On Shortest Path Problems with "Non-Markovian" Link Contribution to Path Lengths

Conference Paper *in* Lecture Notes in Computer Science · May 2000

DOI: 10.1007/3-540-45551-5_72 · Source: DBLP

Arunabha Sen
Arizona State University

K. Selcuk Candan
Arizona State University

On Shortest Path Problems with "Non-Markovian" Link Contribution to Path Lengths

Arunabha Sen[1], K. Selçuk Candan[1], Afonso Ferreira[2], and Bruno Beauquier[2], and Stephane Perennes[2]

[1] Department of Computer Science and Engineering
Arizona State University
Tempe 85287, AZ, USA
{asen, candan}@asu.edu
[2] SLOOP, CNRS - INRIA - UNSA
2004 Rote des Lucioles, BP 93
F-06902 Sophia-Antipollis, FRANCE
{Afonso.Ferreira, Beaquier, Stephane.Perennes}@sophia.inria.fr

Abstract. In this paper we introduce a new class of shortest path problems, where the contribution of a link to the path length computation depends not only on the weight of that link but also on the weights of the links already traversed. This class of problems may be viewed as "non-Markovian". We consider a specific problem that belong to this class, which is encountered in the multimedia data transmission domain. We consider this problem under different conditions and develop algorithms. The shortest path problem in multimedia data transmission environment can be solved in $O(n^2)$ or $O(n^3)$ computational time.

1 Introduction

Path problems have been extensively studied by many researchers of Computer Science and Operations Research because of its applications in many problems in these domains. In most of these problems, one or more weights are associated with a link representing, among other things, the *cost, delay* or the *reliability* of that link. The objective most often is to find a least weighted path between a specified source-destination pair. In almost all the path problems studied so far (and discussed in the literature), the contribution of a link to the path length computation depends only on the weight of that link and is independent of the weights of the links already traversed. This condition is similar to a Markov chain where the next state is dependent only on the current state and is independent of the past states. In this paper, we introduce a new variant of the path problem. In this variant, the contribution of a link to the path length computation depends not only on the weight of that link but also on the weights of the links already traversed. This class of problems may be viewed as "non-Markovian" as the contribution of a link towards the path length depends on the current link as well as the links traversed in the past on the path from the source to the destination.

G. Pujolle et al. (Eds.): NETWORKING 2000, LNCS 1815, pp. 859–870, 2000.

As an example, we consider a specific problem that belongs to this class. This problem is encountered in the multimedia data transmission domain. We consider this problems under different conditions and develop appropriate algorithms. The shortest path problem in multimedia data transmission environment can be solved in $O(n^2)$ or $O(n^3)$ computational time. We also provide mathematical programming solutions for this problem.

2 Prior Work

Shortest path problems are among the most widely studied problems in Computer Science and Operations Research. Because of its wide applications in many diverse domains, these problems have been studied for at least the last forty years [1,4,5]. In the earliest version of the shortest path problem [1,4,5], a weight is associated with each link of the network and the path length is computed by summing up the weights of the links belonging to the path. In a generalization of this version of the problem, multiple weights are associated with each link of the network. If there are m different weights associated with each link of the network, there are m different path lengths associated with each path. The i-th path length, $(1 \leq i \leq m)$, is obtained by summing up the i-th weights of the links belonging to the path. This version of the shortest path problem is known as *multicriteria shortest path problem* or *constrained shortest path problem* and is fairly extensively studied in [6,8,12]. In view of the attention that the *Quality of Service* issues have received in the networking community in recent years, study of this version of the shortest path problem has become increasingly important [14].

Another version of the shortest path problem that has received considerable attention is the one where the weights associated with the links of the network are allowed to change with time. Both centralized as well as distributed algorithms for the shortest path in this scenario have been developed under various waiting constraints in [9,10]. In yet another version of the problem, each link, e, of the network has two weights, transit time $b(e, u)$ and cost $c(e, u)$, where u is the departure time at the starting node of the link. In this version, the problem is to find the least cost path such that the total traversal time is below some prespecified threshold value T. A dynamic programming algorithm for the shortest path problem with time windows and additional linear costs on node service start times is presented in [7]. In [11] the authors consider a version of the problem termed as the *quickest path problem*, where the objective is to transfer a specified amount of data from the source to the destination with minimum transmission time. The transmission time in this problem is dependent on both the capacities and the traversal times of the links in the network. The shortest path problem in multimedia data transmission environment is discussed in [3].

3 Problem Formulation

In the classical path problem, each edge $e_i \in E$ of the graph $G = (V, E)$ has a weight w_i associated with it and if there is a path P from the node v_0 to v_k

$$v_0 \stackrel{w_1}{\rightarrow} v_1 \stackrel{w_2}{\rightarrow} v_2 \stackrel{w_3}{\rightarrow} \dots \stackrel{w_k}{\rightarrow} v_k$$

then the *path length* or the *distance* between the nodes v_0 and v_k is given by

$$PL(v_0, v_k) = w_1 + w_2 + \dots + w_k$$

This model is valid as long as the weights on the links represents the *cost* or the *delay* associated with the link. However, if the weight represents the *reliability* or the *bandwidth* associated with the link, then *addition* of the link weights on the path is not meaningful. In case, the weights represent the reliability, the calculation once again becomes meaningful if the addition operator is replaced by a *multiplication* operator. In case, the weight represents the bandwidth, the calculation becomes meaningful if a *minimum* operator replaces the addition operator. Thus a generalization of the path length will be

$$PL(v_0, v_k) = w_1 \oplus w_2 \oplus w_3 \oplus \dots \oplus w_k$$

where $\oplus$ is a suitable operator for the particular application. In [14], the authors consider three diferent types of operators and call them *additive, multiplicative*, and *concave metrics* respectively.

At the next level of generalization, the path length computation is based on not the link weight itself but a function of the link weight. In this case the path length is given by

$$PL(v_0, v_k) = f(w_1) \oplus f(w_2) \oplus f(w_3) \oplus \dots \oplus f(w_k)$$

where $f(w_i)$ can be any function of the link weight w_i, appropriate for the particular application.

At the next higher level of generalization each link has multiple weights associated with it. This model realistically captures the data transmission environment where the *Quality of Service (QoS)* issues are of paramount importance. The various weights associated with a link may represent among other things, the *delay*, the *cost*, the *jitter*, the *cell loss rate* etc. In this case the path length computation is carried out in one of the following two ways:

Case I: In this case each path has multiple *path lengths* associated with it. If $(w_{i,1}, w_{i,2}, \dots, w_{i,m})$ are m different link weights associated with the link e_i, then there are m different path lengths, $[PL_1(v_0, v_k), \dots, PL_m(v_0, v_k)]$, associated with a *path* between a given source node v_0 and a given destination node v_k, where

$$PL_i(v_0, v_k) = f(w_{1,i}) \oplus f(w_{2,i}) \oplus \dots \oplus f(w_{k,i})$$

This class of problems is known as the *multicriteria optimization problems* and is studied in [6,8,12,14].

Case II: In this case each path has a single path length associated with it:

$$PL(v_0, v_k) = f(w_{1,1}, \ldots, w_{1,m}) \oplus f(w_{2,1}, \ldots, w_{2,m}) \oplus \cdots \oplus f(w_{k,1}, \ldots, w_{k,m})$$

It may be noted that this formulation gives rise to a *single criterion optimization problem* as opposed to the *multiple criteria optimization problem* in Case I.

Both Case I and Case II of the previous level can be further generalized at the next higher level. As in the previous case, each link has multiple weights associated with them. In this level of generalization, the contribution of a link in the path length computation depends not only on the weights associated with that link but also on the weights of the links already traversed. In this case the path length, $[PL_1(v_0, v_k), \ldots, PL_m(v_0, v_k)]$, for Case I is such that

$$PL_i(v_0, v_k) = f(w_{1,i}) \oplus \cdots \oplus f(w_{1,i}, \cdots, w_{k,i}).$$

At this level of generalization, the path length for Case II is

$$\begin{aligned} PL(v_0, v_k) = f(w_{1,1}, \ldots, w_{1,m}) \oplus f(w_{1,1}, \ldots, w_{1,m}, w_{2,1}, \ldots, w_{2,m}) \oplus \cdots \oplus \\ f(w_{1,1}, \ldots, w_{1,m}, \ldots, w_{k,1}, \ldots, w_{k,m}). \end{aligned}$$

We say that the edges in this category have “non-Markovian” link contributions.

4 Path Problem in Multimedia Data Transmission

The example path problem discussed in this paper belongs to this last category. This problem is based on a multimedia data transmission model we recently presented in [3]. The model allows an active network to perform certain operations to the data at the network nodes. These operations, such as format conversions for distributed multimedia collaborations and lossy/lossless compressions may change the sizes and qualities of multimedia object being transmitted. In this paper, we use a subset of this model, where the quality is not taken into account for path selection.

In the variant of the path problem for the multimedia data transmission, each edge e_i has two weights, δ_i and s_i associated with it, $\delta_i \geq 0$ and $s_i \geq 0$. These two weights are referred to as (i) the *per unit delay factor* and (ii) the *size factor* respectively. If P is a path from the node v_0 to v_k,

$$v_0 \stackrel{\delta_1, s_1}{\rightarrow} v_1 \stackrel{\delta_2, s_2}{\rightarrow} v_2 \stackrel{\delta_3, s_3}{\rightarrow} \ldots \stackrel{\delta_k, s_k}{\rightarrow} v_k.$$

then the *path length* or the *total delay* between the nodes v_0 and v_k denoted $PL(v_0, v_k)$ is given by

$$\begin{aligned} PL(v_0, v_k) &= \delta_1 + s_1\delta_2 + s_1 s_2 \delta_3 + \ldots + s_1 \ldots s_{k-1}\delta_k \\ &= \sum_{i=1}^{k} \delta_i \prod_{j=1}^{i-1} s_j \;\; \text{with} \prod_{j=1}^{0} s_j = 1. \end{aligned}$$

It is clear that the path length in this case fits into the most general case discussed in the previous paragraph with $m = 2, w_{i,1} = \delta_i, w_{i,2} = s_i$ and $f(w_{1,1}, w_{1,2}, w_{2,1}, w_{2,2}, \ldots, w_{i,1}, w_{i,2}) = s_1 s_2 \ldots s_{i-1}\delta_i$, for all i, $1 \leq i \leq k$, and $s_0 = 1$.

The physical significance of the parameters δ_i and s_i are as follows: The transmission delay is clearly proportional to the size of the multimedia data file being transmitted. Therefore we consider the *per unit delay factor* δ_i and to compute the total delay, we multiply δ_i with the size of the file being transmitted. As a multimedia data file travels through different nodes in a network on its journey from the source to the destination, it passes through some transformation algorithms. As a result, the size of the multimedia data file may change. The size factor s_i captures this aspect of multimedia data transmission. We remark, however, that the total delay remains proportional to the amount of data transmitted along the path. Therefore, the expression for the path length (or total delay) stated above is given for transmitting *one unit of data* from the source to the destination.

4.1 Why Is This Problem Different?

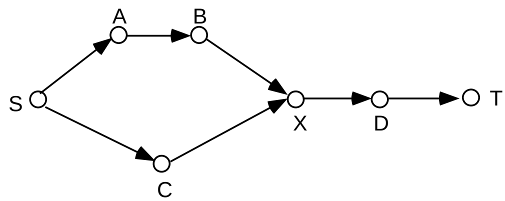
The length of a path $P(v_0, v_k) : v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k$, in the multimedia data transmission problem, is given by $PL(v_0, v_k) = \delta_1 + s_1\delta_2 + s_1 s_2 \delta_3 + \ldots + s_1 \ldots s_{k-1}\delta_k$. The traditional shortest path algorithms such as the *Dijkstra's algorithm* make the observation that *"subpaths of shortest paths are shortest paths"* and exploits it to develop the shortest path algorithm.

In other words, to get to the destination from the source using the shortest path, the intermediate nodes must be visited using the shortest path from the source to the intermediate nodes. This is true because, the *path length* in this case is computed as the sum of weights on the links that make up the path. In case of multimedia data transmission problem, where *the path length is not computed as the sum of the links weights, this is no longer true.* This is demonstrated with an example shown in Figure 1. The δ_i and s_i values associated with the links of this graph is also given in Figure 1.

With this data set the length of the path, $S \rightarrow C \rightarrow X \rightarrow D \rightarrow T$, is $\delta_{S,C} + s_{S,C}\delta_{C,X} + s_{S,C}s_{C,X}\delta_{X,D} + s_{S,C}s_{C,X}s_{X,D}\delta_{D,T} = 1 + 3.1 + 3.1.1 + 3.1.1.1 = 1+3+3+3 = 10$ whereas the length of the path $S \rightarrow A \rightarrow B \rightarrow X \rightarrow D \rightarrow T$ is $\delta_{S,A} + s_{S,A}\delta_{A,B} + s_{S,A}s_{A,B}\delta_{B,X} + s_{S,A}s_{A,B}s_{B,X}\delta_{X,D} + s_{S,A}s_{A,B}s_{B,X}s_{X,D}\delta_{D,T} = 1 + 1.1 + 1.1.1 + 1.1.4.1 + 1.1.4.1.1 = 1 + 1 + 1 + 4 + 4 = 11$. Thus the path $S \rightarrow C \rightarrow X \rightarrow D \rightarrow T$ is shorter than the path $S \rightarrow A \rightarrow B \rightarrow X \rightarrow D \rightarrow T$ in the example. However, in this example the length of the path $S \rightarrow C \rightarrow X$ is $1 + 3.1 = 4$, which is *greater* than the length of the path $S \rightarrow A \rightarrow B \rightarrow X$, $1 + 1.1 + 1.1.1 = 3$.

On the other hand, the path length function in the multimedia data transmission problem has an interesting property and this property is utilized to establish the following lemma.

Lemma 1. *Given a weighted directed graph $G = (V, E)$ with weight functions (δ_i, s_i), associated with each link e_i, $(e_i \in E, 1 \leq i \leq |E|)$, and the length of a*



	(S,A)	(S,C)	(A,B)	(B,X)	(C,X)	(X,D)	(D,T)
Delay factor (δ)	1	1	1	1	1	1	1
Size factor (s)	1	3	1	4	1	1	1

Fig. 1. An Example Graph for MMD Transmission and the Associated Delay and Size Factors

path $P(v_0, v_k) : v_0 \rightarrow v_1 \rightarrow \ldots \rightarrow v_k$ *is computed as* $PL(v_0, v_k) = \delta_1 + s_1\delta_2 + s_1 s_2 \delta_3 + \ldots + s_1 \ldots s_{k-1}\delta_k$. *Let* $P(v_0, v_k) : v_0 \rightarrow v_1 \rightarrow \ldots \rightarrow v_k$ *be a shortest path from vertex* v_0 *to vertex* v_k *and for any* i, $1 \leq i \leq k-1$, *let* $P(v_i, v_k) : v_i \rightarrow v_{i+1} \rightarrow \ldots \rightarrow v_k$ *be a subpath of* P *from vertex* v_i *to vertex* v_k. *Then* $P(v_i, v_k)$ *is a shortest path from* v_i *to* v_k, $1 \leq i \leq k$.

The proof of the lemmas and theorems in this paper are omitted for space considerations. The proofs may be found in [13].

4.2 Path Problem in Multimedia Data Transmission with No Reduction in Size

In this subsection, we consider a special case where the size factor, s_i, associated with a link e_i is greater than or equal to unity for all the links. This implies that the data size will never reduce from its original size while passing through a link. The more general case where the size factor, s_i, does not have any such restriction (i.e., s_i is allowed to be less than unity) will be considered in the next subsection.

Beacuse of lemma 3 and the fact $s_i \geq 1, \delta_i \geq 0$, we can apply a modified version of Dijkstra's algorithm to solve the shortest path problem in the multimedia data transmission environment. The traditional version of the algorithm starts from the source node and computes the shortest path to other nodes until it finds the shortest path to the destination. In this modified version, we start from the destination node and compute the shortest path from other nodes to the destination nodes until it finds the shortest path from the source to the destination node. The algorithm is given in Figure 2.

Theorem 1. *If* $\forall i, j$ *the delay factor* $\delta(i,j) \geq 0$ *and the size factor* $s(i,j) \geq 1$ *then the above algorithm correctly computes the shortest path from any node* $i, 1 \leq i \leq n-1$ *to the destination node* n.

Theorem 2. *The complexity of the algorithm is* $O(n^2)$.

Shortest Path Algorithm for Multimedia Data Transmission Environment

Input: The directed graph $G = (V, E)$, ($V = \{1, 2, \ldots, n\}$), two $n \times n$ matrices δ and s, the (i, j)-th entry of the matrices stores the delay factor δ and the size factor s of the link from the node i to node j. If there is no link from the node i to j, both $\delta_{i,j}$ and $s_{i,j}$ is taken to be ∞. Without any loss of generality, we assume that the node 1 is the source node and node n is the destination node.

Output: Array $D(1, \ldots, n)$, that stores the shortest path length from node i to the destination node n for all i, $1 \leq i \leq n$.

Comments: The algorithm starts from the destination node and in each iteration finds the shortest path from a node i in the graph to the destination node n, $1 \leq i \leq n - 1$.

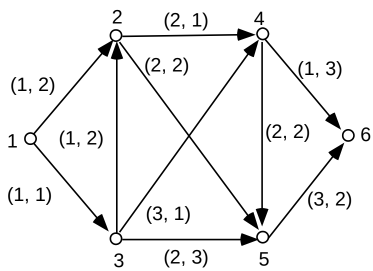
```
begin
    C := {1, 2, ..., n − 1};
    for i := n − 1 downto 1 do
        D[i] := δ[i, n]
    repeat n − 2 times
        begin
            v := i ∈ C such that D[i] has the minimum value;
            C := C \ {v};
            for each w ∈ C do
                D[w] := min(D[w], δ[w, v] + s[w, v]D[v]);
        end
end
```

Fig. 2. Shortest Path Algorithm for Multimedia Data Transmission Environment

An example graph and the corrsonding result of the execution of the algorithm on the graph is shown in Figure 3 (the source node is 1 and the destination is 6). The shortest path length from the node 1 to node 6 is 5 and the path is $v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_6$.

It is well known that if the path length is measured as the sum of the weights on the links, Dijkstra's algorithm fails to compute the shortest path betwen the source-destination nodes, in case some of the link weights are *negative*. For exactly the same reason, our modified version of the Dijkstra's algorithm fails to compute the shortest path if $s_{i,j} < 1$. An example of the case where the above algorithm fails to compute the shortest path is shown in Figure 4. The δ_i and s_i values associated with the links of this graph is also given in Figure 4 (a). The result of the execution of the modified Dijkstra algorithm on this graph is shown in Figure 4.

At the termination of the algorithm, the shortest path length between the source node S and the destination node T is given as 6 and the path is $S \rightarrow A \rightarrow X \rightarrow D \rightarrow T$ ($\delta_{S,A} + s_{S,A}\delta_{A,X} + s_{S,A}s_{A,X}\delta_{X,D} + s_{S,A}s_{A,X}s_{X,D}\delta_{D,T} = 1 + 1.2 + 1.1.2 + 1.1.1.1 = 6$). However, this result is *incorrect* because the length of the path $S \rightarrow A \rightarrow X \rightarrow B \rightarrow C \rightarrow T$ is $\delta_{S,A} + s_{S,A}\delta_{A,X} + s_{S,A}s_{A,X}\delta_{X,B} +$

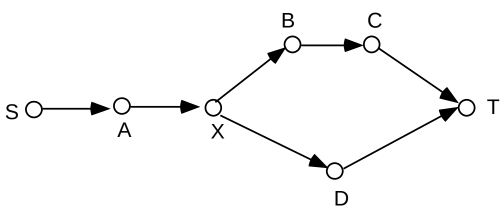


(a)

Iteration	Nodes of the Graph					
	1	2	3	4	5	6
1	∞	∞	∞	1^*	3	0
2	∞	3	4	1	3^*	0
3	∞	3^*	4	1	3	0
4	7	3	4^*	1	3	0
5	5^*	3	4	1	3	0

(b)

Fig. 3. (a) An Example Graph for MMD transmission and (b) the Corresponding Shortest Path Computation



	(S,A)	(A,X)	(X,B)	(X,D)	(B,X)	(C,T)	(D,T)
δ	1	2	1	2	2	2	1
s	1	1	0.25	1	1	1	1

(a)

Iteration	Nodes of the Graph						
	S	A	X	B	C	D	T
1	∞	∞	∞	∞	2	1^*	0
2	∞	∞	3	∞	2^*	1	0
3	∞	∞	3^*	4	2	1	0
4	∞	5	3	4^*	2	1	0
5	∞	5^*	3		2	1	0
6	6^*	5	3	4	2	1	o

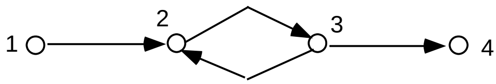
(b)

Fig. 4. (a) An Example Graph for MMD Transmission and (b) the Corresponding Shortest Path Computation

$s_{S,A}s_{A,X}s_{X,B}\delta_{B,C} + s_{S,A}s_{A,X}s_{X,B}s_{B,C}\delta_{C,T} = 1 + 1.2 + 1.1.1 + 1.1.(0.25).2 + 1.1.(0.25).1.2 = 5$. In this case, the algorithm computes the shortest path length incorrectly, because one of the size factors, $s_{X,B} < 1$ ($s_{X,B} = 0.25$).

4.3 Path Problem in Multimedia Data Transmission with Reduction in Size

It was mentioned in the previous section that in this path problem if some size factor $s_i < 1$, it has the same effect as a negative weighted link in a traditional shortest path problem. The example given earlier, shows that our version of the Dijktra's algorithm fails to correctly compute the shortest path from the source to the destination in this situation. In the traditional shortest path problem, where the path length is computed as the sum of the weights on the links of a path, there is a notion of a *negative weighted cycle*. A cycle is referred to as a *negative weighted cycle* if the sum of the weights on the links making up the cycle is a negative number. The multimedia data transmission problem that is



Delay Factor δ	Size Factor s
$\delta_{1,2} = d_1$	$s_{1,2} = 1$
$\delta_{2,3} = \epsilon$	$s_{2,3} = 1/p$
$\delta_{3,2} = \epsilon$	$s_{3,2} = 1/p$
$\delta_{3,4} = d_2$	$s_{3,4} = 1$

Fig. 5. An Example Graph with Negative Weighted Cycle and the Associated Delay and Size Factors

presently under consideration, both the weights (the delay factor δ_i and size factor s_i) associated with a link e_i, are non-negative. However, in this problem path length is computed in a different way. In this problem also we have a notion of a negative weighted cycle. The implication of such a negative weighted cycle is that the data size decreases every time it goes around such a cycle in such a way that the total delay is also reduced every time the data goes around such a cycle. An example of such a phenomenon is given next.

Consider the graph in Figure 5 with the nodes 1 and 4 being the source and the destination respectively. The nodes 2 and 3 form a loop, as shown in the figure. Now, consider the path $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$. This is a *no loop* path between the nodes 1 and 4. The path length of this path is $d_1 + 1.\epsilon + 1.(1/p).d_2 = d_1 + \epsilon + d_2/p$. The length of the path from 1 to 4, if it passes through the loop $1, 2, \ldots, k$ times is

$$d_1 + (1 + 1/p + \ldots + 1/p^{2k})\epsilon + d_2/p^{2k+1}.$$

Thus the path length increases by $((p+1)/p^{2k+2})\epsilon$ and decreases by $((p^2 - 1)/p^{2k+3})d_2$ if the number of times the path goes through the loop increases from k to $k+1$. If d_2 is much larger than ϵ, then the total decrease is much larger than the total increase and as a result if the path goes through the loop one more time, the path length decreases. The situation is similar to the *negative weighted cycle* problem in the traditional shortest path length.

In the path problem for multimedia data transmission environment, we can compute the shortest path between a specified source-destination pair, even with "negative" weights (i.e., with size factor $s_i < 1$) on the links, as long as there is no negative weighted cycles in the graph. We use a modified version of the Bellman-Ford algorithm for this purpose.

Just like the traditional Bellman-Ford algorithm, we find the shortest path lengths subject to the constraint that paths contain at most one link, then relax the condition on the length of the path and find the shortest path length subject to the constraint that paths contain at most two links and so on. Using the same terminology and notations as in [2] we call the shortest path that uses at most h links as the *shortest ($\leq h$) path.*

Suppose that $D^h{}_i$ denotes the shortest ($\leq h$) path length from node i to the destination node n, $(1 \leq i \leq n-1)$. $D^h{}_n = 0$ for all h. The algorithm is given in Figure 6.

Shortest Path Algorithm for Multimedia Data Transmission Environment

Input: The directed graph $G = (V, E)$, $(V = \{1, 2, \ldots, n\})$, two $n \times n$ matrices δ and s, the (i, j)-th entry of the matices stores the delay factor δ and the size factor s of the link from the node i to node j. If there is no link from the node i to j, both $\delta_{i,j}$ and $s_{i,j}$ is taken to be ∞. Without any loss of generality, we assume that the node 1 is the source node and node n is the destination node.

Output: The shortest path length from every node in the graph to the destination node n.

Comments: The algorithm starts from the destination node and in each iteration finds the shortest $(\leq h)$ path from a node i in the graph to the destination node n, $1 \leq i, h \leq n - 1$.

begin
  for $i := 1$ **to** n-1 **do**
    ${D^0}_i := \infty$;
  for $i := 1$ **to** n-1 **do**
    ${D^1}_i := \delta(i, n)$;
  for $h := 1$ **to** n-2 **do**
    for $i := 1$ **to** n-1 **do**
      ${D^{h+1}}_i := \min_{1 \leq j \leq n-1}[s(i, j){D^h}_j + \delta(i, j)]$;
end

Fig. 6. Shortest Path Algorithm for Multimedia Data Transmission Environment

Theorem 3. *If the graph $G = (V, E)$ does not contain any negative weighted cycle, then the above algorithm correctly computes the shortest path length from any node $i, 1 \leq i \leq n - 1$ to the destination node n, even when some of the size factors s_i associated with a link e_i is less than 1.*

Theorem 4. *The complexity of the algorithm is $O(n^3)$.*

An example of the result of the execution of the algorithm on the graph in Figure 4 is shown in Table 1.

4.4 Mathematical Programming Solution to the Path Problem in Multimedia Data Transmission

In this subsection we show that the shortest path problem for the multimedia data transmission problem can also be solved using mathematical programming techniques.

Given a graph $G = (V, E)$ with weights δ_i and s_i associated with each link $e_i \in E$ and two specified vertices s and t, the problem is to find a shortest (or the least weighted) path from s to t.

In the mathematical programming formulation of the problem, we associate a binary indicator variable $x_{i,j}$ with each link i, j of the directed graph $G = (V, E)$.

Table 1. Shortest Path Computation for the Graph in Figure 4 using modified Bellman-Ford Algorithm

Iteration Number	Nodes of the Graph						
	S	A	X	B	C	D	T
1	∞	∞	∞	∞	2	1	0
2	∞	∞	3	4	2	1	0
3	∞	∞	2	4	2	1	0
4	∞	4	2	4	2	1	0
5	5	4	2	4	2	1	0
6	5	4	2	4	2	1	0

By assigning a zero or a one to the variable $x_{i,j}$ the solution indicates whether or not the link i, j is a part of the shortest path from the source to the destination. We also introduce two other variables $y_{i,j}$ and $z_{i,j}$, $(1 \leq i, j \leq |V|)$. By assigning a value to the variable $y_{i,j}$ (resp. $z_{i,j}$), the solution indicates how much data is entering (resp. leaving) the link (i, j).
Minimize $\sum_{(i,j)\in E} \delta_{i,j} y_{i,j}$ *subject to the following constraints:*

1. $\sum_{\{j|(i,j)\in E\}} x_{i,j} - \sum_{\{j|(j,i)\in E\}} x_{j,i} = r_i, \forall i \in V$ $r_i = 1$ if i is the source node, $r_i = -1$ if i is the destination node and $r_i = 0$ if i is any other node.
2. $y_{s,j} = x_{s,j} \forall j \in V - \{s\}$ and $(s, j) \in E$, (s is the source node)
3. $\sum_{\{j|(i,j)\in E\}} y_{i,j} - \sum_{\{j|(j,i)\in E\}} z_{j,i} = 0, \forall i \in V - \{s, t\}$
4. $z_{i,j} = s_{i,j} y_{i,j}, \forall (i, j) \in E$
5. $z_{i,j} \leq K x_{i,j}, \forall (i, j) \in E$ where K is a large constant.

The first constraint establishes a path from the source to the destination. As data passes through a link (i, j), its size changes by a factor $s_{i,j}$. This is ensured by the constraint (iv). The delay keeps accumulating as the data file passes through various links on its journey from the source to the destination. This aspect is captured by the constraint (iii). The purpose of constraint (v) is to ensure that the variable $z_{i,j}$ does not have a non-zero value when $x_{i,j} = 0$, i.e., when the the link (i, j) is not a part of the path from the source to the destination. The contribution of the delay factor $\delta_{i,j}$ associated with the link (i, j) is taken into account in the objective function of the formulation.

5 Conclusion

In this paper we introduced a new class of the shortest path problems. In this class of path problems, the contribution of a link towards the path length depends not only on the weight of the link itself but also on the weights of all the links traversed before traversing the link under consideration. We considered a specific path problem that belong to this new class. This problem is encountered in multimedia data transmission domain. Exploiting an interesting property of the multimedia transmission path problem, we could develop low order polynomial time algorithms for this problem. Additionally, we could solve this problem using mathematical programming techniques. Other path problems that belong to this class are currently under investigation.

References

1. R. Bellman,"On a Routing Problem", *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87-90, 1958.
2. D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987.
3. K.S. Candan and Y.Yang,"Least-Cost High-Quality Object Retrieval for Distributed Multimedia Collaborations", *IEEE Multimedia Computing and Systems Conference*, pp. 649-654, Florence, Italy, June 1999.
4. E.W. Dijkstra, "A Note on Two Problems in Connection with Graphs", *Numerische Mathematik*, vol. 1, pp. 269- 271, 1959.
5. L.E. Ford and S.M. Johnson, "A Tournament Problem", *The American Mathematical Monthly*, vol. 66, pp. 387-389, 1959.
6. G.Y. Handler and I. Zang, " A Dual Algorithm for the Constrained Shortest Path Problem",*Networks*,vol. 10, pp. 293-310, 1980.
7. I. Ioachim *et. al.*, " A Dynamic Programming Algorithm for the Shortest Path Problem with Time Windows and Linear Node Costs", *Networks*,vol. 31, pp. 193-204, 1998.
8. J.M. Jaffe, "Algorithms for Finding Paths with Multiple Constraints", *Networks*,vol. 14, pp. 95-116, 1984.
9. A. Orda and R. Rom, "Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Lengths", *Journal of the Association for Computing Machinery*, vol. 37, no. 3, pp. 605-625, 1990.
10. A. Orda and R. Rom, "Minimum Weight Paths with Time-Dependent Networks", *Networks*, vol. 21, pp. 295-319, 1991.
11. J.B. Rosen, S. Z. Sun and G.L. Xue, "Algorithms for the Quickest Path Problem and the Enumeration of Quickest Paths", *Computers and Operation Research*, vol. 18, no. 6, pp. 579-584, 1991.
12. H.F. Salama, D.S. Reeves and Y. Viniotis, "A Distributed Algorithm for Delay-Constrained Unicast Routing", *IEEE INFOCOM'97*, pp. 1c.2.1-1c.2.8, 1997.
13. A. Sen, K. S. Candan, A. Ferreira, B. Beauquier, S. Perennes, "On Shortest Path Problems with "non-Markovian" Link Contribution to Path Lengths", Tech. Report-00-002, Dept. of Computer Science and Engineering, Arizona State University.
14. Z. Wang and J. Crowcroft, " Quality-of-Service Routing for Supporting Multimedia Applications", *IEEE Journal on Selected Areas of Communications*, vol. 14, no. 7, pp. 1228-1234, 1996.