# A Bicriterion Approach for Routing Problems in Multimedia Networks

**João C. N. Clímaco**
*Faculdade de Economia da Universidade de Coimbra, 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*

**José M. F. Craveirinha**
*Departamento de Engenharia Electrotécnica e de Computadores, Polo II da Universidade de Coimbra, Pinhal de Marrocos, 3030-290 Coimbra, Portugal*

*Instituto de Engenharia de Sistemas e Computadores—Coimbra, Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal*

**Marta M. B. Pascoal**
*Centro de Informática e Sistemas and Departamento de Matemática, Polo I da Universidade de Coimbra, Apartado 3008, 3001-454 Coimbra, Portugal*

**Routing problems in communication networks supporting multiple services, namely, multimedia applications, involve the selection of paths satisfying multiple constraints (of a technical nature) and seeking simultaneously to "optimize" the associated metrics. Although traditional models in this area are single-objective, in many situations, it is important to consider different, eventually conflicting, objectives. In this paper, we consider a bicriterion model dedicated to calculating nondominated paths for specific traffic flows (associated with video services) in multiservice high-speed networks. The mathematical formulation of the problem and the bicriterion algorithmic approach developed for its resolution are presented together with computational tests regarding an application to video-traffic routing in a high-speed network. The algorithmic approach is an adaptation of recent work by Ernesto Martins and his collaborators, namely, the MPS algorithm.** © 2003 Wiley Periodicals, Inc.

**Keywords:** path ranking; bicriterion shortest path; multimedia network; routing problems

## 1. INTRODUCTION

Routing problems in communication networks supporting multiple services, namely, multimedia applications, involve the selection of paths satisfying multiple constraints of a technical nature, designated as QoS (Quality of Service) requirements and seeking simultaneously to "optimize" the chosen objective functions. The objective functions are concerned with the necessity of minimizing the consumption of (transmission) resources along a path and to obtain a minimum negative impact on all other traffic flows that may use the network. The specific models of these cost functions and of the QoS constraints depend on the type of multimedia service associated with the "calls" which are being routed from origin to destination. In this context, the term traffic flow is the representation (usually of a stochastic nature) of the calls associated with a given application/service, which are being offered and transported by the network. Typical objective functions are the number of arcs (usually designated in telecommunications as hops or links) and the cost of accepting a call in each arc, as measured by an appropriate traffic model related to the bandwidth available in each link. As for the constraints on the paths, in the case of multimedia applications, these are typically the minimum bandwidth required by the call and the maximum allowed delay and jitter.

Although traditional models in this area are single-objective, in many situations, it is important to consider different, eventually conflicting objectives. Routing algorithms that have been employed in current networks, or proposed for this type of problem, are heuristics based on the Dijkstra or Bellman–Ford shortest path algorithms. Significant examples of this type of approach are found in Kompella et al. [7], Lee et al. [8], and Pornavalai et al. [16].

Received October 2001; accepted January 2003
*Correspondence to*: M. M. B. Pascoal; marta@mat.uc.pt
Contract grant sponsor: Fundaçáo para a Ciência e Tecnologia (FCT); contract grant number: POCTI/GES/37707/2001
Contract grant sponsor: Portuguese Ministry of Science and Technology (MCT)

Having in mind to explore the multicriterion nature of this type of problem, this paper considers a bicriterion model dedicated to calculating the whole set of nondominated paths for traffic flows associated with multimedia-type services in multiservice networks. For this purpose, an exact algorithmic approach is developed based on the bicriterion shortest path algorithm by Clímaco and Martins [5] and on the MPS algorithm [11, 12]. Note that both algorithms belong to a research stream headed by Ernesto Martins at the University of Coimbra during the last two decades. The speed of the proposed approach in calculating the set of nondominated solutions seems to make it rather appropriate for application to the selection of nondominated solutions in networks of practicable size, up to certain limits, as discussed in the analysis of computational results and in the Conclusions.

The major contributions of this paper are the following. It is an application of a bicriterion shortest path model to a multimedia network routing problem (concerning the relevance of multicriterion shortest path models in practical applications, see [15]). As far as we know, this is the first time that an exact algorithm is used for obtaining the solutions of a bicriterion model of this specific type; for this purpose, it was necessary to adapt a ranking algorithm for generating the set of nondominated paths. It would be expected that the use of a labeling algorithm could be a better approach; however, the explicit consideration of additional constraints in the bicriterion shortest path problem showed that the ranking algorithm leads to a better performance. This new approach was tested on randomly generated networks and on U.S. intercity-based networks, thereby simulating realistic types of applications. Although the number of nondominated solutions obtained is not very high, the advantages of using a bicriterion approach in many problems of this type will be made clear.

The notation, the basic definitions, and the mathematical formulation of the routing problem are presented in Section 2 of the paper. The algorithmic approach dedicated to the calculation of the set of nondominated solutions is described in Section 3. An application of this model to a specific routing problem of video traffic in a high-speed network together with extensive computational results are presented in Section 4. Conclusions concerning the inherent advantages of this approach and its applicability are outlined in the final section.

## 2. MATHEMATICAL FORMULATION

In this section, we will recall some basic concepts and present the multimedia traffic routing problem and its formulation in terms of a bicriterion shortest path problem.

In a teletraffic routing problem, we consider a representation of a communication network, the nodes of which may represent routers, servers, or switches and the arcs of which represent links in the network with a certain transmission capacity expressed in terms of bandwidth.

Let $(\mathcal{N}, \mathcal{A})$ represent an undirected network, where $\mathcal{N} = \{v_1, \ldots, v_n\}$ denotes the set of nodes and $\mathcal{A} = \{a_1, \ldots, a_m\}$ denotes the set of arcs (or links). Every arc $a_k \in \mathcal{A}$ corresponds to an unordered pair $(i, j)$, where $i, j \in \mathcal{N}$. Two distinct nodes are considered in this network: $s$ (the initial node) and $t$ (the terminal node).

With no loss of generality, it is assumed that there is, at most, one arc between a given pair of nodes. Therefore, a path from $i \in \mathcal{N}$ to $j \in \mathcal{N}$ in $(\mathcal{N}, \mathcal{A})$ is defined as a sequence of nodes in the network, $p = \langle i \equiv v'_1, v'_2, \ldots, j \equiv v'_\ell \rangle$, where $(v'_k, v'_{k+1}) \in \mathcal{A}$, for any $k \in \{1, \ldots, \ell - 1\}$. A path is said to be a null path if it is formed only by one node. A cycle (or loop) is a path with no repeated nodes, except the first one which coincides with the last. A path $p$, the nodes of which are all different, that is, a path without cycles, is said to be a loopless path.

The set of paths (loopless paths) from $i$ to $j$ in $(\mathcal{N}, \mathcal{A})$ will be denoted by $\mathcal{P}_{ij}$ $(\bar{\mathcal{P}}_{ij})$ and $\mathcal{P}_{st}$ $(\bar{\mathcal{P}}_{st})$ will be denoted by $\mathcal{P}$ $(\bar{\mathcal{P}})$. A subpath of a path $p$ is a subsequence of nodes of $p$. Let $u$ be a node of $p$; then, $\text{sub}_p(s, u)$ represents its subpath from $s$ to $u$. Given two paths $p \in \mathcal{P}_{iu}$ and $q \in \mathcal{P}_{uj}$, the concatenation of $p$ and $q$, denoted by $p \diamond q \in \mathcal{P}_{ij}$, is the path formed by $p$ and followed by $q$. Sometimes, $p \diamond (i, j)$ will be written instead of $p \diamond \langle i, j \rangle$.

Let us now introduce the multimedia traffic routing problem as a network problem. Assume that three values are associated with each arc (or link) $(i, j)$ in $(\mathcal{N}, \mathcal{A})$ namely, $c_{ij} > 0$, representing the cost of $(i, j)$, $b_{ij} > 0$, representing the available bandwidth of $(i, j)$, and, finally, $d_{ij} > 0$, representing the associated delay. Moreover, let $c$, $b$, and $d$ be functions which assign to each path $p$, respectively, $c(p) = \sum_{(i,j) \in p} c_{ij}$, $b(p) = \min_{(i,j) \in p}\{b_{ij}\}$ and $d(p) = \sum_{(i,j) \in p} d_{ij}$. Let us still consider another function $h$ which assigns to each path $p$ its number of arcs. Note that the cost $c_{ij}$ of accepting a call on arc $(i, j)$ is, in general, a function of some associated link working condition and the objective of minimizing $c$ is to obtain the most favorable traffic distribution in the overall network (maximum traffic carried). In most models, $c_{ij}$ is a function of the blocking probability or the available bandwidth of the arc.

Given the values $\Delta_{\text{jitter}} \in \mathbb{N}$, $\Delta_{\text{bandwidth}} \in \mathbb{R}^+$, and $\Delta_{\text{delay}} \in \mathbb{R}^+$, the goal of the problem presented in this work is to determine loopless paths $p \in \bar{\mathcal{P}}$ with, simultaneously, a minimum cost and a minimum number of arcs and also satisfying the following constraints:

- $b(p) \geq \Delta_{\text{bandwidth}}$;
- $d(p) \leq \Delta_{\text{delay}}$;
- $p$ has at most $\Delta_{\text{jitter}}$ arcs (jitter constraints are expressed in terms of the maximum number of arcs).

In other words, considering $f : \mathcal{P} \to \mathbb{R}^2$ such that $f(p) = (c(p), h(p))$, we want to

$$
\begin{array}{rll}
\text{“min”} & \{f(p) : p \in \bar{\mathcal{P}}\} & (1) \\
\text{s. a.} & b(p) \geq \Delta_{\text{bandwidth}} & (2) \\
 & d(p) \leq \Delta_{\text{delay}} & (3) \\
 & h(p) \leq \Delta_{\text{jitter}}. & (4)
\end{array}
$$

Thus, this problem may be considered as a bicriterion loopless path problem where the loopless paths to be computed should satisfy several additional constraints.

Initially, we will ignore the constraints over the paths being determined and analyze only the underlying bicriterion shortest path problem. In general, such a problem does not have a solution, in the sense that if there is a conflict between the considered functions it may happen that no path minimizes both functions simultaneously. Thus, the procedure usually used consists of determining a set of solutions called "efficient" solutions, in the sense that there is no other feasible path which improves one objective function without worsening at least one of the other objective functions. These concepts are summarized in Definitions 1 and 2.

**Definition 1.** *Given $p$, $q \in \mathcal{P}_{ij}$, it is said that $p$ dominates $q$ or that $q$ is dominated by $p$, $p_D q$, if and only if $c(p) \leq c(q)$, $h(p) \leq h(q)$, and at least one of the inequalities is strict.*

**Definition 2.** *A path $q \in \mathcal{P}_{ij}$ is said to be dominated if and only if there is another path $p \in \mathcal{P}_{ij}$ such that $p_D q$.*

Thus, a path is nondominated if there is no other path which dominates this one, and the resolution of problem (1) will consist of computing the nondominated paths or usually a subset of $\mathcal{P}_N$ which is defined in Definition 3.

**Definition 3.** *The set $\mathcal{P}_N = \{p \in \mathcal{P} : \nexists q \in \mathcal{P}$ such that $q_D p\}$ is called set of nondominated paths from $s$ to $t$ in $(\mathcal{N}, \mathcal{A})$.*

Theorem 1 states that, under some assumptions, the bicriterion shortest path problem and the bicriterion shortest loopless path problem are equivalent. This theorem's proof is omitted since it is a generalization of a well-known result for the single-objective case.

**Theorem 1.** *Assume that for any cycle $\mathcal{C}$ in the network $c(\mathcal{C}) \geq 0$ and $h(\mathcal{C}) \geq 0$. Then, the bicriterion shortest path problem and the bicriterion shortest loopless path problem are equivalent.*

Using this theorem, and since $c(p) > 0$ and $h(p) > 0$, for any nonnull path $p$ in $(\mathcal{N}, \mathcal{A})$, we can conclude that it is sufficient to solve the bicriterion shortest path problem with constraints (2), (3), and (4).

## 3. ALGORITHM DESCRIPTION

To solve this bicriterion shortest path problem with additional constraints, we will now introduce an adaptation of an algorithm for the bicriterion shortest path problem:

The first algorithm for bicriterion problems was presented in 1980 by Hansen [6] and it is a labeling algorithm, generalizing Dijkstra's algorithm for the single-objective shortest path problem. Later, in 1984, this algorithm was generalized by Martins [10], for more than two objective functions.

Simultaneously, another class of algorithms for the multicriterion problem was developed, based on the adaptation of ranking path algorithms. The first contribution for an algorithm of this class was due to Clímaco and Martins [4] and it concerned multicriterion optimal path problems. Later on, this algorithm was specialized by the same authors for the bicriterion case with linear objective functions [5]. In [18], Skriver presented a very complete survey on bicriterion shortest path algorithms.

On the other hand, in [4] and [5], it was suggested that the enumeration of paths, in the context of these algorithms, be made by an algorithm due to Martins [9]. Originally, this algorithm was conceived to rank optimal paths, but in the years following its specialization for the $K$ shortest path problem, it was successively improved and its versions originated the so-called class of deletion algorithms [1–3, 13, 14].

More recently, Martins et al. presented other approaches [11, 12] concerning the $K$ shortest path and $K$ shortest loopless path problems. In these works, another class of algorithms for the determination of paths (known as deviation algorithms) was described. Among them is the highly efficient MPS algorithm [11]. Concerning the loopless paths determination, those papers also described the possibility of adapting deviation algorithms to decrease the generation of paths with loops, thus creating more efficient algorithms.

Concerning the complexity order, Hansen [6] presented a family of graphs for which the number of nondominated paths grows exponentially, proving the bicriterion shortest path problem to be intractable. He also showed the adaptation of the label-setting algorithm to the bicriterion shortest path problem to be pseudo-polynomial, that is, it solves this problem in polynomial time, depending on the instance's characteristics.

This labeling algorithm is supported by an adaptation of the Optimality Principle for the shortest path problem, which states that every nondominated path is formed by nondominated subpaths (also valid when determining loopless paths). It may be proved that if the network does not contain negative cycles then the bicriterion shortest path problem satisfies the Optimality Principle. The algorithm constructs the tree of nondominated paths from $s$ to every node, and a set of labels containing the costs of each of those paths is associated with each node. In each step, the node with a lexicographically smallest label is chosen, assuring that it is correct. If that label is not dominated, it corresponds to a nondominated path starting at $s$ and it allows other nodes to be labeled.

Some computational tests comparing the label-setting algorithm and the ranking algorithm (using the MPS algorithm for loopless path determination) were made, where the first one showed a better performance (see also [17]). However, for the current problem to be solved, one cannot generalize the Optimality Principle. Consider, for instance, the network represented in Figure 1(a) and the respective

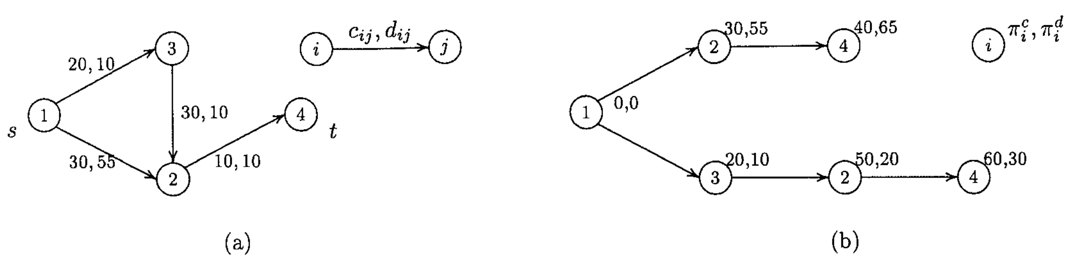


FIG. 1. (a) Network ($\mathcal{N}$, $\mathcal{A}$); (b) tree of paths rooted at $s$ in ($\mathcal{N}$, $\mathcal{A}$).

tree of paths represented in Figure 1(b), where $\pi_i^c$ and $\pi_i^d$ denote the cost and delay, respectively, of the path from $s$ to $i$ in that tree.

Notice that to simplify the example only the constraint associated with the path delay is considered, with $\Delta_{\text{delay}} = 60$. Recalling that it is intended to compute the path with the least cost and the fewest number of arcs, it can be seen in Figure 1(b) that $p = \langle 1, 3, 2, 4 \rangle$ is the only solution of the problem. Nevertheless, the subpath of $p$ from 1 until 2 is dominated by $\langle 1, 2 \rangle$, which is formed only by one arc and the cost of which is 30, lower than $c(\langle 1, 3, 2 \rangle) = 50$. Therefore, one may conclude that it is not possible to perform the dominance test at the time a node is to became an element of the tree rooted at $s$, but only when it corresponds to the determination of a path from $s$ until $t$. More tests were made using this procedure, but now the adaptation of the label-setting algorithm has shown to be less efficient than is the adaptation of the ranking loopless path algorithm. Thus, in the following, only the ranking algorithm approach will be considered. This approach consists mainly of using a ranking algorithm to list paths by nondecreasing costs and using that ordered listing to choose the paths which are nondominated.

Concerning the MPS algorithm's adaptation for ranking loopless paths, it has a very efficient performance from a computational point of view, both in terms of the running times and number of paths it generates to compute the intended loopless paths. These have been two of the reasons which led us to use this adaptation as a subroutine of the algorithm proposed in [5] for computing the set of nondominated loopless paths in the multimedia traffic routing problem.

For a better understanding of the resolution of the problem, Clímaco and Martins' algorithm for the bicriterion problem and the MPS algorithm will now be briefly described. With no loss of generality, $c$ will be assumed to be the first objective function, and $h$, the second one, that is, it will be assumed that the paths are ranked according to $c$.

As mentioned above, ranking paths in a bicriterion problem consists of determining paths by a nondecreasing order of their costs, which allows us to partition $\mathcal{P}_N$ into several subsets. Thus, some results proved in [4] and [5] and which support the bicriterion algorithm will now be stated: The first one, Lemma 1, establishes a ranking stopping condition.

**Lemma 1.** *Let $\mathcal{P}_c$ be the set of paths from $s$ to $t$ with minimum cost and $\mathcal{P}_h$ be the set of paths from $s$ to $t$ with a minimum number of arcs. Let $h^* = \min\{h(p) : p \in \mathcal{P}_c\}$ and $c^* = \min\{c(p) : p \in \mathcal{P}_h\}$. If $p$ is a nondominated path, then $h(p) \leq h^*$ and $c(p) \leq c^*$.*

The second result allows us to define the dominance test which determines whether a path is nondominated or not. Let $S_c$ be the set $S_c = \{ p \in \mathcal{P} : c(p) \leq c^* \}$ with a finite number of paths (which can be ordered by their costs $c$) and consider the following $S_c$ partition:

$$S_c = \bigcup_{i=1}^{k} S_c^i, \text{ where } S_c^i \cap S_c^j = \varnothing \text{ for any } i \neq j,$$

and given $p \in S_c^i$ and $q \in S_c^j$:

- If $i < j$, then $c(p) < c(q)$,
- If $i = j$, then $c(p) = c(q)$.

**Lemma 2.** *Let $p^* \in S_c^i$, for some $i \in \{2, \ldots, m\}$. Path $p^*$ is nondominated if and only if $h(p^*) = \min\{h(p) : p \in S_c^i\}$ and $h(q) > \min\{h(p) : p \in S_c^i\}$, for any $q \in S_c^j$ such that $j < i$.*

It is intended to generate the nondominated solutions by an order such that the cost values are nondecreasing, and, thus, the number of arcs should be nonincreasing. Therefore, at a given step of the algorithm, the nondominated candidate paths are stored in a set designated by $\mathcal{P}_X$, and when analyzing $p \in S_c^i$, the dominance test will consist of comparing $c(p)$ and $h(p)$ with $M_c$ and $m_h$, respectively, where $M_c$ is the greatest cost of the determined paths, while $m_h$ is the smallest number of arcs of those paths. As paths are ranked by a nondecreasing order of $c$, two situations may arise: $c(p) = M_c$ or $c(p) > M_c$. In the first case, $p$ is dominated by some other path if $m_h < h(p)$; otherwise, it is a potential nondominated path and thus is stored in $\mathcal{P}_X$. In the second case, a new set $S_c^i$ begins being computed; therefore, $p$ will be the only element in $\mathcal{P}_X$.

However, this problem is not just a bicriterion shortest path problem, once additional constraints have been imposed on the loopless paths, namely, (2), (3), and (4). Hereafter, the paths from $s$ to $t$ satisfying such constraints will be called feasible paths and their determination can be done by using a ranking path algorithm adaptation, whether for general paths or loopless paths. To decrease the number of generated paths, nondominated loopless candidates, the MPS algorithm for loopless paths [11] will be used. This procedure is now sketched in Algorithm 1.

**Algorithm 1:** *Adaptation of a ranking algorithm for the shortest feasible loopless path problem with two objective functions*

```
$p_h^* \leftarrow$ feasible loopless path with minimum number of
    arcs; $\hat{c} \leftarrow c(p_h^*)$;
$p_c^* \leftarrow$ feasible loopless path with minimum cost; $M_c \leftarrow$
    $c(p_c^*)$; $m_h \leftarrow h(p_c^*)$;
$\mathcal{P}_X \leftarrow \emptyset$; $\mathcal{P}_N \leftarrow \emptyset$; $continue \leftarrow$ True; $k \leftarrow 0$;
While ($continue$) Do
    $k \leftarrow k + 1$;
    $p_k \leftarrow k$-th shortest feasible loopless path in $c$;
    If ($c(p_k) = M_c$) Then          /* Dominance test */
      If ($h(p_k) = m_h$) Then $\mathcal{P}_X \leftarrow \mathcal{P}_X \cup \{p_k\}$
      Else
        If ($h(p_k) < m_h$) Then
          $\mathcal{P}_X \leftarrow \{p_k\}$; $m_h \leftarrow h(p_k)$
        EndIf
      EndIf
    Else
      If ($h(p_k) < m_h$) Then
        $\mathcal{P}_N \leftarrow \mathcal{P}_N \cup \mathcal{P}_X$; $\mathcal{P}_X \leftarrow \{p_k\}$; $M_c \leftarrow c(p_k)$; $m_h$
          $\leftarrow h(p_k)$;
        If ($c(p_k) > \hat{c}$) Then $continue \leftarrow$ False
      EndIf
    EndIf
EndWhile
```

Note that, since we are considering only feasible loopless paths, in the following $\mathcal{P}_N$ will denote the set of nondominated feasible solutions. In this description, there are still some unanswered questions, such as the determination of the feasible loopless path with fewer arcs, the determination of the feasible loopless path with minimum cost, and the feasible loopless path ranking. It should also be noticed that in Algorithm 1 $\hat{c}$ is not exactly the value $c^*$ presented in Lemma 1, but an upper bound on the cost of the loopless paths to be determined (i.e., $\hat{c} \geq c^*$). Before analyzing these points, the MPS algorithm for loopless paths ranking will be briefly described; details can be found in [11].

As has been mentioned, this algorithm belongs to the class of deviation algorithms, which is characterized by using a set $X$ of candidates for the $k$-th shortest loopless path, for some $k \in \mathbb{N}$. The first loopless path stored in $X$ is the shortest one (obtained by solving the shortest path problem); after that, the shortest path is repeatedly selected from $X$. If the path chosen is loopless, then it is a $k$-th shortest loopless path for some $k \in \mathbb{N}$; in any event, its nodes are analyzed, to generate new paths "deviating" from $p$ at one of its nodes, candidates for future shortest loopless paths. Denoting by $\mathcal{T}_t$ the shortest path tree rooted at $t$ and by $\mathcal{T}_t(i)$ the loopless path from $i \in \mathcal{N}$ to $t$ in $\mathcal{T}_t$, each new path $q$ is the shortest one deviating from $p$ at the analyzed node and it has the form $q = \text{sub}_p(s, d_q) \diamond (d_q, i) \diamond \mathcal{T}_t(i)$, where $d_q$ (known as the deviation node of $q$) is the analyzed node in $p$ and where the arc $(d_q, i)$ is computed in a specific manner. Path $p$ is said to be the parent of $q$.

To simplify the determination of $q$, the MPS algorithm begins by computing $\mathcal{T}_t$, and after that computation, each arc's cost is replaced by the respective reduced cost to allow one to choose only the arc with the lowest reduced cost when finding the shortest deviation. In the following, the reduced cost associated with a generic arc's cost $g_{ij}$ and the tree $\mathcal{T}_t$ will be denoted by $\bar{g}_{ij} = g_{ij} + \pi_i - \pi_j$, where $\pi_i$ is the cost of the path from $i$ to $t$ in $\mathcal{T}_t$, while the reduced cost of a path $p$ will be given by $\bar{g}(p) = \sum_{(i,j)\in p} \bar{g}_{ij}$. Moreover, the set $\mathcal{A}$ is rearranged and arcs are ordered according to the tail node; those with the same tail node are ordered according to the reduced cost. This is known as the sorted forward star form (see [12]) and its utilization, together with the reduced costs, allows the number of performed operations to decrease. As mentioned, the paths' ranking begins by successively choosing the shortest element in $X$ and generating new paths, by analyzing its nodes following the deviation node. These new generated paths are also stored in $X$. For each analyzed node $x$, a path with the presented form, defined by the arc with tail node $x$ following the one used in $p$, in the sorted forward star form, is computed.

With the algorithms of Clímaco and Martins and the MPS now described, Algorithm 1 for the traffic routing problem will now be analyzed in more detail.

We begin by considering the determination of the feasible loopless path with the fewest arcs. In fact, one could think of using a shortest path algorithm to calculate the loopless path with the minimum number of arcs from $s$ to $t$. However, one cannot guarantee that this path satisfies (2), (3), and (4); therefore, this procedure may not be sufficient to compute the feasible loopless path with the fewest arcs. Yet, the computation of this loopless path can be made by using, once again, the MPS algorithm for ranking loopless paths according to the values of function $h$, but this algorithm has to be suitably adapted since its application has to end as the first feasible loopless path is computed. Furthermore, some steps will also be added to decrease the number of generated nonfeasible paths.

As far as the determination of the least-cost feasible loopless path is concerned, a similar procedure will be used, but now using the $c$ values for ranking paths.

It should be noted that in the model used for simulating the traffic routing problem the upper bound $\Delta_{\text{jitter}}$ depends on the loopless path with the minimum number of arcs. This is the reason why, in Algorithm 2 that follows, constraint (4) is not verified from the very beginning. The application model used for the traffic routing problem will be described in the next section.

It still remains to define the changes in each ranking in order to consider constraints (2) and (3), in the case of function $h$, and constraints (2), (3), and (4), in the case of function $c$. These constraints can be divided into two types: On one hand, we have constraint (2), and on the other, we have constraints (3) and (4), where the corresponding function is linear.

Let us start by analyzing this second set of constraints, recalling that when computing loopless paths, $X$ may contain paths with loops. However, one tries to minimize the number of these kind of paths when generating new candidates, and only nodes preceding the first one which forms a cycle in $p$ (if it exists) are analyzed, as well as only arcs which do not form a cycle with $\text{sub}_p(s, x)$ are considered. The case of constraints (3) and (4) can be treated in an analogous way as Lemma 3 states.

**Lemma 3.** *Let $p \in \mathcal{P}_{ij}$.*

1. *If $d(p) \geq \Delta_{delay}$, then $d(p \diamond (j, x)) > \Delta_{\text{delay}}$.*
2. *If $h(p) \geq \Delta_{jitter}$, then $h(p \diamond (j, x)) > \Delta_{\text{jitter}}$.*

According to this result, one may conclude that if $d(\text{sub}_p(s, v_i)) \geq \Delta_{\text{delay}}$ or $h(\text{sub}_p(s, v_i)) \geq \Delta_{\text{jitter}}$ then neither $p$ nor the paths still to be generated from the nodes following $v_i$ in $p$ are feasible and thus its analysis can be avoided. Moreover, a selection of the arcs selected when analyzing a given node can also be made to decrease the number of computed nonfeasible candidate paths. Moreover, when analyzing node $v_i$, a feasible path contains only feasible arcs; therefore, every arc $(v_i, j)$ such that $d_{v_ij} > \Delta_{\text{delay}}$ is not considered for generating a new candidate. A similar analysis is not necessary if the paths are ranked according to the number of arcs.

With respect to constraint (2), the function $b(p) = \min_{(i,j)\in p}\{b_{ij}\}$ is not linear and so the same procedure cannot be used. However, Lemma 4 allows us to solve this problem.

**Lemma 4.** *Let $p \in \mathcal{P}_{ij}$. If $b_{uv} < \Delta_{\text{bandwidth}}$ for some* arc $(u, v)$ *of $p$, then $b(p) < \Delta_{\text{bandwidth}}$.*

Based on this lemma, one may conclude that no arc satisfying $b_{ij} < \Delta_{\text{bandwidth}}$ can belong to the solutions one wants to compute and its removal from $\mathcal{A}$ assures that every path in the modified network satisfies constraint (2).

Thus, the algorithm for solving the traffic routing problem will be based on the adaptation of the MPS algorithm for the bicriterion shortest loopless path problem, where the arcs which do not satisfy $b_{ij} \geq \Delta_{\text{bandwidth}}$ are removed from $(\mathcal{N}, \mathcal{A})$, where the feasibility of paths is checked, and where the conditions of arc choices when generating candidate paths are changed [recalling that constraints (3) and (4) are included in the former algorithm].

For simplicity, the next algorithm will be divided into two parts, the first of which, Algorithm 2, refers to the computation of the feasible loopless path with minimum cost, when considering a generic cost $g$.

**Algorithm 2:** *MPS algorithm adaptation—Determination of a feasible loopless path with minimum cost g*

```
Delete arcs $(i, j) \in \mathcal{A}$ such that $b_{ij} < \Delta_{\text{bandwidth}}$ from
  the network;
$\mathcal{T}_t \leftarrow$ tree of the shortest paths from $i \in \mathcal{N}$ to $t$
  according to $g$;
$p_g \leftarrow \mathcal{T}_t(s)$;
If ($p_g$ is not defined) Then Stop;
            /* There are no feasible loopless paths */
$\bar{g}_{ij} \leftarrow \pi_i - \pi_j + g_{ij}, \forall (i, j) \in \mathcal{A}$;
Represent $\mathcal{A}$ in the sorted forward star from according to $\bar{g}$;
$d_{p_g} \leftarrow s$; $X \leftarrow \{p_g\}$; feasible $\leftarrow$ False;
While (($X \neq \emptyset$) and (not feasible)) Do
  $p \leftarrow$ path in $X$ such that $\bar{g}(p)$ is minimum;
        /* $p = \langle s \equiv v_1, v_2, \ldots, v_{\ell-1}, v_\ell \equiv t\rangle$ */
  $X \leftarrow X - \{p\}$;
  $i \leftarrow$ index such that $v_i = d_p$;
  While (($v_i \neq t$) and ($\text{sub}_p(s, v_i)$ is loopless) and
    (constraints are satisfied)) Do
    $l \leftarrow$ index such that $a_l = (v_i, v_{i+1})$; $p_i \leftarrow \text{sub}_p(s, v_i)$;
    While (($v_i$ is the tail node of $a_l$) and (($a_{l+1}$ forms
      a loop with $p_i$) or (constraints aren't satisfied))) Do
      $l \leftarrow l + 1$
    EndWhile
    If ($v_i$ is the tail node of $a_l$) Then
      $v_j \leftarrow$ head node of $a_l$; $q \leftarrow p_i \diamond a_l \diamond \mathcal{T}_t(v_j)$; $d_q$
        $\leftarrow v_i$; $X \leftarrow X \cup \{q\}$
    EndIf
    $v_i \leftarrow v_{i+1}$
  EndWhile
  If (($p$ is loopless) and (constraints are satisfied))
    Then feasible $\leftarrow$ True
EndWhile
If (not feasible) Then Stop;
            /* There are no feasible loopless paths */
```

It should be noticed that in determining the tree of the shortest paths with the fewest arcs in Algorithm 2 [i.e., considering $g_{ij} = 1$ for any $(i, j) \in \mathcal{A}$], it is sufficient to use a label-correcting algorithm, where the set of node labels is manipulated in a FIFO manner. Moreover (assuming that arcs not satisfying the bandwidth constraint have been deleted), when ranking paths according to their number of arcs, constraint (3) has to be verified, and when ranking paths according to $c$, the constraints to be checked are (3) and (4).

The correctness of Algorithm 2 follows from the correctness of the MPS algorithm, noticing that nonfeasible paths are not considered and that the generation of a new path is avoided whenever we can assure it is not feasible and it cannot originate any feasible solution (once its subpath from $s$ to the analyzed node violates at least one of the constraints).

Also, recall that after computing $p$ the feasible loopless path with a minimum number of arcs [when $g_{ij} = 1$ for any $(i, j) \in \mathcal{A}$], $\Delta_{\text{jitter}}$ is updated and the upper bound $\hat{c}$, to be used in the enumeration of nondominated loopless paths, can be obtained by considering $\hat{c} = c(p)$. Analogously, after computing $q$, the least-cost feasible loopless path [when $g_{ij} = c_{ij}$ for any $(i, j) \in \mathcal{A}$], values $M_c$ and $m_h$ can be obtained by using $M_c = c(q)$ and $m_h = h(q)$.

In a general manner, Algorithm 3 consists of the adaptation of the MPS algorithm so that constraints are verified (as in Algorithm 2); also a dominance test is added whenever a $k$-th shortest loopless feasible path is determined. The computation of the set $\mathcal{P}_N$ is summarized in Algorithm 3:

**Algorithm 3:** *MPS algorithm adaptation for the traffic routing problem—Determination of* $\mathcal{P}_N$

```
$\mathcal{P}_X \leftarrow \{p_c^*\}$; $continue \leftarrow$ True;
While (($X \neq \emptyset$) and $continue$) Do
  $p \leftarrow$ path in $X$ such that $\bar{c}(p)$ is minimum;
           /* $p = \langle s \equiv v_1, v_2, \ldots, v_{\ell-1}, v_\ell \equiv t\langle$ */
  $X \leftarrow X - \{p\}$;
  $i \leftarrow$ index such that $v_i = d_p$;
  While (($v_i \neq t$) and ($sub_p(s, v_i)$ is loopless) and
        ($d(\mathrm{sub}_p(s, v_i)) < \Delta_{\mathrm{delay}}$) and ($h(\mathrm{sub}_p(s, v_i))$
          $< \Delta_{\mathrm{jitter}}$)) Do
    $l \leftarrow$ index such that $a_l = (v_i, v_{i+1})$; $p_i \leftarrow \mathrm{sub}_p(s, v_i)$;
    While (($v_i$ is the tail node of $a_l$) and
          (($a_{l+1}$ forms a loop with $p_i$) or ($d(p_i \diamond a_{l+1})$
            $> \Delta_{\mathrm{delay}}$) or ($h(p_i \diamond a_{l+1}) > \Delta_{\mathrm{jitter}}$))) Do
      $l \leftarrow l + 1$
    EndWhile
    If ($v_i$ is the tail node of $a_l$) Then
      $v_j \leftarrow$ head node of $a_l$; $q \leftarrow p_i \diamond a_l \diamond \mathcal{T}_t(v_j)$; $d_q$
        $\leftarrow v_i$; $X \leftarrow X \cup \{q\}$
    EndIf
    $v_i \leftarrow v_{i+1}$
  EndWhile
  If (($p$ is loopless) and ($d(p) \leq \Delta_{\mathrm{delay}}$) and ($h(p)$
    $\leq \Delta_{\mathrm{jitter}}$)) Then
    If ($c(p) = M_c$) Then       /* Dominance test */
      If ($h(p) = m_h$) Then $\mathcal{P}_X \leftarrow \mathcal{P}_X \cup \{p\}$;
      Else
        If ($h(p) < m_h$) Then
          $\mathcal{P}_X \leftarrow \{p\}$; $m_h \leftarrow h(p)$
        EndIf
      EndIf
    Else
      If ($h(p) < m_h$) Then
        $\mathcal{P}_N \leftarrow \mathcal{P}_N \cup \mathcal{P}_X$; $\mathcal{P}_X \leftarrow \{p\}$; $M_c \leftarrow c(p)$;
          $m_h \leftarrow h(p)$;
        If ($c(p) > \hat{c}$) Then $continue \leftarrow$ False
      EndIf
    EndIf
  EndIf
EndWhile
```

An analogous algorithm (suitably adapted from Algorithms 2 and 3) could also be sketched by ranking paths according to $h$ instead of $c$, that is, according to the loopless paths' number of arcs.

It should also be remarked that, even if the initial network is connected, some of its paths may not satisfy (2), (3), or (4). Therefore, it is possible that no loopless path in ($\mathcal{N}$, $\mathcal{A}$) is feasible and that the subset of $\mathcal{P}_N$ one wants to compute is empty, in which case the problem has no solution.

The number of operations performed by the presented algorithm depends on the network characteristics. In fact, the algorithm consists of three main phases: Ranking loopless paths twice (each one considering a different objective function) until the best feasible path is computed, and then to continue the second ranking, performing an additional dominance test for each path, until an upper bound (determined after the first ranking) is achieved. If a polynomial algorithm is used for listing loopless paths by order, for instance, Yen's algorithm [19] which has a worst-case complexity of $\mathbb{O}(Kn^3)$ when $K$ loopless paths are listed, then, in the worst case, the number of operations performed by the algorithm presented here is $\mathbb{O}((K_1 + K_2 + K_3)n^3)$, where $K_1$, $K_2$, and $K_3$ represent the number of loopless paths computed in each phase of the algorithm and which cannot be known in advance. It must be remarked that by using the MPS algorithm instead of Yen's algorithm computational performance is improved, although the complexity order of MPS is not polynomial. Further details can be found in [11].

## 4. APPLICATION

To show the applicability and performance of the algorithm, a specific model of application to a routing problem of video traffic in Asynchronous Transfer Mode (ATM)-type networks is now presented. First, the specific model for this application and then results from extensive computational experiments will be presented and discussed. Also, computational results using networks defined from real geographic coordinates will be shown in this section.

### *4.1 Application Model*

First, a simulation of the video traffic routing problem in undirected communication networks with $n$ nodes and $m = 4n$ arcs is presented. Each node of the network corresponds to a point randomly chosen in a rectangular grid with dimension 400 × 240 and a mesh size unit of 10 km. Each node is adjacent to at least 2 and at most 10 other nodes (recall that the average node degree of the network is 4). Besides, the generated networks contain at least a Hamiltonian path, to assure that $\mathcal{P}_{si} \neq \emptyset$ and $\mathcal{P}_{it} \neq \emptyset$ for any $i \in \mathcal{N}$.

The video traffic routing problem was simulated using a flow specification ($\sigma_k$, $r_k$, $S^k_{\max}$), where:

- $\sigma_k = 10\ S_{\max}$ denotes the token bucket size or maximum burst size (in bits),
- $r_k = r = 1.5 \times 10^6$ bits/second is the token generation rate of the leaky bucket (stochastic model associated with the nodes),

TABLE 1. Bandwidth distributions.

| Distribution | $I_0$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ |
|---|---|---|---|---|---|
| Dist 1 | 20% | 20% | 20% | 20% | 20% |
| Dist 2 | 50% | 20% | 15% | 10% | 5% |
| Dist 3 | 41% | 6% | 6% | 6% | 41% |

TABLE 2. No. video traffic routing problems solved.

| $n$ | No. $s$–$t$ pairs for each seed | Total no. problems solved |
|---|---|---|
| 500 | 10 | 100 |
| 1000 | 40 | 400 |
| 1500 | 90 | 900 |
| 2000 | 160 | 1600 |
| 2500 | 250 | 2500 |
| 3000 | 360 | 3600 |

- $S^k_{\max} = S_{\max}$ is the maximum packet size of the flow $k$ (in bits),
  and $S_{\max} = 53 \times 8$ bits, which is the size of an ATM cell.

As in [16], it was assumed that each node in the communication network is modeled as a queueing system which uses the Weighted Fair Queueing (WFQ) service discipline, allowing the problem to be formulated as follows, concerning the delay and jitter constraints. To try to obtain the lowest call blocking probability, the cost function that we used was $c(p) = \sum_{(i,j)\in p} c_{ij}$, where $c_{uv} = 1/b_{uv}$ for any $(u, v) \in \mathcal{A}$. Thus, the values associated with each arc $(i, j)$, where $i$ and $j$ correspond, respectively, to points $(x_i, y_i)$ and $(x_j, y_j)$ in the initial grid, are

- Its available bandwidth, in Mb/second, that is, a random value denoted by $b_{ij} \in \{0.52, 2.52, \ldots, 150.52\}$, which corresponds to a link capacity of 155.52 Mb/second;
- Its cost, given by $c_{ij} = 1/b_{ij}$;
- Its delay, in ms, given by

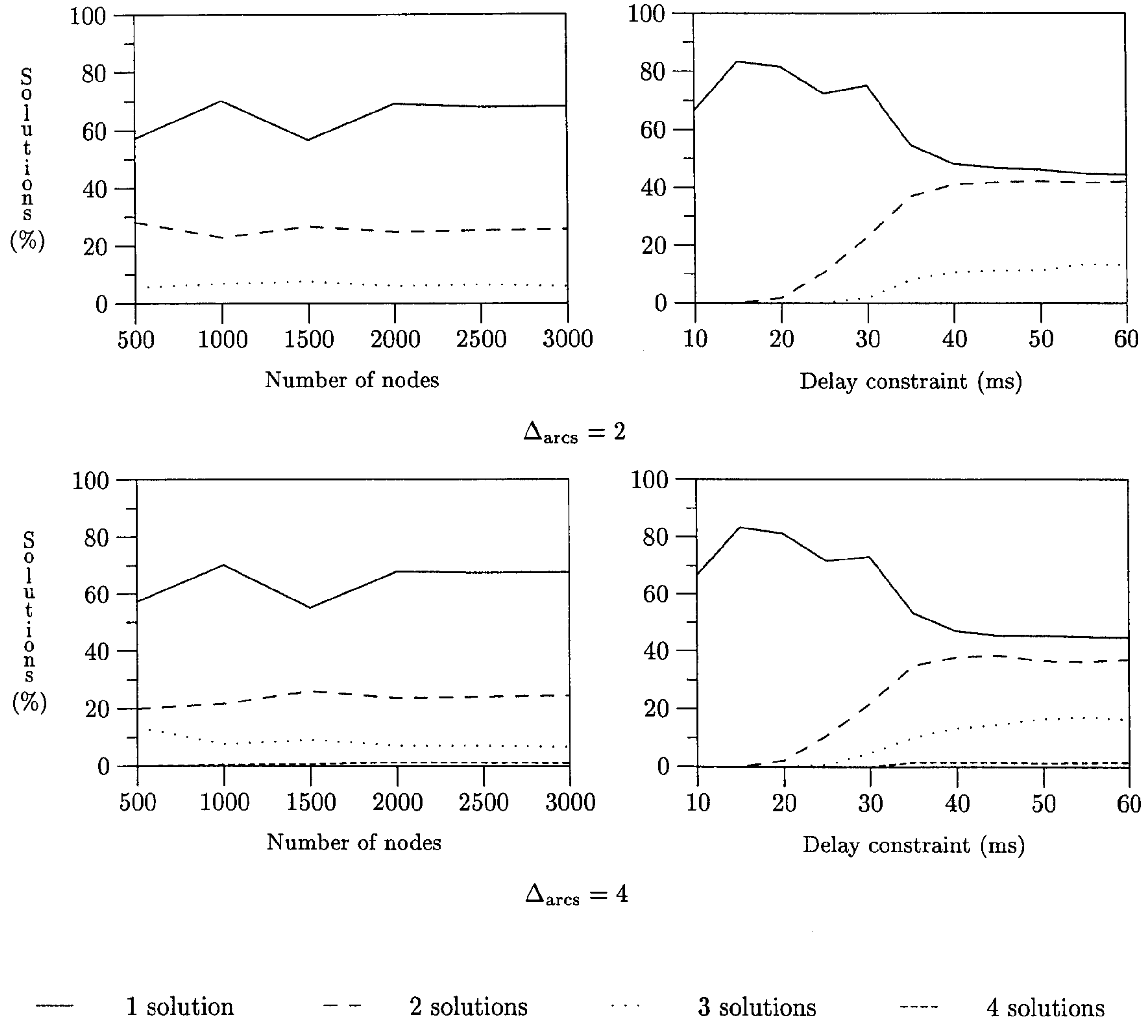


FIG. 2. Percentage of problems with one, two, three, and four nondominated solutions: Distribution 1.

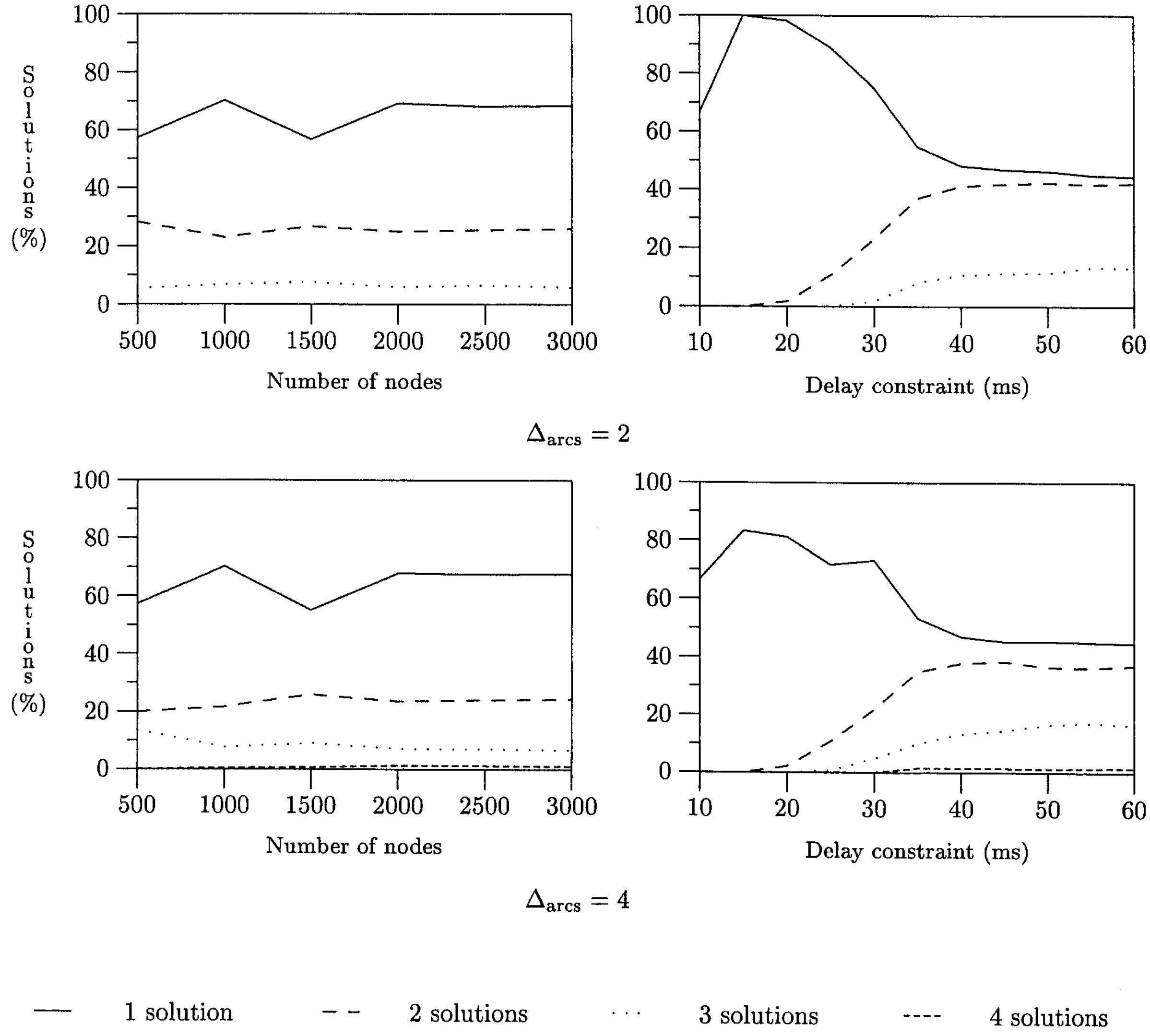


FIG. 3. Percentage of problems with one, two, three, and four nondominated solutions: Distribution 2.

$$d_{ij}=\left(\frac{S^k_{\max}}{r_k}+\frac{S_{\max}}{R_{ij}}\right)+\frac{\ell_{ij}}{2c/3}=\left(\frac{53\times8}{1.5}+\frac{53\times8}{155.52}\right)\times10^{-3}+\frac{\ell_{ij}}{200},$$

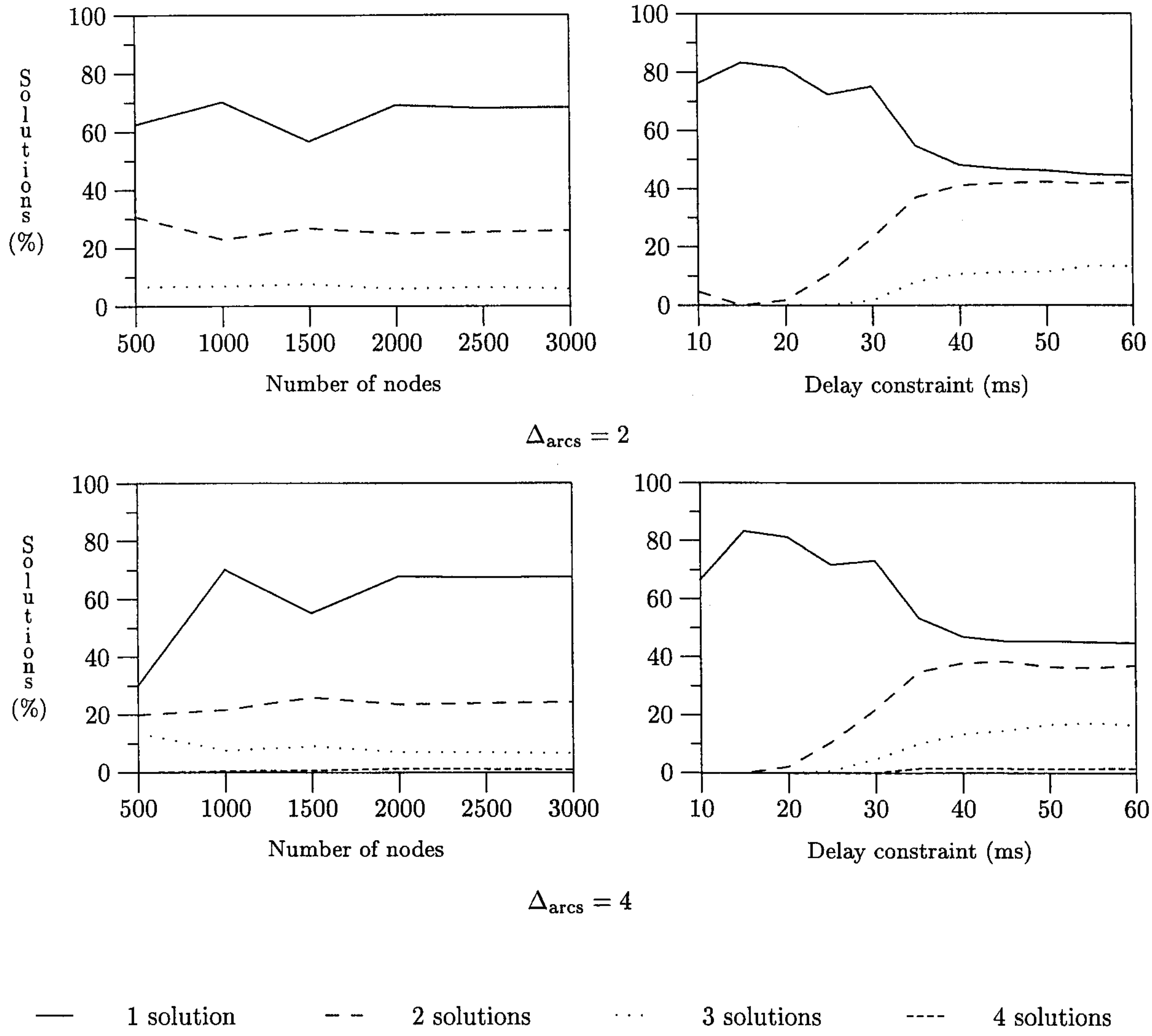
where $\ell_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ represents the Euclidean distance, in km, between $(x_i, y_i)$ and $(x_j, y_j)$, $c = 300$ km/ms is the speed of light, and for any $(u, v) \in \mathcal{A}$, $R_{uv} = 155.52 \times 10^6$ bits/second is the bandwidth capacity of arc $(u, v)$.

Concerning the problem's constraints, we considered that every path should have at least an available bandwidth of $r_k = 1.5$ Mb/second (transmission rate required by the video traffic), that is, $\Delta_{\text{bandwidth}} = 1.5$, and that the delay upper bound $\Delta_{\text{delay}}$ varied in $\{10, 15, \dots, 60\}$ ms. Since one is assuming a WFQ service discipline, the jitter constraint can be expressed in terms of the number of arcs, namely, $\Delta_{\text{jitter}} = ma(s, t) + \Delta_{\text{arcs}}$, where $ma(s, t)$ denotes the minimum number of arcs of a feasible loopless path from $s$ to $t$ and $\Delta_{\text{arcs}} \in \{2, 4\}$.

Initially, the bandwidth values were randomly obtained in $\{0.52, 2.52, \dots, 150.52\}$. Later, other methods were used for generating these values to study the number of nondominated solutions in different problems. In a first set of experiments with randomly generated networks, we considered two types of networks concerning the bandwidth values. The first type consists of partitioning $\{0.52, 2.52, \dots, 150.52\}$ into classes, assigning to each one a predefined percentage of the $b_{ij}$ values. In the experiments performed, five classes with equal size and three different distributions were considered. So, the set $\{0.52, 2.52, \dots, 150.52\}$ was partitioned into

$$\{0.52 + 2k : k = 0, \dots, 75\} = \bigcup_{i=0}^{4} I_i,$$

where



$\Delta_{\text{arcs}} = 2$

$\Delta_{\text{arcs}} = 4$

FIG. 4. Percentage of problems with one, two, three, and four nondominated solutions: Distribution 3.

$$I_i = \{0.52 + 2k : k = 15i, \ldots, 15(i + 1) - 1\},$$
$$i = 0, 1, 2, 3, \text{ and, } I_4 = \{0.52 + 2k : k = 60, \ldots, 75\}.$$

The bandwidth distributions used were those presented in Table 1. It should be noticed that using Distribution 1 corresponds simply to generating values uniformly in $\{0.52, 2.52, \ldots, 150.52\}$, which was the initial procedure.

In the second type of experiment, the bandwidth values were generated considering the following spatial partition of the original grid according to $x$:

$$\{0, 10, \ldots, 400\} \times \{0, 10, \ldots, 240\}$$
$$= (\{0, \ldots, 120\} \cup \{130, \ldots, 250\}$$
$$\cup \{260, \ldots, 400\}) \times \{0, \ldots, 240\}.$$

Thus, given the arc $(i, j)$, where $i$ and $j$ correspond, respectively, to points $(x_i, y_i)$ and $(x_j, y_j)$ in the initial grid,

- If $x_i, x_j \in \{0, 10, \ldots, 120\}$, then $b_{ij} \in \{0.52 + 2k : k = 0, \ldots, 24\}$;
- If $x_i, x_j \in \{130, 140, \ldots, 250\}$, then $b_{ij} \in \{0.52 + 2k : k = 25, \ldots, 49\}$;
- If $x_i, x_j \in \{260, 270, \ldots, 400\}$, then $b_{ij} \in \{0.52 + 2k : k = 50, \ldots, 75\}$;
- Otherwise, $b_{ij} \in \{0.52 + 2k : k = 0, \ldots, 75\}$.

In a second set of experiments, networks were obtained by using geographic coordinates of 1088 U.S. cities from the contiguous 48 states plus the District of Columbia. These data can be found on the Internet, at the url `www.realestate3d.com/gps/latlong.htm`. Considering those cities as nodes of the network, $m = 8n$ arcs between them were randomly generated (assuring that the network is connected). The bandwidth values were randomly generated from $\{0.52, 2.52, \ldots, 150.52\}$ and the additional constraints considered for this problem were the previous ones.

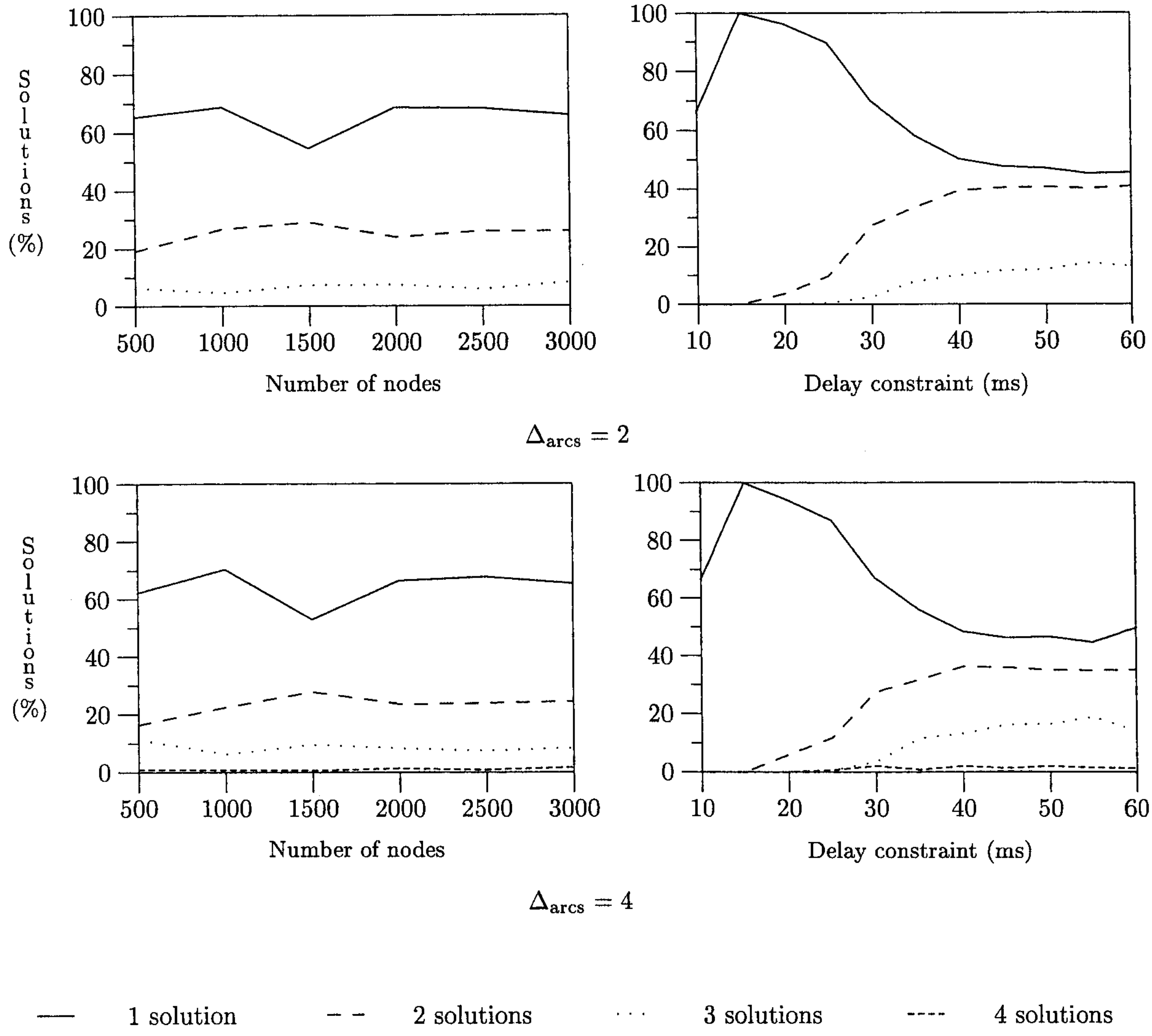


FIG. 5. Percentage of problems with one, two, three, and four nondominated solutions: vertical partition.

The computational results obtained are summarized in the following subsection.

### *4.2. Computational Tests and Results*

In the first set of tests, undirected networks with $n \in \{500, 1000, \ldots, 3000\}$ nodes were used, the number of arcs being $m = 4n$. For each value of $n$, we generated networks using 10 distinct seeds, and for each of them, the algorithm was tested by considering $n^2/25{,}000$ origin–destination node pairs. Table 2 shows the number of origin–destination node pairs for each network and the total number of problems solved for each network size.

For the second set of tests, with networks obtained from the U.S. city coordinates, $m = 8n$ ($n = 1088$) arcs were used and 20 origin–destination node pairs were considered for each one of the 10 chosen seeds.

The code was written in the C language and the computational results were obtained using an AMD Athlon server, running at 1.3 GHz, with 256 Mbytes of RAM and using Linux.

As referred to, several values of $n$ and $\Delta_{\text{delay}}$ have been considered, and for each of these pairs, the average values corresponding to certain features of the problem were calculated. The plots and tables presented below show the variation rate of the average values of some of these features, depending on the parameters $n$ and $\Delta_{\text{delay}}$. The jitter related upper bound was set to $\Delta_{\text{jitter}} = ma(s, t) + \Delta_{\text{arcs}}$, with $\Delta_{\text{arcs}} = 2$, except when studying the number of nondominated solutions of the problem for the randomly generated networks; in this case, we also used $\Delta_{\text{arcs}} = 4$. In the case of the U.S. city-based networks, the values $\Delta_{\text{arcs}} \in \{5, 6\}$ were also used for testing purposes.

The relative number of problems with one feasible ideal solution or 2, 3, or 4 nondominated solutions found by the algorithm for the various empirical statistical distributions of available bandwidth values are represented in Figures 2–4, as a function of the number of network nodes and of

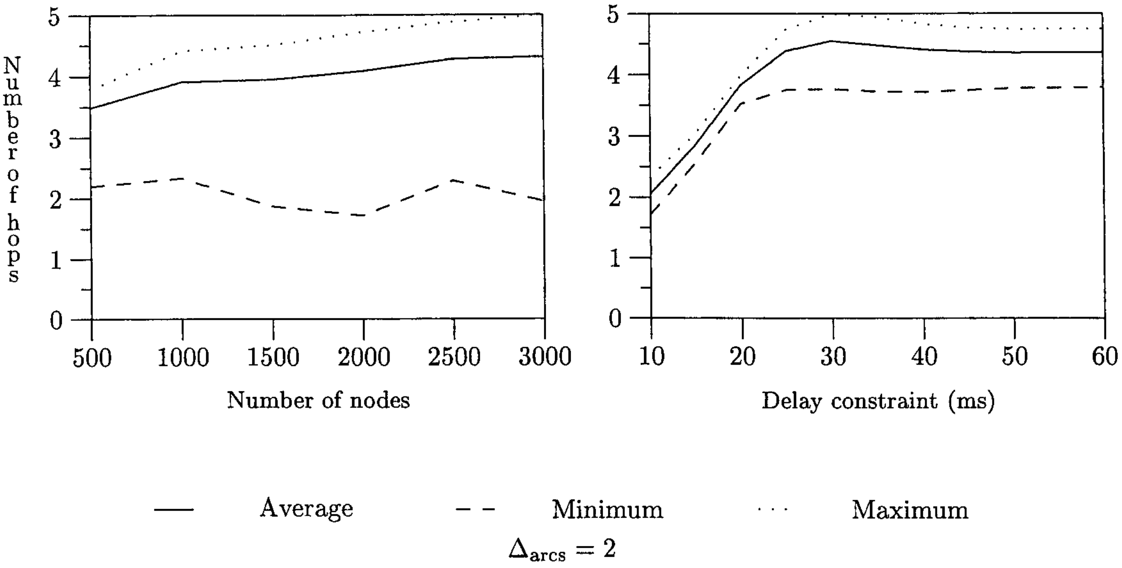


FIG. 6. No. arcs of video traffic.

the delay constraint. In each figure, two sets of curves, one for $\Delta_{\text{arcs}} = 2$ and the other for $\Delta_{\text{arcs}} = 4$, are shown. Also, the relative number of problems with one feasible ideal solution or two, three, or four nondominated solutions found by the algorithm for the spatial partition mentioned in the previous subsection are displayed in Figure 5.

From these results, it is apparent that the increase in the delay constraint (allowing paths with increased length, as shown in Figure 6, where the minimum, the average, and the maximum number of arcs of the nondominated solutions, are presented) tends to produce an increase in the number of problems with a greater number of nondominated solutions, reflecting an increased conflict between the objective functions. The increase in the jitter bound parameter $\Delta_{\text{arcs}}$ from 2 to 4 has an even more relevant impact in the increase in the number of nondominated solutions by allowing the occurrence of problems with four nondominated solutions. It can also be concluded that the profile of the problem's nondominated solutions does not vary significantly with the empirical statistical or spatial distribution of the available bandwidths in the arcs, so that we may consider, with some degree of confidence, that the previous conclusions are representative of the typical features of this type of problem. Also note that there are some problems with no feasible solution, a situation which, when occurring in practice, would naturally require the relaxation of some constraints.

The computational efficiency of the algorithm is illustrated through Figure 7 for the first set of tests, where running times are represented as a function of the number of nodes (values averaged over all delay constraints) and of the delay constraint (values averaged over all numbers of nodes). In Figure 7, the running times increase steadily with the number of nodes, as could be expected. Those times increase very sharply with delay constraints beyond 40 ms as a result of the very rapid increase in the number of candidate paths in Algorithm 3, having in mind typical

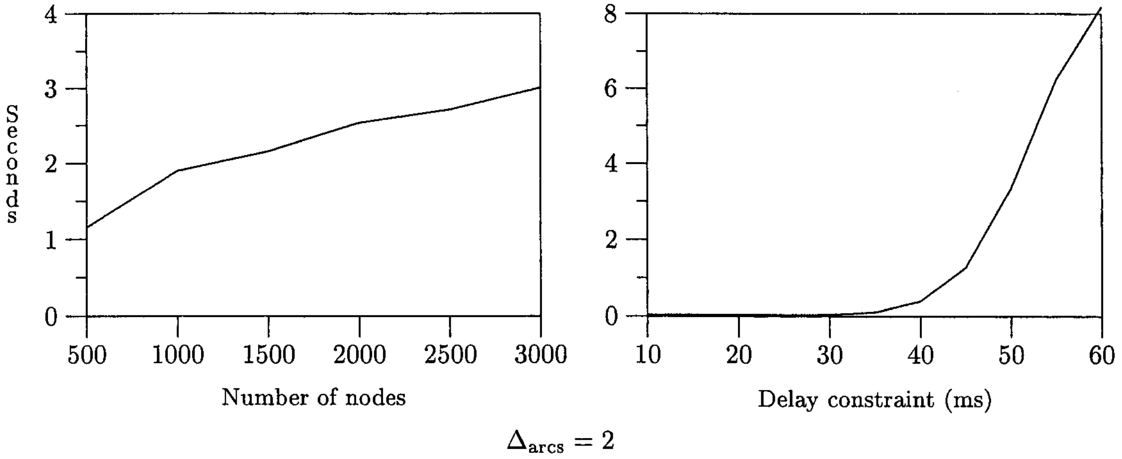
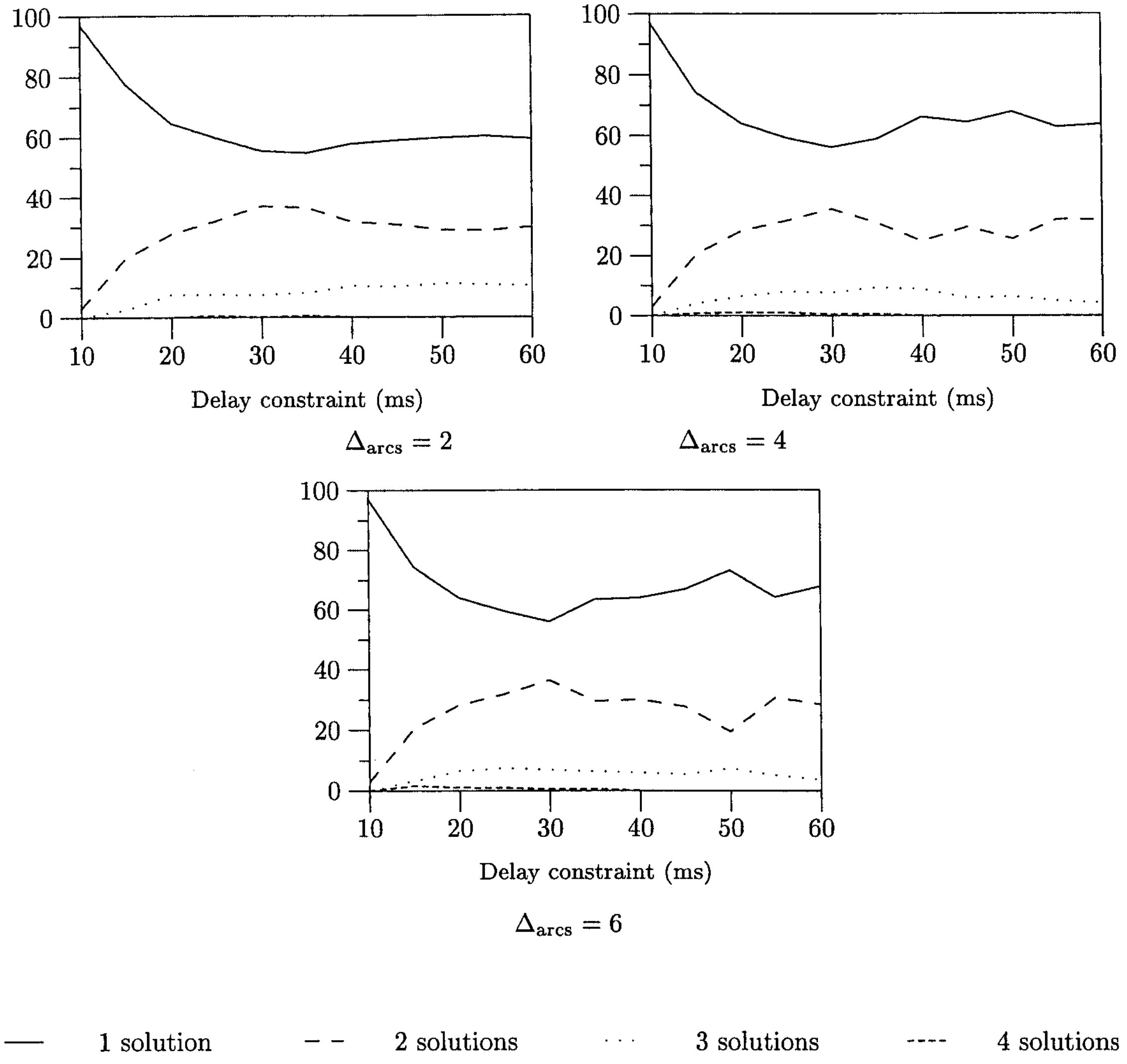


FIG. 7. Running times of video traffic.



$\Delta_{\text{arcs}} = 2$

$\Delta_{\text{arcs}} = 4$

$\Delta_{\text{arcs}} = 6$

FIG. 8. Percentage of problems with one, two, three, and four nondominated solutions in the USA video traffic network.

values of delay in many arcs, rapidly increasing the number of feasible paths with respect to the delay constraint.

The plots in Figures 8 and 9 show the analogous results obtained with the U.S.-based networks. In general, the results for these networks follow the same trends obtained for the randomly generated networks. Taking into account the major conditions considered for generating the random networks and the characteristics of the networks, these results are not surprising. As for the experiments with higher $\Delta_{\text{arcs}}$ values, namely, $\Delta_{\text{arcs}} = 6$ (see Fig. 8), the number of nondominated solutions tends to increase as an expected result of the increase in the upper bound in the jitter constraint. However, it should be noticed that increasing $\Delta_{\text{arcs}}$ implies a substantial increase in the number of feasible paths, hence leading, in some cases, to a situation where the program exceeds the maximum memory allowed, and thus the computation of nondominated solutions is not completed. In Figure 9, the running times follow a trend similar to that of Figure 7 except for high-delay bounds (greater than 35 ms) and $\Delta_{\text{arcs}} = 4$, where the curve starts decreasing. This apparently paradoxical phenomena is explained by the fact that, beyond a certain delay bound, the number of candidate paths increases very rapidly, so that, in this second type of networks (denser and with a greater number of arcs), the effect of lack of memory becomes dominant, leading the algorithm to be unable to terminate in an increasing number of cases.

Note that this algorithmic approach, giving all the nondominated solutions, enables us to solve problems of large size in a few seconds. We considered the number of nodes varying from hundreds to thousands which may be of practical interest in applications to Wide-Area Networks (WANs), where the number of nodes (representing routers or switches) may be very high. By contrast, the heuristics, such as that in Pornavalai et al. [16], have only been tested with some success on networks having up to 500 nodes.

Overall, it may be concluded that, although the objective functions are not strongly conflicting (as a result of the significant correlation between cost and number of arcs), there is a significant percentage of problems with two, three,

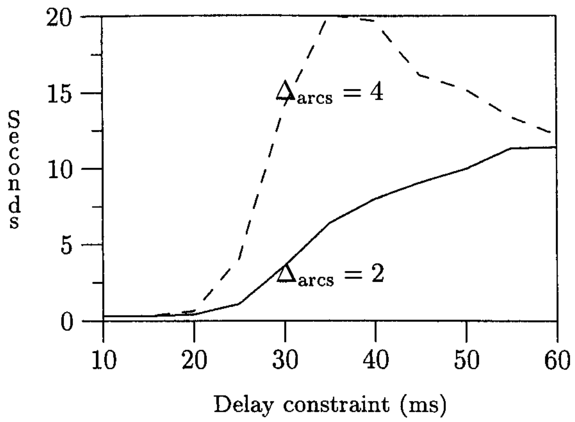


FIG. 9. Average running times of the USA video traffic network.

and even four nondominated solutions that are calculated exactly in relatively short processing times and with modest memory requirements. In many situations, we seek only one solution for each node pair, which could be selected from the nondominated solution set by some practical preference aggregation rule, weighting more or less the cost or the number of links. Concerning the above-mentioned heuristic methods, none of the main features of the proposed approach is possible since they are approximate methods where there is no guarantee of finding an acceptable solution in a reasonable time or even less that it is a nondominated solution.

## 5. CONCLUSIONS

Routing problems in communication networks supporting multiple services, namely, multimedia applications, involve the selection of paths satisfying multiple constraints of a technical nature, designated as QoS requirements and seeking simultaneously to "optimize" the chosen objective functions. Typical objective functions are the number of arcs and the cost of accepting a call in each arc, related to the bandwidth available in each link. As for the constraints on the paths, in the case of multimedia applications, these are, typically, the minimum bandwidth required by the call and the maximum allowed delay and jitter. Although traditional models in this area are single-objective, in many situations, it is important to consider different, eventually conflicting objectives.

To explore the multicriterion nature of this type of problem, we developed, in this paper, a bicriterion model dedicated to the calculation of the whole set of nondominated paths for traffic flows associated with multimedia-type services in multiservice networks. In this context, the major issue dealt with was the adaptation of a ranking-type approach for a bicriterion shortest path problem including additional constraints. For this purpose, an exact algorithm was presented based on the bicriterion shortest path algorithm by Clímaco and Martins [5] and on the MPS algorithm [11, 12].

This model was applied to a specific routing problem of video traffic in a high-speed network and extensive computational results were obtained and discussed. Overall, it may be concluded that, although the objective functions are not strongly conflicting, there is a significant percentage of problems with two, three, and four nondominated solutions in the range of tested networks. Furthermore, the algorithm presented here calculated the set of nondominated solutions in test networks with up to 3000 nodes and an average degree of 4, in relatively short processing times and with modest memory requirements. Note that, as recognized in [16], an average node degree of 4 is close to the typical average node degree in many current real networks of this type. However, for the set of tests with denser networks (with average node degree equal to 8), the execution was affected by lack of memory when the acceptable path delay is beyond certain values. This is due to the rapid increase in the number of candidate paths. Although more extensive tests would be desirable, one can conclude that the exact algorithm presented here should have good computational performance for networks of practicable size up to certain limits with respect to the average node degree and up to certain bounds on the acceptable delay. Beyond such limits, heuristic approaches seem unavoidable.

## Acknowledgments

The authors thank the anonymous referees for the helpful comments on the initial draft of this paper.

## 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, Eur J Oper Res 69 (1993), 97–106.

[2] J.A. Azevedo, J.J.E.R.S. Madeira, E.Q.V. Martins, and F.M.A. Pires, A shortest paths ranking algorithm, Proc Ann Conf AIRO'90, Models and Methods for Decision Support, Operational Research Society of Italy, 1990, pp. 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, Eur J Oper Res 73 (1994), 188–191.

[4] J.C.N. Clímaco and E.Q.V. Martins, On the determination of the nondominated paths in a multiobjective network problem, Proc V Symp über Operations Research, Köln, 1980, *Methods in operations research,* Anton Hain, Königstein, 1981, Vol. 40, pp. 255–258.

[5] J.C.N. Clímaco and E.Q.V. Martins, A bicriterion shortest path algorithm, Eur J Oper Res 11 (1982), 399–404.

[6] P. Hansen, "Bicriterion path problems," *Multiple criteria decision making: Theory and applications,* G. Fandel and T. Gal (Editors), Lectures Notes in Economics and Mathematical Systems, Springer, Heidelberg, 1980, Vol. 177, pp. 109–127.

[7] V.P. Kompella, J.C. Pasquale, and G.C. Polyzos, Multicast routing for multimedia communications, IEEE/ACM Trans Netwrk 1 (1993), 286–292.

[8] W.C. Lee, M.G. Hluchyj, and P.A. Humblet, Routing subject to quality of service constraints in integrated communication networks, IEEE Netwrk July/Aug. (1995), pp. 46–55.

[9] E.Q.V. Martins, An algorithm for ranking paths that may contain cycles, Eur J of Oper Res 18 (1984), 123–130.

[10] E.Q.V. Martins, On a multicriteria shortest path problem, Eur J Oper Res 16 (1984), 236–245.

[11] E.Q.V. Martins, M.M.B. Pascoal, and J.L.E. Santos, A new algorithm for ranking loopless paths, Research report, CISUC, 1997 (http://www.mat.uc.pt/~marta/Publicacoes/mps.ps.gz).

[12] E.Q.V. Martins, M.M.B. Pascoal, and J.L.E. Santos, Deviation algorithms for ranking shortest paths, Int J Found Comput Sci 10 (1999), 247–263 (http://www.mat.uc.pt/~marta/Publicacoes/deviation.ps.gz).

[13] E.Q.V. Martins, M.M.B. Pascoal, and J.L.E. Santos, A new improvement for a *k* shortest paths algorithm, Invest Oper 21 (2001), 47–60 (http://www.mat.uc.pt/~marta/Publicacoes/IMP_MS.ps.gz).

[14] E.Q.V. Martins and J.L.E. Santos, A new shortest paths ranking algorithm, Invest Oper 20 (2000), 47–62 (http://www.mat.uc.pt/~eqvm/cientificos/investigacao/Artigos/K.ps.gz).

[15] M. Mueller-Hannemann and K. Weihe, Pareto shortest paths is often feasible in practice, *Proc 5th Int WAE'01, Lecture Notes in Computer Science,* Aarhus, Denmark, 2001, Vol. 2141, pp. 185–198.

[16] C. Pornavalai, G. Chakraborty, and N. Shiratori, Routing with multiple QoS requirements for supporting multimedia applications, Telecommun Syst 9 (1998), 357–373.

[17] J.L.E. Santos, Uma abordagem ao problema do trajecto óptimo multiobjectivo, Invest Oper 19 (1999), 211–226.

[18] A. Skriver, A classification of bicriterion shortest path (bsp) algorithms, Asia–Pac J Oper Res 17 (2000), 192–212 (http://www.imf.au.dk/cgi-bin/w3-msql/publications/genericpublication.html?publ=242).

[19] J.Y. Fen, Finding the *k* shortest loopless paths in a network, Mgmt Sci 17 (1971), 712–716.