t in the original network, according to Theorem 1.

An illustrative example of the algorithm summarised is now presented. The intention is to compute the quickest path to transmit $\sigma = 100$ u.i. from 1 to 6 in the undirected network $(\mathcal{N}, \mathcal{A})$ represented in Fig. 1.

When applying Chen and Chin's algorithm, the augmented network of $(\mathcal{N}, \mathcal{A})$, which is depicted in Fig. 2, is constructed, and the tree of shortest paths from s^3 to any node in that network is computed. Then the authors compute the paths $\langle 1^3, 2^3, 4^1, 6^1\rangle$, $\langle 1^3, 3^3, 5^3, 4^2, 6^2\rangle$ and $\langle 1^3, 3^3, 5^3, 6^3\rangle$, respectively from s^3 to t^1, t^2 and t^3, with total transmission times of 85, 80 and 75 in the original network. The optimal solution is the one corresponding to $\langle 1, 3, 5, 6\rangle$.

Rosen et al.'s algorithm. After the publication of Chen and Chin's work, Rosen, Sun, and Xue (1991) presented another algorithm to determine the quickest path. It improved on the former insofar as it does not enlarge the original network. However, the main idea is still to compute several paths, with the best values in one of the objective functions of the bicriteria problem, while ensuring that the other function remains constant. This goal is achieved by considering networks with $b_1, \ldots, b_r$ as lower bounds of the arcs bandwidth and determining the shortest paths in these networks. The arcs with a given bandwidth are deleted, as new shortest paths are determined. Let $(\mathcal{N}, \mathcal{A}(w))$, with $w \geq 0$, be the subnetwork of $(\mathcal{N}, \mathcal{A})$ where $\mathcal{A}(w) = \{(i, j) \in \mathcal{A} : b_{ij} \geq w\}$.

Theorem 2. *Let p_i be the shortest path from s to t in $(\mathcal{N}, \mathcal{A}(b_i))$, $i = 1, \ldots, r$. Then, the quickest path in $\mathcal{P}$ is p^* such that $T(p^*) = \min_{1 \leq i \leq r}\{T(p_i)\}$.*

Rosen et al. have also pointed out that, if p_i is a shortest path in $(\mathcal{N}, \mathcal{A}(b_i))$ and $b(p_i) > b_{i'}$, then p_i is also a shortest path in $(\mathcal{N}, \mathcal{A}(b_{i'}))$, and therefore, the number of paths computed is

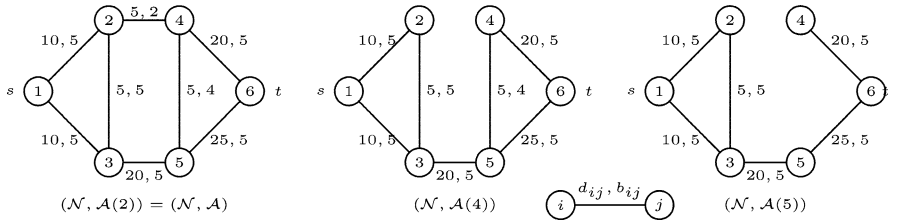

Fig. 3 Rosen et al.'s networks

smaller than r.[1] In using this approach one can alternate between finding shortest paths and removing the arcs with a bandwidth no greater than the bandwidth of the computed path. As before, the quickest path is the computed path registering the minimum total transmission time.

It should be noted that, as each $(\mathcal{N}, \mathcal{A}(b_i))$, $i = 1, \ldots, r$, is a subnetwork of $(\mathcal{N}, \mathcal{A})$ and $(\mathcal{N}, \mathcal{A}(b_i))$ is a subnetwork of $(\mathcal{N}, \mathcal{A}(b_{i-1}))$, the only modification between two consecutive networks is the deletion of some of the arcs, unlike the previous approach, by Chen and Chin, which enlarges the original network.

Using Rosen et al.'s algorithm to solve the quickest path problem in the previous example, the paths $q_1 = \langle 1, 2, 4, 6\rangle$ $(b(q_1) = 2, d(q_1) = 35)$, $q_2 = \langle 1, 3, 5, 4, 6\rangle$ $(b(q_2) = 4, d(q_2) = 55)$[2], and $q_3 = \langle 1, 3, 5, 6\rangle$ $(b(q_3) = 5, d(q_3) = 55)$, are computed in networks $(\mathcal{N}, \mathcal{A}(2))$, $(\mathcal{N}, \mathcal{A}(4))$ and $(\mathcal{N}, \mathcal{A}(5))$, respectively, as displayed in Fig. 3. Since q_3 records the best total transmission time of the three paths determined $(T(q_1) = 85, T(q_2) = 80, T(q_3) = 75)$, this path proves to be the solution to the problem.

Martins and Santos' algorithm. Martins and Santos (1997) explicitly interpreted the quickest path problem as the bicriteria problem seen above, and used the algorithm specifically for minsum-maxmin problems published earlier by Martins (1984). The result is very similar to the algorithm of Rosen et al.. However, Martins and Santos noted that in order to avoid determination of dominated solutions, in other words to solve fewer shortest path problems, the widest-shortest path (that is, the shortest path with a maximum bandwidth) should be found. This can be performed by adapting Dijkstra's algorithm, Dijkstra (1959), and labelling a node when its delay is improved (as in the original algorithm), or when there is a tie in the delay but the bandwidth is improved—see Martins et al. (1999a). As the set of arcs decreases along the sequence of networks $(\mathcal{N}, \mathcal{A}(b_1)), \ldots, (\mathcal{N}, \mathcal{A}(b_r))$, Martins and Santos' algorithm (as well as Rosen et al.'s) halts when no further paths exist to determine in one of these networks. As mentioned above, the use of the bicriteria variation of Dijkstra's algorithm enables one to omit some bandwidth levels that would lead to dominated paths.

Following up the previous example, Martins and Santos' algorithm determines the widest-shortest path in $(\mathcal{N}, \mathcal{A})$, and obtains $q_1 = \langle 1, 2, 4, 6\rangle$. Thus, after deleting the arcs with bandwidths at the most $b(q_1) = 2$ the widest-shortest path of the network is $q_2 = \langle 1, 3, 5, 6\rangle$ and, as $b(q_2) = 5$, no further paths are found. Therefore the quickest path from 1 to 6 is q_2, but only two paths were obtained (while, using Rosen et al.'s method three paths were computed).

[1] Since $d(p) < d(q)$ and $b(p) \geq b(q)$ implies that p dominates q, and therefore q is not a quickest path.

[2] Note, however, that we could also have $q_2 = \langle 1, 3, 5, 6\rangle$, since it has the same delay as path $\langle 1, 3, 5, 4, 6\rangle$.

Boffey et al.'s algorithm. As mentioned above, Boffey et al. have also related the quickest path problem and the bicriteria problem ($\min_{\mathcal{P}}\{d(p)\}$, $\max_{\mathcal{P}}\{b(p)\}$). On the basis of the procedure introduced by Rosen et al., Boffey et al. (2002) applied a method already used by Boffey (1996) for minsum-maxmin problems. Its application sets out to take advantage of the similarity between two successive Rosen et al. networks. Since the only difference between two of these successive networks is that the set of arcs of each is a subset of the set of arcs of the previous one, the proposal of Boffey et al. consists of replacing some resolutions of the shortest path problem by a simplified version of Dijkstra's algorithm. However, empirical tests performed by these authors show no substantial improvement on the algorithm's efficiency.

In terms of the number of operations undertaken, all the three algorithms, by Chen and Chin, Rosen et al. and Martins and Santos, are reported to have an $\mathcal{O}(r(m + n \log n))$ worst-case. This corresponds to computing the shortest path from s to any node in a network with rn nodes and rm arcs, or to solving r shortest path problems in a sequence of $(\mathcal{N}, \mathcal{A})$ subnetworks, using Dijkstra's algorithm. This worst-case occurs when r shortest paths have to be found. But it should be remembered that the number of paths computed can be smaller than r, when using Rosen et al.'s or Martins and Santos' methods. It should also be noted that the adaptation of Dijkstra's algorithm to find the widest-shortest path shares the same order of complexity as the usual algorithm.

Chen and Chin's algorithm proves to be worse in terms of space complexity, $\mathcal{O}(r(m + n))$, as they have to store the augmented network $(\mathcal{N}', \mathcal{A}')$, whereas in the two other algorithms, it is simply of $\mathcal{O}(m + n)$.

Albeit not included in this survey, several variants of the quickest path problem have also been studied. Among them we can mention Chen and Hung (1993), Lee and Papadopoulou (1993), who consider the quickest path among all node pairs, Calvete (2004), Lin (2003) and Rao (2004) who address non-deterministic values associated with the arcs, and Kagaris et al. (1999) who propose parallel algorithms to find the quickest path in sparse networks, besides algorithms to find the dynamic quickest path problem, where the parameters of the problem may change over time.

2. The K quickest path problems

One of the methods known for finding the non-dominated solutions of bicriteria path problems was presented in Clímaco and Martins (1981, 1982). Their algorithm ranks paths according to one of the objective functions, that is, it lists the best path, the second best path and so forth. Moreover, in analysing these paths, only the non-dominated ones are selected. This leads to the interest in ranking algorithms, which also have applications in other areas such as sensitivity analysis, dealing with the possible uncertainty of the problem, optimal path problems with additional constraints, or simply the generation of alternative solutions. Although most of the research about ranking problems has focused on the shortest paths, the ranking of solutions for other combinatorial problems, such as spanning trees, assignments or shortest paths in time-window networks, among others, has also been studied. Determination of a sequence of non-decreasing "cost" quickest paths is one of the subjects that has merited the attention of researchers. This problem is usually considered in the form of two versions: the general case, known as the K quickest path problem, where every path is allowed, and the constrained case, known as the K quickest loopless path problem, where only loopless paths can be determined. Its goal is to compute paths (or loopless paths) $p_1, \ldots, p_K$ between a pair of nodes, by non-decreasing order of the total transmission time, that is, such that $T(p_1) \leq \cdots \leq T(p_K) \leq T(p)$, for any $p \in \mathcal{P} - \{p_1, \ldots, p_K\}$ (or $p \in \bar{\mathcal{P}} - \{p_1, \ldots, p_K\}$, for loopless paths).

In the following, the k quickest path (loopless path) will be denoted $p_k = \langle v_1 = s, \ldots, v_{\ell(p_k)} = t\rangle$, for $k = 1, \ldots, K$.

2.1. Ranking K quickest paths

As far as we know the only published work concerning this specific problem is due to Chen (1993), and is based on the transformation of the K quickest path problem into the ranking of shortest paths in a special sequence of subnetworks of $(\mathcal{N}, \mathcal{A})$. Although some algorithms for ranking optimal paths can be adapted for the K quickest path problem, Chen's proposal is the one with the best theoretical worst-case behaviour.

Chen's algorithm. Similar to the method used in the quickest path problem, Chen's idea is to fix a bandwidth bound, while ranking shortest paths in a sequence of networks, and to keep one candidate to next k-th quickest path for each bandwidth value. The usage of Rosen et al.'s networks is not convenient, since the same path can be found in several networks.[3] Therefore, to prevent the determination of repeated paths, Chen considers the set of arcs arranged as $\mathcal{A} = \{a_1, \ldots, a_m\}$ such that $b_{a_1} \geq \cdots \geq b_{a_m}$, and the sequence of networks $\{(\mathcal{N}, \mathcal{A}(a_i))\}_{i=1}^{m}$, where $\mathcal{A}(a_i) = \{a_1, \ldots, a_i\}$, $i = 1, \ldots, m$. Furthermore, he determines a_i-constrained paths in each $(\mathcal{N}, \mathcal{A}(a_i))$, $i = 1, \ldots, m$, that is, paths from s to t with the form $q \diamond a_i \diamond q'$, where q and q' are paths from s to x and from y to t in $(\mathcal{N}, \mathcal{A}(a_i))$, and $a_i = (x, y)$.

As one may see, p_1 is the shortest a_i-constrained path in $(\mathcal{N}, \mathcal{A}(a_i))$, for some $i \in \{1, \ldots, m\}$, with the minimum total transmission time, and that, for $k \geq 1$, p_{k+1} is the j-th shortest a_i-constrained path in $(\mathcal{N}, \mathcal{A}(a_i))$, with $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, k+1\}$, thus differing from the previously determined paths $p_1, \ldots, p_k$, which registers a minimum total transmission time. Let a_i be the arc (x, y) and $p_{a_i}^k$ be the k-th shortest a_i-constrained path in $(\mathcal{N}, \mathcal{A}(a_i))$, $i \in \{1, \ldots, m\}$, and assume paths $p_{a_i}^1, \ldots, p_{a_i}^k$ have been computed. Chen has also proved that the next shortest a_i-constrained path is $p_{a_i}^{k+1} = p_{sx}^{j} \diamond a_i \diamond p_{yt}^{j'}$, with $j, j' \in \{1, \ldots, k+1\}$, where p_{uv}^{g} denotes the g-th shortest path from node u to node v in $(\mathcal{N}, \mathcal{A}(a_i))$. According to this result, if the k shortest paths from s to x and from y to t, are already known and stored by the algorithm, then the computation of $p_{a_i}^{k+1}$ can be summarised in the following two steps:

- find the next shortest path from s to x and the next shortest path from y to t in $(\mathcal{N}, \mathcal{A}(a_i))$,
- choose the quickest path from the paths with the form $p_{a_i}^{k+1} = p_{sx}^{j} \diamond a_i \diamond p_{yt}^{j'}$, $j, j' \in \{1, \ldots, k+1\}$.

Algorithm 1. *Chen's algorithm to rank K quickest paths*
Sort $\mathcal{A}$ by non increasing order of the arc bandwidths
`For` $(i \in \{1, \ldots, m\})$ `Do`
    $\mathcal{A}' \leftarrow \{a_1, \ldots, a_i\}$
    $P_i \leftarrow$ the shortest a_i-constrained path in $(\mathcal{N}, \mathcal{A}')$
`EndFor`
$k \leftarrow 0$
`While` $(k < K$ and $\{P_1, \ldots, P_m\} \neq \emptyset)$ `Do`
    $k \leftarrow k + 1$
    $p_k \leftarrow P_i$ such that $T(P_i) = \min_{1 \leq j \leq m}\{T(P_j)\}$
    $\mathcal{A}' \leftarrow \{a_1, \ldots, a_i\}$

[3] For instance, in the example, path $\langle 1, 3, 5, 6\rangle$ is the shortest path in, both, $(\mathcal{N}, \mathcal{A}(4))$ and $(\mathcal{N}, \mathcal{A}(5))$

$P_i \leftarrow$ the next shortest a_i-constrained path in $(\mathcal{N}, \mathcal{A}')$
`EndWhile`

Now consider the ranking of $K = 3$ quickest paths from 1 to 6 in the network $(\mathcal{N}, \mathcal{A})$ of Fig. 1 with Chen's algorithm. First assume that $\mathcal{A} = \{(1, 2), (1, 3), (2, 3), (3, 5), (4, 6), (5, 6), (4, 5), (2, 4)\}$. Then the shortest a_i-constrained paths found are $\langle 1, 3, 5\rangle \diamond \langle 5, 6\rangle$ in $(\mathcal{N}, \mathcal{A}(5, 6))$ (with a transmission time of 75), $\langle 1, 3, 5\rangle \diamond \langle 5, 4\rangle \diamond \langle 4, 6\rangle$ in $(\mathcal{N}, \mathcal{A}(4, 5))$ (with a transmission time of 80), and $\langle 1, 2\rangle \diamond \langle 2, 4\rangle \diamond \langle 4, 6\rangle$ in $(\mathcal{N}, \mathcal{A}(2, 4))$ (with a transmission time of 85). Thus, $p_1 = \langle 1, 3, 5, 6\rangle$, and is replaced by the 2-nd shortest (5, 6)-constrained path in $(\mathcal{N}, \mathcal{A}(5, 6))$, $\langle 1, 2, 3, 5\rangle \diamond \langle 5, 6\rangle$ (with a transmission time of 80), and the second quickest path will be $p_2 = \langle 1, 3, 5, 4, 6\rangle$ in $(\mathcal{N}, \mathcal{A}(4, 5))$. Then, path $\langle 1, 2, 3, 5\rangle \diamond \langle 5, 4\rangle \diamond \langle 4, 6\rangle$ (with a transmission time of 85) replaces p_2 in $(\mathcal{N}, \mathcal{A}(4, 5))$. Therefore $p_3 = \langle 1, 2, 3, 5, 6\rangle$ and the algorithm halts.

According to Chen, given an arc $a_i = (x, y)$ the next shortest a_i-constrained path, $p_{a_i}^{k+1}$, can be found by selecting the pair of paths $(p_{sx}^{j}, p_{yt}^{j'})$ with a minimum delay, such that $j \times j' \leq k + 1$, while ignoring previously obtained paths. Thus, this step takes $\mathcal{O}(\sqrt{k} + k \log k)$, for any $k \in \{1, \ldots, K\}$. The number of operations to find the next shortest path from s to x and from y to t depends on the ranking shortest paths algorithm employed. For instance, by using Martins et al.'s algorithm, Martins, Pascoal, and Santos (1999b), one may conclude that, in a worst-case scenario, the K quickest paths can be found by following Chen's algorithm with $\mathcal{O}(m^2 + mn \log n + Kn + K^{3/2} \log K)$ operations.

2.2. Ranking K quickest loopless paths

The algorithms for this problem can be classified in two groups, according to the methodology used. On the one hand we have adaptations of K shortest loopless path algorithms, while on the other hand we can consider a generalisation of the methods used to solve the quickest path problem. In the first group we describe the approaches by Rosen, Sun, and Xue (1991), which is an adaptation of the algorithm by Yen (1971), and Pascoal, Captivo, and Clímaco (2005), who, in turn, adapted the algorithm by Katoh, Ibaraki, and Mine (1982). In the second we review the algorithm by Chen (1994). Furthermore, we propose a new variant of Chen's algorithm, with the same worst-case theoretical complexity, but we anticipate that its behaviour will be better in practice.

Rosen et al.'s algorithm. The algorithm presented in Rosen, Sun, and Xue (1991) was the first one to compute K quickest loopless paths. It uses a set X where the loopless paths generated are stored, and each p_k is the quickest path selected from X, when $k - 1$ loopless paths have already been determined, $k \in \{1, \ldots, K\}$. The first path stored in X is the quickest one, and after p_1 is selected, other paths with a short transmission time are generated.

Let $p_k = \langle v_1 = s, \ldots, v_{\ell(p_k)} = t\rangle$ be the k quickest loopless path from s to t, $k = 1, \ldots, K$. Yen (1971) proposed a partition of the set of loopless paths in order to rank shortest loopless paths, such that

$$\begin{aligned} \bar{\mathcal{P}} - \{p_1\} &= \bigcup_{i=1}^{\ell(p_1)} \bar{\mathcal{P}}^1(v_i), \\ \bar{\mathcal{P}}^j\big(v_{d(p_k)}\big) - \{p_k\} &= \bigcup_{i=d(p_k)}^{\ell(p_k)} \bar{\mathcal{P}}^k(v_i), k > 1, \end{aligned} \tag{1}$$

where $\bar{\mathcal{P}}^j(v_i)$ is the set of the loopless paths, different from $p_1, \ldots, p_j$, that have $\text{sub}_{p_j}(s, v_i)$ as the initial subpath, common with path p_j, for some $1 \leq j < k$. When a p_k is picked up in X, $k \geq 1$, the set $\bar{\mathcal{P}}^j(v_{d(p_k)})$ where p_k was determined is considered, which means that it is partitioned by computing the shortest loopless path in each of the subsets in (1). Yen used this method for ranking shortest loopless paths, after noticing that the best deviation from p_k at node v_i is $\text{sub}_{p_k}(s, v_i) \diamond q_i$, where q_i is the shortest path from v_i to t, when the nodes $v_1, \ldots, v_{i-1}$ and the arcs $(v_i, x) \in \{p_1, \ldots, p_k\}$ are removed from the network. Then p_k is called the parent of the new paths determined (which are known as their sons or p_k deviations) and $v_{d(p_k)}$ the deviation node of p_k.

Partition (1) is independent of the objective function considered and can still be used, so long as we are able to find the best constrained path in $\bar{\mathcal{P}}^k(v_i)$. Rosen et al. proved the following result,

Theorem 3. *Let $p = \langle s = v_1, \ldots, v_\alpha \rangle$ be a loopless path from s to v_α in $(\mathcal{N}, \mathcal{A})$, and $(\mathcal{N}', \mathcal{A}')$ be a network where $\mathcal{N}' = \mathcal{N} - \{v_1, \ldots, v_{\alpha-1}\}$, and $\mathcal{A}' = \mathcal{A}|_{\mathcal{N}'}$. Let q_j be the shortest path from v_α to t in $(\mathcal{N}', \mathcal{A}'(b_j))$ and $p_j = p \diamond q_j$, $j = 1, \ldots, r$. Then*:

1. $T(p_j) = d(p) + d(q_j) + \frac{\sigma}{\min\{b(p), b(q_j)\}}$, $j = 1, \ldots, r$;
2. *the quickest path with p as the initial subpath is p^*, such that $T(p^*) = \min_{1 \leq j \leq r}\{T(p_j)\}$.*

Thus, the best path in $\bar{\mathcal{P}}^k(v_i)$ may be determined by resorting to the following steps:

- construct $(\mathcal{N}', \mathcal{A}')$ such that $\mathcal{N}' = \mathcal{N} - \{v_1, \ldots, v_{i-1}\}$ and $\mathcal{A}' = \mathcal{A}|_{\mathcal{N}'} - \{(v_i, v_{i+1})\} - \{(v_{d(p_k)}, x) \in \{p_1, \ldots, p_k\}\}$,
- apply Theorem 3 and compute the shortest path in $(\mathcal{N}', \mathcal{A}'(b_j))$, $j = 1, \ldots, r$,

then choose the loopless path $\text{sub}_{p_k}(s, v_i) \diamond q_j$ with the minimum total transmission time.

Algorithm 2. *Rosen et al.'s algorithm to rank K quickest loopless paths*

$p \leftarrow$ the quickest loopless path from s to t in $(\mathcal{N}, \mathcal{A})$; $v_{d(p)} \leftarrow s$
$X \leftarrow \{p\}$
$k \leftarrow 0$
`While` $(X \neq \emptyset$ and $k < K)$ `Do`
  $k \leftarrow k + 1$
  $p_k \leftarrow$ the quickest loopless path in X /* $p_k = \langle v_1, \ldots, v_{\ell(p_k)} \rangle$ */
  $X \leftarrow X - \{p_k\}$
  $\mathcal{A}' \leftarrow \mathcal{A} - \{(v_{d(p_k)}, x) \in p_1, \ldots, p_{k-1}\}$
  `For` $(v_i \in \{v_{d(p_k)}, \ldots, v_{\ell(p_k)}\})$ `Do`
    $\mathcal{A}' \leftarrow \mathcal{A}' - \{(v_i, v_{i+1})\}$
    $q^* \leftarrow$ the quickest path from s to t, with initial path $\text{sub}_{p_i}(s, v_i)$ in $(\mathcal{N}, \mathcal{A}')$
    $v_{d(q^*)} \leftarrow v_i$
    $X \leftarrow X \cup \{q^*\}$
  `EndFor`
`EndWhile`

Procedure 1. *The quickest loopless path from s to t, containing $p = \langle s = v_1, \ldots, v_\alpha \rangle$, in $(\mathcal{N}, \mathcal{A})$*

$(b_1, \ldots, b_r) \leftarrow$ arc bandwidth values by increasing order
$\mathcal{N}' \leftarrow \mathcal{N} - \{v_1, \ldots, v_{\alpha-1}\}$; $\mathcal{A}' \leftarrow \mathcal{A}|_{\mathcal{N}'}$
$L \leftarrow \emptyset$

$i \leftarrow 1$
`While` $(i \leq r)$ `Do`
  $\mathcal{A}' \leftarrow \{(x, y) \in \mathcal{A}' : b_{xy} \geq b_i\}$
  $q \leftarrow$ the shortest path from v_α to t in $(\mathcal{N}', \mathcal{A}')$
  $L \leftarrow L \cup \{q\}$
  $j \leftarrow k$ such that $b(q) = b_k$; $i \leftarrow j + 1$
`EndWhile`
$p^* \leftarrow$ the quickest path in $\{p \diamond q : q \in L\}$

Now consider the ranking of $K = 3$ quickest loopless paths from 1 to 6 in the network $(\mathcal{N}, \mathcal{A})$ of Fig. 1. Rosen et al.'s algorithm begins by determining the quickest path from 1 to 6 in the network $(\mathcal{N}, \mathcal{A})$ of Fig. 1, $p_1 = \langle 1, 3, 5, 6\rangle$ (with transmission time of 75). Its nodes are then analysed and the new loopless paths stored in the set of candidates $X = \{\langle 1, 2, 3, 5, 6\rangle, \langle 1, 3, 2, 4, 6\rangle, \langle 1, 3, 5, 4, 6\rangle\}$. Since loopless path $\langle 1, 2, 3, 5, 6\rangle$ is the quickest in X, it is p_2 (with a transmission time of 80), which is selected and analysed, generating $\langle 1, 2, 4, 6\rangle$ and $\langle 1, 2, 3, 5, 4, 6\rangle$. Set X is updated as $X = \{\langle 1, 3, 2, 4, 6\rangle, \langle 1, 3, 5, 4, 6\rangle, \langle 1, 2, 4, 6\rangle, \langle 1, 2, 3, 5, 4, 6\rangle\}$, where we can pick up $p_3 = \langle 1, 3, 5, 4, 6\rangle$ (with a transmission time of 80).

Pascoal et al.'s algorithm. Recently Pascoal, Captivo, and Clímaco (2005) have proposed an approach analogous to Rosen et al.'s algorithm, since they adapted Katoh et al.'s algorithm for ranking shortest loopless paths in undirected networks. As in Rosen et al.'s, a set X of candidates is used, where each p_k is selected, $k = 1, \ldots, K$. The difference between the two algorithms is the partition used to generate new candidates to store in X, as presented by Katoh, Ibaraki, and Mine (1982). Let p_j be the parent of some loopless path p_k and:

- v_δ be the deviation node of a son of p_j, previous to $v_{d(p_k)}$ and farther from s,
- v_γ be the deviation node of another son of p_j, closer to s but after $v_{d(p_k)}$.

If $\bar{\mathcal{P}}_k^j(v_\delta, v_\gamma)$ denotes the set of loopless paths of the form $q' = \text{sub}_{p_j}(s, v_\delta) \diamond q \notin \{p_1, \ldots, p_k\}$, where q is a path from v_δ to t that deviates from p_j before v_γ, then,

$$\begin{aligned} \bar{\mathcal{P}} - \{p_1\} &= \bar{\mathcal{P}}_2^1(s, t), \\ \bar{\mathcal{P}}_k^j(v_\delta, v_\gamma) - \{p_k\} &= \bar{\mathcal{P}}_{k+1}^j\big(v_\delta, v_{d(p_k)}\big) \cup \bar{\mathcal{P}}_{k+1}^j\big(v_{d(p_k)}, v_\gamma\big) \cup \bar{\mathcal{P}}_{k+1}^k\big(v_{d(p_k)+1}, t\big), k > 1. \end{aligned} \tag{2}$$

As before, this partition does not depend on the objective function considered. It can therefore be applied to the ranking of loopless paths according to the total transmission time. The analysis of each p_k consists of determining the quickest loopless path in each of the above subsets, namely:

- the best path in $\bar{\mathcal{P}}_{k+1}^j(v_\delta, v_{d(p_k)})$, i.e., which deviates from p_j between v_δ and $v_{d(p_k)}$,
- the best path in $\bar{\mathcal{P}}_{k+1}^j(v_{d(p_k)}, v_\gamma)$, i.e., which deviates from p_j between $v_{d(p_k)}$ and v_γ,
- and the best path in $\bar{\mathcal{P}}_{k+1}^k(v_{d(p_k)+1}, t)$, i.e., which deviates from p_k between $v_{d(p_k)+1}$ and t.

To find the quickest loopless path in some $\bar{\mathcal{P}}_k^j(v_x, v_y)$ Katoh et al. calculate the shortest path from v_x to t, that deviates from p_j before v_y, after deleting the nodes of $\text{sub}_{p_j}(s, v_{x-1})$ and the arcs in paths $p_1, \ldots, p_{j-1}$, that is, the second shortest loopless path in that network, and concatenate it with $\text{sub}_{p_j}(s, v_x)$. This method cannot be used in a straightforward manner,

since when $T(q') < T(q'')$ one may not be sure that $T(\text{sub}_{p_j}(s,i) \diamond q') < T(\text{sub}_{p_j}(s,i) \diamond q'')$, with $q', q'' \in \bar{\mathcal{P}}_{it}$. Thus the determination of the quickest path from v_x to t that deviates before v_y is not sufficient. However, Pascoal et al. proved a result analogous to Theorem 3, which reduces the problem of finding the quickest path in $\bar{\mathcal{P}}_k^j(v_x, v_y)$ to several (at most r) shortest path related problems.

Theorem 4. *Let v_x and v_y be two nodes of a loopless path p from s to t in $(\mathcal{N}, \mathcal{A})$, and let q_i be the shortest loopless path from s to t, deviating from p between v_x and v_y, in $(\mathcal{N}, \mathcal{A}(b_i))$, $i = 1, \ldots, r$. Then, the quickest loopless path from s to t, that deviates from p between v_x and v_y in $(\mathcal{N}, \mathcal{A})$ is q^* such that $T(q^*) = \min_{1 \le i \le r}\{T(q_i)\}$.*

Let $(\mathcal{N}', \mathcal{A}')$ be a network (similar to the one used by Rosen et al.), such that $\mathcal{N}' = \mathcal{N} - \{v_1, \ldots, v_{x-1}\}$ and $\mathcal{A}' = \mathcal{A}|_{\mathcal{N}'} - \{(v_x, v_{x+1})\} - \{(v_{d(p_z)}, v_l) \in \{p_1, \ldots, p_j\} \wedge d(p_j) \le d(p_z) \le x\}$. Then the shortest path from s to t that deviates from p_j between v_x and v_y in some $(\mathcal{N}, \mathcal{A}(b_i))$, with $i \in \{1, \ldots, r\}$, is $q_i = \text{sub}_{p_j}(s, v_x) \diamond q$, where q is the shortest path to deviate from $\text{sub}_{p_j}(v_x, t)$ before v_y in $(\mathcal{N}, \mathcal{A}(b_i))$. Still the remaining problem of finding such a path q is different from the one solved by Katoh et al., where the path $\text{sub}_{p_j}(v_x, t)$ is always the shortest path in $(\mathcal{N}', \mathcal{A}')$, which may not be so in this case.

Let $\mathcal{T}_s$ be the tree of shortest paths from s to any node, and $\mathcal{T}_t$ be the tree of shortest paths from any node to t. Let $\mathcal{T}_s(i)$ also be the path from s to $i \in \mathcal{N}$ in $\mathcal{T}_s$, and $\xi_s(i)$ be the index of the node where $\mathcal{T}_s(i)$ and $p^* = \mathcal{T}_s(t) = \mathcal{T}_t(s)$ split (analogously for $\mathcal{T}_t(i)$ and $\xi_t(i)$). It can be assumed, with no loss of generality, that $\xi_s(i) \le \xi_t(i)$, $i \in \mathcal{N}$—see Katoh, Ibaraki, and Mine (1982)—, and thus the shortest path q in $(\mathcal{N}, \mathcal{A}(b_i))$, that deviates from $p' = \langle s = v_1, v_2, \ldots, v_{\ell(p')} = t\rangle$ before $v_\alpha \in p'$, can be obtained according to the following scheme:

- if $p' \neq \mathcal{T}_s(t)$ then $q = \mathcal{T}_s(t) = \mathcal{T}_t(s)$,
- or else, $p' = \mathcal{T}_s(t) = \mathcal{T}_t(s)$ and Katoh et al.'s result can be applied, which states that q is the shortest path of type 1 or 2:

  1. $\mathcal{T}_s(i) \diamond \mathcal{T}_t(i)$, with $i \in \mathcal{N}$ such that $\xi_s(i) < \alpha$,
  2. $\mathcal{T}_s(i) \diamond \langle i, j\rangle \diamond \mathcal{T}_t(j)$, with $(i, j) \in \mathcal{A} - (\mathcal{T}_s \cup \mathcal{T}_t)$ and $\xi_s(i) < \alpha$.

Algorithm 3. *Pascoal et al.'s algorithm to rank K quickest loopless paths*

$p_1 \leftarrow$ the quickest path from s to t in $(\mathcal{N}, \mathcal{A})$
$p \leftarrow$ the quickest loopless path from s to t, that deviates from p_1 before t
$X \leftarrow \{p\}$
$k \leftarrow 1$
`While` $(X \neq \emptyset$ and $k < K)$ `Do`
  $k \leftarrow k + 1$
  $p_k \leftarrow$ the quickest loopless path in X /* $p_k = \langle v_1, \ldots, v_{\ell(p_k)}\rangle$ */
  $X \leftarrow X - \{p_k\}$
  $p_j \leftarrow$ loopless path parent of p_k
  $v_\delta \leftarrow$ deviation node of a son of p_j farther from s and preceding $v_{d(p_k)}$
  $v_\gamma \leftarrow$ deviation node of a son of p_j closer to s and following $v_{d(p_k)}$
/* `The quickest loopless path in` $\bar{\mathcal{P}}_{k+1}^j(v_\delta, v_{d(p_k)})$ */
  $\mathcal{A}' \leftarrow \mathcal{A} - \{(v_\delta, x) : (v_\delta, x) \in \{p_j, \ldots, p_{k-1}\}\}$
  $P_c \leftarrow$ the quickest loopless path, that deviates from p_j between v_δ and $v_{d(p_k)}$ in $(\mathcal{N}, \mathcal{A}')$
  $X \leftarrow X \cup \{P_c\}$

/* The quickest loopless path in $\bar{\mathcal{P}}^{j}_{k+1}(v_{d(p_k)}, v_\gamma)$ */
$\mathcal{A}' \leftarrow \mathcal{A}' - \{(v_{d(p_k)}, x) : (v_{d(p_k)}, x) \in \{p_j, \ldots, p_k\}\}$
$P_b \leftarrow$ the quickest loopless path, that deviates from p_j between $v_{d(p_k)}$ and v_γ in $(\mathcal{N}, \mathcal{A}')$
$X \leftarrow X \cup \{P_b\}$
/* The quickest loopless path in $\bar{\mathcal{P}}^{k}_{k+1}(v_{d(p_k)+1}, t)$ */
$P_a \leftarrow$ the quickest loopless path, that deviates from p_k between $v_{d(p_k)+1}$ and t in $(\mathcal{N}, \mathcal{A}')$
$X \leftarrow X \cup \{P_a\}$
EndWhile

Procedure 2. *The quickest loopless path that coincides with q up to v_α and deviates before v_β*
$(b_1, \ldots, b_r) \leftarrow$ arc bandwidth values by increasing order
$\mathcal{N}' \leftarrow \mathcal{N} - \{v_1, \ldots, v_{\alpha-1}\}$; $\mathcal{A}' \leftarrow \mathcal{A}|_{\mathcal{N}'}$
$L \leftarrow \emptyset$
$i \leftarrow 1$
While $(i \leq r)$ Do
$\mathcal{A}' \leftarrow \{(x, y) \in \mathcal{A}' : b_{xy} \geq b_i\}$
$p \leftarrow$ the shortest loopless path from v_α to t, that deviates from q before v_β in $(\mathcal{N}', \mathcal{A}')$
$L \leftarrow L \cup \{p\}$
$j \leftarrow k$ such that $b(p) = b_k$; $i \leftarrow j + 1$
EndWhile
$p^* \leftarrow$ the quickest loopless path in $\{q \diamond p : p \in L\}$

In the following procedure π^s_i denotes the delay of path $\mathcal{T}_s(i)$, for any $i \in \mathcal{N}$, and analogously for π^t_i.

Procedure 3. *The shortest loopless path that deviates from q before v_α in $(\mathcal{N}, \mathcal{A})$*
Compute $\mathcal{T}_s$
If (q is not defined or $q \neq \mathcal{T}_s(t)$) Then $p \leftarrow \mathcal{T}_s(t)$
Else
Compute $\mathcal{T}_t$
$d^* \leftarrow +\infty$; $L \leftarrow \{s\}$
While $(L \neq \emptyset)$ Do
$i \leftarrow$ element of L; $L \leftarrow L - \{i\}$
If $(\xi_s(i) = \xi_t(i))$ Then
For $((i, j) \in \mathcal{A} - \mathcal{T}_s - \mathcal{T}_t$ such that $\xi_s(i) < \xi_t(j))$ Do
If $(\pi^s_i + d_{ij} + \pi^t_j < d^*)$ Then
$d^* \leftarrow \pi^s_i + d_{ij} + \pi^t_j$; $i^* \leftarrow i$; $j^* \leftarrow j$
EndIf
For $((i, j) \in \mathcal{T}_s$ such that $\xi_s(j) < \alpha)$ Do $L \leftarrow L \cup \{j\}$
EndIf
If $(\xi_s(i) < \xi_t(i))$ Then
If $(\pi^s_i + \pi^t_i < d^*)$ Then
$d^* \leftarrow \pi^s_i + \pi^t_i$; $i^* \leftarrow i$
EndIf
For $((i, j) \in \mathcal{T}_s$ such that $\xi_s(j) < \alpha)$ Do $L \leftarrow L \cup \{j\}$
EndIf

```
    EndWhile
    If ((i*, j*) is defined) Then p ← T_s(i*) ◇ ⟨i*, j*⟩ ◇ T_t(j*)
    If (i* is defined) Then p ← T_s(i*) ◇ T_t(i*)
EndIf
```

Fewer loopless paths need be determined when applying Pascoal et al.'s algorithm to the $K = 3$ quickest loopless path problem in the network of Fig. 1. In fact, after computing $p_1 = \langle 1, 3, 5, 6\rangle$ (with $T(p_1) = 75$), $p_2 = \langle 1, 2, 3, 5, 6\rangle$ (with $T(p_2) = 80$) is the quickest path from 1 to 6 that deviates from p_1 between nodes 1 and 6. From p_2, loopless paths $\langle 1, 3, 5, 4, 6\rangle$, different from p_2 and that deviates from p_1 after 1, and $\langle 1, 2, 4, 6\rangle$, that deviates from p_2 after 1, are computed. Then $X = \{\langle 1, 3, 5, 4, 6\rangle, \langle 1, 2, 4, 6\rangle\}$ and $p_3 = \langle 1, 3, 5, 4, 6\rangle$ (with $T(p_3) = 80$), since that is the quickest path in that set.

Chen's algorithm. In 1994 Chen looks at the problem from a different perspective, analogous to the one used in the quickest path problem. In fact, Chen (1994) transforms the K quickest loopless path problem into that of finding K shortest loopless paths m times. However, with a minor modification his procedure may compute K shortest loopless paths only r times, based on the following result,

Theorem 5. *Let p_{jk} be the k-th shortest loopless path from s to t in $(\mathcal{N}, \mathcal{A}(b_j))$, for $k \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, r\}$. Then, p_i, the i-th quickest path from s to t in $(\mathcal{N}, \mathcal{A})$, is such that $p_i \in \{p_{jk} : 1 \leq k \leq i \wedge 1 \leq j \leq r\}$.*

Chen finds the K shortest loopless paths in every network of the sequence $(\mathcal{N}, \mathcal{A}(b_1)), \ldots, (\mathcal{N}, \mathcal{A}(b_r))$, and then selects the K quickest loopless paths from the Kr candidates obtained, at most, $p_{11}, \ldots, p_{1K}, \ldots, p_{r1}, \ldots, p_{rK}$, which are paths $p_1, \ldots, p_K$. Since a path can belong to several networks, it can be computed more than once. Therefore it is possible to store only the distinct loopless paths in the set of candidates which can be represented by a heap. It should be noted that any shortest loopless path ranking algorithm can be applied and this may modify the theoretical and computational behaviour.

The main difference between this algorithm and the other ranking quickest loopless path algorithms described here lies in the fact that, in this case, the paths computation and the selection of $p_1, \ldots, p_K$ is separated into two phases, while in the other algorithms, computation and selection alternate.

Algorithm 4. *Chen's algorithm to rank K quickest loopless paths*

```
(b_1, ..., b_r) ← arc bandwidth values by increasing order
A' ← A
X ← ∅
For (i ∈ {1, ..., r}) Do
    A' ← {(x, y) ∈ A' : b_xy ≥ b_i}
    For (j ∈ {1, ..., K}) Do
        p_j ← the j-th shortest loopless path in (N, A')
        X ← X ∪ {p_j}
    EndFor
EndFor
k ← 0
```

```
While (X ≠ ∅ and k < K) Do
    k ← k + 1
    p_k ← the quickest loopless path in X
    X ← X − {p_k}
EndWhile
```

Returning to the network of Fig. 1 where $K = 3$ quickest loopless paths are to be determined, Chen's algorithm solves the $K = 3$ shortest loopless paths problem in networks $(\mathcal{N}, \mathcal{A}(2))$, $(\mathcal{N}, \mathcal{A}(4))$ and $(\mathcal{N}, \mathcal{A}(5))$, thus obtaining the loopless paths:[4]

- $q_1^2 = \langle 1, 2, 4, 6\rangle$, $q_2^2 = \langle 1, 3, 2, 4, 6\rangle$ and $q_3^2 = \langle 1, 2, 4, 5, 6\rangle$ in $(\mathcal{N}, \mathcal{A}(2))$;
- $q_1^4 = \langle 1, 3, 5, 4, 6\rangle$, $q_2^4 = \langle 1, 3, 5, 6\rangle$ and $q_3^4 = \langle 1, 2, 3, 5, 4, 6\rangle$ in $(\mathcal{N}, \mathcal{A}(4))$;
- $q_1^5 = \langle 1, 3, 5, 6\rangle$ and $q_2^5 = \langle 1, 2, 3, 5, 6\rangle$ in $(\mathcal{N}, \mathcal{A}(5))$.

By selecting the 3 quickest paths among the distinct ones, $q_1^2, q_2^2, q_3^2, q_1^4, q_2^4, q_3^4, q_2^5$, we have $p_1 = q_2^4$, $p_2 = q_2^5$ and, finally, $p_3 = q_1^4$.

As mentioned above, besides the motivation, Chen's algorithm also differs from Rosen et al.'s and Pascoal et al.'s algorithms in the methodology. In fact, only when all the Kr paths have been computed and stored are we able to pick up $p_1, \ldots, p_K$. This also requires that K be known in advance. In terms of the execution time, this results in a sharp increase in the phase of determining the Kr paths, and it becomes almost irrelevant when simply selecting the K best ones among those paths. On the other hand, Rosen et al.'s and Pascoal et al.'s algorithms alternate determination of each p_k (selected in X) and computation of new candidates stored in X.

Furthermore, Chen's algorithm has to determine more paths than the two other algorithms. As for the memory, both Rosen et al. and Pascoal et al. use $\mathcal{O}(m + Kn)$ worst-case space, when considering that only K loopless paths are stored, while in this respect, Chen's approach is of $\mathcal{O}(m + Krn)$. The computational time depends on the use of Yen's or Katoh et al.'s approach. Therefore, Rosen et al. and Chen's algorithms (if using Yen's method) have $\mathcal{O}(Krn(m + n \log n))$ worst-case, since, for each p_k analysed, they solve $\mathcal{O}(n)$ shortest path problems in each of the r subnetworks of $(\mathcal{N}, \mathcal{A})$. Here Pascoal et al. and Chen's algorithms (with Katoh et al.'s method) solve at the most three shortest path problems in $(\mathcal{N}, \mathcal{A}(b_1)), \ldots, (\mathcal{N}, \mathcal{A}(b_r))$, for each p_k. Therefore the time complexity is of $\mathcal{O}(Kr(m + n \log n))$. It should be noted that Katoh et al.'s method, and therefore Pascoal et al.'s and Chen's algorithm, when using the former researcher's method, are only valid in undirected networks.

A new variant of Chen's algorithm. As mentioned above, in terms of efficiency, it is difficult to compare the methods of Rosen et al. and of Pascoal et al., which determine and select each p_k along the algorithm, with Chen's method, which separates determination and selection of paths into two distinct phases. It was also noted that the two phases are reflected in the need for a larger storage space. These two issues may be solved by adapting Chen's algorithm for ranking simple paths. This variant is also inspired in Chen's algorithm for ranking unconstrained paths, as it keeps several candidates for $p_1, \ldots, p_K$, each of a certain type. Whenever one of these candidates is chosen as some p_k, it is replaced by another loopless path of the same type. This variant is an immediate consequence of Theorem 5,

[4] Note that, in this case all the loopless paths from 1 to 6 have to be determined.

Corollary 1. *Let p_{jk} be the k-th shortest loopless path from s to t in $(\mathcal{N}, \mathcal{A}(b_j))$, for any $k \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, r\}$. Then, p_i is the quickest loopless path in $\{p_{jk} : 1 \leq k \leq i \wedge 1 \leq j \leq r\} - \{p_1, \ldots, p_{i-1}\}$.*

Therefore, not unlike Chen's algorithm for loopless paths, one intends to transform the ranking of the K quickest loopless paths into the ranking of K shortest loopless paths in $(\mathcal{N}, \mathcal{A}(b_1)), \ldots, (\mathcal{N}, \mathcal{A}(b_r))$. The new variant maintains the shortest loopless path that has not been chosen as some p_k, $k \in \{1, \ldots, K\}$, in each $(\mathcal{N}, \mathcal{A}(b_i))$, which will be denoted by P_i, $i = 1, \ldots, r$. Loopless path p_k is found by selecting a P_i such that $T(P_i) = \min_{1 \leq j \leq r}\{T(P_j)\}$, once $p_1, \ldots, p_{k-1}$ are determined. After that P_i is updated with the next shortest loopless path in $(\mathcal{N}, \mathcal{A}(b_i))$, obtained by any algorithm for ranking shortest loopless paths. Since the same loopless path can be traversed in several networks, some of them can be computed more than once, therefore $p_k = P_i$ if and only if $P_i \notin \{p_1, \ldots, p_{k-1}\}$. The new variant of Chen's algorithm is summarised below.

Algorithm 5. *Variant of Chen's algorithm to rank K quickest loopless paths*

$(b_1, \ldots, b_r) \leftarrow$ arc bandwidth values by increasing order
$\mathcal{A}' \leftarrow \mathcal{A}$
`For` $(i \in \{1, \ldots, r\})$ `Do`
  $\mathcal{A}' \leftarrow \{(x, y) \in \mathcal{A}' : b_{xy} \geq b_i\}$
  $P_i \leftarrow$ the shortest path in $(\mathcal{N}, \mathcal{A}')$
`EndFor`
$k \leftarrow 0$
`While` $(k < K$ and $\{P_1, \ldots, P_r\} \neq \emptyset)$ `Do`
  $p \leftarrow P_i$ such that $T(P_i) = \min_{1 \leq j \leq r}\{T(P_j)\}$
  $\mathcal{A}' \leftarrow \{(x, y) \in \mathcal{A} : b_{xy} \geq b_i\}$
  $P_i \leftarrow$ the next shortest loopless path in $(\mathcal{N}, \mathcal{A}')$
  `If` $(p \notin \{p_1, \ldots, p_k\})$ `Then`
    $k \leftarrow k + 1$
    $p_k \leftarrow p$
  `EndIf`
`EndWhile`

In applying Chen's new variant to the previous example we obtain $q_1^2 = \langle 1, 2, 4, 6\rangle$, $q_1^4 = \langle 1, 3, 5, 4, 6\rangle$ and $q_1^5 = \langle 1, 3, 5, 6\rangle$ in networks $(\mathcal{N}, \mathcal{A}(2))$, $(\mathcal{N}, \mathcal{A}(4))$ and $(\mathcal{N}, \mathcal{A}(5))$, respectively, thus concluding that $p_1 = \langle 1, 3, 5, 6\rangle$. This loopless path is replaced by the second shortest path in $(\mathcal{N}, \mathcal{A}(5))$, $q_2^5 = \langle 1, 2, 3, 5, 6\rangle$, which is also p_2. No new loopless path is found now, and so $p_3 = q_1^4 = \langle 1, 3, 5, 4, 6\rangle$. Note that only 4 loopless paths were computed, instead of the 7 determined with the former algorithm.

The theoretical worst-case for this variant coincides with Chen's original algorithm for the K quickest loopless paths, since the number of loopless paths listed is, at most, Kr. So this variant is of $\mathcal{O}(Knr)$ in memory and of $\mathcal{O}(Knr(m + n \log n))$ or $\mathcal{O}(Kr(m + n \log n))$ in time, when using Yen's or Katoh et al.'s algorithm, respectively, as the original algorithm. However, in an average case the new variant is expected to achieve better results than the former version.

It is also possible to adopt a similar strategy to rank paths eventually with loops, although in this case the complexity order is worse than the one established by Chen for that problem.

3. Applications and empirical experiments

This formulation of quickest path problems tends naturally to be used to model transmission issues, where the required transmission time depends both on the size of the data and the characteristics of the network. This includes transportation and telecommunication problems, such as the transportation of cargo between two locations, road transportation, or information to be sent via a communications network. Very recently Clímaco et al. (2003) presented the application of Pascoal et al.'s algorithm to the Internet packet routing. Their work focuses on determining alternative routes for data packets, taking into account uncertainty and/or reliability issues, and reports the corresponding computational results.

Despite the practical interest of the quickest path problem and the quickest paths ranking problem, to the best of our knowledge the work by Pascoal, Captivo, and Clímaco (2005) is the only one to present comparative computational results. The versions tested in their work concern the ranking of loopless path algorithms by Rosen et al., Chen (using both Yen and Katoh et al.'s algorithms) and Pascoal et al.. The computational tests reported considered random, complete and grid undirected networks. The results differ in the case of Rosen et al.'s or Pascoal et al.'s approaches, that use a K shortest loopless paths algorithm-like method, besides Chen's approach. The main conclusion to be drawn from these results, in keeping with the theoretical complexities of the algorithms, is that, in general, the algorithms based on Katoh et al.'s method outperform the ones based on Yen's method.

As mentioned above, the experimental results are also presented in Clímaco et al. (2003), concerning the application of Pascoal et al.'s algorithm to the specific problem of routing packets in the Internet.

References

Boffey, T.B. (1996). "Multiobjective Routing Problems." *TOP*, 3(2), 167–220.

Boffey, T.B., R.C. Williams, B. Pelegrín, and P. Fernandez. (2002). "The Maximum Capacity Shortest Path Problem: Generation of Efficient Solution Sets." *RAIRO Oper. Res.*, 36, 1–19.

Calvete, H.I. (2004). "The Quickest Path Problem with Interval Lead Times." *Computers & Operations Research*, 31(3), 383–395.

Chen, G.-H. and Y.-C. Hung. (1993). "On the Quickest Path Problem." *Information Processing Letters*, 46(3), 125–128.

Chen, Y.L. (1993). "An Algorithm for Finding the K Quickest Paths in a Network." *Computers & Operations Research*, 20, 59–65.

Chen, Y.L. (1994). "Finding the K Quickest Simple Paths in a Network." *Information Processing Letters*, 50, 89–92.

Chen, Y.L. and Y.H. Chin. (1990). "The Quickest Path Problem." *Computers & Operations Research*, 17(2), 153–161.

Clímaco, J.C.N. and E.Q.V. Martins. (1981). "On the Determination of the Nondominated Paths in a Multiobjective Network Problem." In *Proc. of the V Sympösium über Operations Research, Köln, 1980, in Methods in Operations Research*, vol. 40, pp. 255–258. Anton Hain, Königstein.

Clímaco, J.C.N. and E.Q.V. Martins. (1982). "A Bicriterion Shortest Path Algorithm." *European Journal of Operational Research*, 11, 399–404.

Clímaco, J.C.N., M.M.B. Pascoal, M.E.V. Captivo, and J.M.F. Craveirinha. (2003). "Internet Packet Routing: Application of a K-Quickest Path Algorithm." In *Proc. of the III International Conference on Decision Support for Telecommunications and Information Society*, pp. 29–36. Warsaw, Poland. (`www.mat.uc.pt/~marta/Publicacoes/rank_IP.ps.gz`).

Dijkstra, E. (1959). "A Note on Two Problems in Connection with Graphs." *Numerical Mathematics*, 1, 269–271.

Hansen, P. (1980). "Bicriterion Path Problems." In G. Fandel and T. Gal (Eds.), *Multiple Criteria Decision Making: Theory and Applications*, Lectures Notes in Economics and Mathematical Systems, vol. 177, pp. 109–127. Springer Heidelberg.

Kagaris, D., G.E. Pantziou, S. Tragoudas, and C.D. Zaroliagis. (1999). "On the Computation of Fast Data Transmissions in a Network with Capacities and Delays." *Networks*, 33(3), 167–174.

Katoh, N., T. Ibaraki, and H. Mine. (1982). "An Efficient Algorithm for K Shortest Simple Paths." *Networks*, 12, 411–427.

Lee, D.T. and E. Papadopoulou. (1993). "The All-Pairs Quickest Path Problem." *Information Processing Letters*, 45(5), 261–267.

Lin, Y.-K. (2003). "Extend the Quickest Path Problem to the System Reliability Evaluation for a Stochastic-Flow Network." *Computers & Operations Research*, 30(4), 567–575.

Martins, E.Q.V. (1984). "On a Special Class of Bicriterion Path Problems." *European Journal of Operational Research*, 17, 85–94.

Martins, E.Q.V., M.M.B. Pascoal, D.M. L.D. Rasteiro, and J.L.E. Santos. (1999a). "The Optimal Path Problem." *Investigação Operacional*, 19(1), 43–60, 1999. (`www.mat.uc.pt/~marta/Publicacoes/opath.ps.gz`).

Martins, E.Q.V., M.M.B. Pascoal, and J.L.E. Santos. (1999b). "Deviation Algorithms for Ranking Shortest Paths." *The International Journal of Foundations of Computer Science*, 10(3), 247–263. (`www.mat.uc.pt/~marta/Publicacoes/deviation.ps.gz`).

Martins, E.Q.V. and J.L.E. Santos. (1997). "An Algorithm for the Quickest Path Problem." *Operations Research Letters*, 20, 195–198.

Moore, M.H. (1976). "On the Fastest Route for Convoy-Type Traffic in Flowrate-Constrained Networks." *Transportation Science*, 10, 113–124.

Pascoal, M.M.B., M.E.V. Captivo, and J.C. N. Clímaco. (2005). "An Algorithm for Ranking Quickest Simple Paths." *Computers & Operations Research*, 32(3), 509–520.

Rao, N.S.V. (2004). "Probabilistic Quickest Path Algorithm." *Theoretical Computer Science*, 312(2–3), 189–201.

Rosen, J.B., S.Z. Sun, and G.L. Xue. (1991). "Algorithms for the Quickest Path Problem and the Enumeration of Quickest Paths." *Computers & Operations Research*, 18(6), 571–584.

Yen, J.Y. (1971). "Finding the K Shortest Loopless Paths in a Network." *Management Science*, 17, 712–716.