

Approximation and Heuristic Algorithms for Delay Constrained Path Selection under Inaccurate State Information

Ying Xiao, Krishnaiyan Thulasiraman
University of Oklahoma, Norman, OK
{ying_xiao, thulasi}@ou.edu
Guoliang Xue
Arizona State University, Tempe, AZ
xue@asu.edu

Abstract

Given a communication network modeled as a directed graph with a delay parameter associated with each link, we consider the problem of determining the most probable delay constrained path from a source node to a destination node. Assuming that the link delays are random variables with continuous and differentiable probability density function and using the central limit theorem this problem can be formulated as a path problem which involves simultaneously optimizing two additive path parameters. Two cases arise. When there is one path with mean delay less than the delay bound, we present an exact pseudo polynomial algorithm, a fully polynomial time ε -approximation algorithm and a strongly polynomial heuristic algorithm. In the unlikely case when this assumption is violated, the problem is shown to be NP-hard and no constant factor approximation algorithm exists if $P \neq NP$. We also study the path protection problem under inaccurate state information.

1. Introduction

The constrained shortest path (CSP) problem is to identify a minimum cost path from a source node to a destination node whose delay is within a specified bound. This problem has attracted considerable attention from the telecommunications community. This is a result of a great deal of emphasis on the need to design communication protocols that deliver certain performance guarantees. This need, in turn, is the result of an explosive growth in high bandwidth real time applications that require stringent QoS guarantees. The CSP problem is known to be NP-hard. As a result, research efforts have focused on designing efficient heuristic and approximation

algorithms for this problem. Works in [1] - [7] and the references therein contain most of the results reported to date on the CSP problem.

The above results for the CSP problem have been developed assuming that the exact state of the network is known. However, in practice this is not the case. For several reasons [8], [9], full knowledge of the network state is not available. The existence of inaccuracy in state information has led researchers to study the routing problem with uncertain parameters [8] - [10]. The objective in these papers is to identify a path that is most likely to satisfy the delay requirement. This problem is referred to as the MP-DCP problem. In their pioneering works [8] and [10], the authors studied several aspects of this problem and related computational issues. In their study, they also highlighted the role of the CSP problem and other combinatorial optimization problems in the study of the MP-DCP problem. For the simplicity, unlike the CSP algorithm considering link cost and link delay together on deterministic networks, only one link metric, say delay, is considered in this paper. However, our algorithms can be easily extended to handle multiple constraints.

The MP-DCP is known to be NP-hard [8]. To simplify this problem, Korkmaz and Krunz [9] use the central limit theorem and make mild assumptions on the probability distribution of link delays which lead to a formulation that requires determining an optimal path with respect to a metric involving mean path delay and path variance. They considered two cases, one of which requires minimization of both the mean path delay and path variance and the other requires minimization of mean path delay and maximization of path variance. Using Lagrangian relaxation techniques (as in [3], [5], [6], [7]), they developed heuristic approaches. Intensive numerical simulations given in [9] show that this formulation

produces high quality solutions.

In this paper we develop approximation algorithms for the MP-DCP problem using the formulation developed in [9]. In the following, we formally define the MP-DCP problem and its formulation.

MP-DCP Problem: Consider a network represented by a graph $G(V, E)$, with $n = |V|$, $m = |E|$. Given a maximum delay requirement D for a flow between a given source node s and a destination node $t \neq s$, and probability density function (*pdf*) $p(d)$ for all $l = (i, j) \in E$, such that $p(d)$ is the probability that the link l will introduce a delay at most d units, i.e., $d_l < d$. Let $d(i, j)$ be the random variable (*RV*) associated with the delay of the link (i, j) . For a path p , define

$$d(p) = \sum_{(i,j) \in p} d(i, j) \text{ and } \pi_D(p) = Pr[d(p) \leq D].$$

The MP-DCP problem is to find an s - t path p_{opt} such that for any s - t path p , $\pi_D(p_{opt}) \geq \pi_D(p)$.

To simplify this problem and following Korkmaz and Krunk [9], we assume that $d(i, j)$'s are nonnegative *RV*'s with mean $\mu(i, j) > 0$ and variance $\delta^2(i, j) > 0$ and that for all links $(i, j) \in E$, $d(i, j)$'s are mutually independent. Without loss of generality, we assume $\mu(i, j)$'s and $\delta^2(i, j)$'s are integers (this is because all numbers are represented by finite digits in computers and other digital devices). Furthermore, we assume that the *pdf* of $d(i, j)$ is continuous and differentiable on some interval (a, b) . Under this assumption and using the central limit theorem, the path delay is approximately normally distributed. Without loss of generality, we assume each s - t path is long enough (Note: The sum of as small as three *RV*'s tends to a normal distribution [9]) that $d(p)$ is a normally distributed *RV* with mean $\mu(p) > 0$ and variance $\delta^2(p) > 0$ computed as follows.

$$\mu(p) = \sum_{(i,j) \in p} \mu(i, j) \text{ and } \delta^2(p) = \sum_{(i,j) \in p} \delta^2(i, j)$$

With the above assumption,

$$\pi_D(p) \approx \Phi\left(\frac{D - \mu(p)}{\delta(p)}\right),$$

$$\text{where } \Phi(x) = (1/2\pi)^{1/2} \int_{-\infty}^x e^{-y^2/2} dy$$

Since $\Phi(x)$ is an increasing function, we can reduce the MP-DCP problem as identifying the path p that maximizes

$$\chi_D(p) = (D - \mu(p))/\delta(p),$$

$$\text{where } \delta(p) \equiv \left(\sum_{(i,j) \in p} \delta^2(i, j)\right)^{1/2}.$$

We call $\mu(p)$ and $\delta^2(p)$ as the mean delay and delay variance of path p . The difficulty with this problem arises from the nonseparable square root ($\sqrt{\cdot}$) function. As in [9], we distinguish two cases:

Case 1: There exists a path with mean delay less than

or equal to the specified delay bound D .

Case 2: The mean delay of each path is greater than D .

Obviously, in Case 1 (*resp.* Case 2), $\chi_D(p_{opt}) \geq 0$ and $\mu(p_{opt}) \leq D$ (*resp.* $\chi_D(p_{opt}) < 0, \mu(p_{opt}) > D$).

The rest of the paper is organized as follows. In sections 2 - 4 we consider Case 1. In section 2 we present a pseudo polynomial time exact algorithm for the MP-DCP problem. In section 3 a fully polynomial time ε -approximation algorithm is presented. This is followed by a strongly polynomial time heuristic algorithm and the numerical simulation results for the heuristic algorithm presented in section 4. In section 5 we study the problem for Case 2. Section 6 discusses the application of these results in provisioning two disjoint paths under inaccurate information and its value in designing path protection schemes. Section 7 concludes with a summary of our contributions.

2. An exact algorithm: case 1

In the next three sections, we first assume that there always exists some path p such that $\mu(p) \leq D$. We will also study the MP-DCP problem without invoking this assumption in section 5. In fact, if this assumption does not hold, the value of D may not be a realistic delay bound because in this case, $\forall p, \pi_D(p) < \Phi(0) = 0.5$, i.e., any path p meets the delay bound with probability less than 0.5.

Since $\delta(p)$ is nonadditive, *Procedure exact-mp-dcp* enumerates all the possible values of $\delta^2(p)$ that lie in $[1, U]$ with $U = \min\{\delta^2(p^*) \mid p^* \in P\}$, where $P = \{p^* \mid \mu(p^*) = \min(\mu(p) \mid p \text{ is an } s\text{-}t \text{ path})\}$.

Procedure exact-mp-dcp

```

1   $T_0 \leftarrow U$ 
2   $i \leftarrow 0$ 
3   $\chi_{opt} \leftarrow -\infty$ 
4  while ( $T_i > 0$ )
    /* see next paragraph for special care */
5   $p^* \leftarrow \operatorname{argmin}\{\mu(p) \mid \delta^2(p) \leq T_i\}$ 
6   $\chi_{opt} \leftarrow \max\{\chi_{opt}, (D - \mu(p^*)) / \delta(p^*)\}$ 
7   $i \leftarrow i + 1$ 
8   $T_i \leftarrow \delta^2(p^*) - 1$ 
9  end while
10 return  $\chi_{opt}$  and the corresponding path  $p^*$ 
end procedure
```

For the sake of completeness we also present next the main algorithm for computing $\operatorname{argmin}\{\mu(p) \mid \delta^2(p) \leq T_i\}$ in *Procedure exact-mp-dcp*. When we compute $\operatorname{argmin}\{\mu(p) \mid \delta^2(p) \leq T_i\}$ for the first time, we call the Algorithm *CSP* adopted from the exact algorithm in [2] for the constrained shortest path problem with two metrics on

deterministic networks. This is a simple dynamic programming algorithm with time complexity $O(m T)$. We only need to call the *Algorithm CSP* once and then we can directly use the table $f_i(d)$ (defined in the *Algorithm CSP* below) created by its first invocation. With the table, for a given T_i and $p^* = \arg \min \{\mu(p) \mid \delta^2(p) \leq T_i\}$, it can be seen that $\mu(p^*) = f_i(T_i)$ and $\delta^2(p^*) = T^*$ where T^* is the least value such that $f_i(T^*) = f_i(T_i)$. In fact, the value of $\delta^2(p^*)$ can be computed in constant time using extra data structures.

We keep the formulation of *exact-mp-dcp* as above to make the algorithm conceptually simple.

Let $f_i(d)$ be the minimum mean delay among all 1- j paths with delay variance $\leq d$.

/* T is the delay variance upper bound*/

Algorithm CSP (T):

/* 1 is the source node, n is the target node*/

$f_1(d) = 0, d = 0, \dots, T,$

$f_j(0) = \infty, j = 2, \dots, n,$

$f_j(d) = \min \{f_j(d-1),$

$$\min_{k \mid \delta^2(k,j) \leq d} \{f_k(d - \delta^2(k,j)) + \mu(k,j)\}$$

$\}, j = 2, \dots, n, d = 1, \dots, T$

Theorem 1: Procedure *exact-mp-dcp* finds an optimal solution in $O(U m)$ steps if for all links (i, j) , $\delta^2(i, j) \neq 0$.

Proof: The computation of $\arg \min \{\mu(p) \mid \delta^2(p) \leq T\}$ for the first time is done by the constrained shortest path algorithm which takes $O(T m)$ steps if there are no 0 delay variance links.

Obviously, the computation time of *Algorithm CSP* dominates all the other computations and so the complexity of the whole algorithm is $O(U m)$.

Let p_{opt} be one of the optimal solutions with the least delay variance and p^* be the path such that $U = \delta^2(p^*)$. We first show that $U = \delta^2(p^*) \geq \delta(p_{opt})$. By the definition of p_{opt} and p^* ,

$$\begin{aligned} (D - \mu(p_{opt})) / \delta(p_{opt}) &\geq (D - \mu(p^*)) / \delta(p^*) \text{ and} \\ D &\geq \mu(p_{opt}) \geq \mu(p^*). \end{aligned}$$

If $D - \mu(p_{opt}) > 0$, we have $\delta(p^*) / \delta(p_{opt}) \geq (D - \mu(p^*)) / (D - \mu(p_{opt})) \geq 1$ or $\delta(p^*) \geq \delta(p_{opt})$.

If $D - \mu(p_{opt}) = 0$, then $\chi_D(p^*) = \chi_D(p_{opt}) = 0$ and $D = \mu(p_{opt}) = \mu(p^*)$. Hence $\delta(p^*) \geq \delta(p_{opt})$ because p_{opt} is the optimal path with the least delay variance and p^* is one of the optimal paths.

We next show that Procedure *exact-mp-dcp* will find one of the optimal solutions at termination. Suppose in iteration i , $p_i = \arg \min \{\mu(p) \mid \delta^2(p) \leq T_i\}$. To prove the correctness of the algorithm, it suffices to show that if $\delta^2(p_{opt}) \leq T_i$ and p_i is not optimal, then $\delta^2(p_{opt}) < \delta^2(p_i) = T_{i+1} + 1$. Then we can see that the algorithm has enumerated all possible value of $\delta^2(p_{opt})$ at termination.

If this were not true, then $T_i \geq \delta^2(p_{opt}) \geq \delta^2(p_i)$. Obviously, $D \geq \mu(p_{opt}) \geq \mu(p_i)$ by the definition of p_i and $\delta^2(p_{opt}) \leq T_i$. We obtain

$$(D - \mu(p_{opt})) / \delta(p_{opt}) \leq (D - \mu(p_i)) / \delta(p_i).$$

Since p_i is not optimal by the assumption, this is the desired contradiction. ■

In the next section, based on *exact-mp-dcp*, we shall design a fully polynomial time approximation algorithm.

3. A fully polynomial time approximation algorithm: case 1

To design a fully polynomial time approximation algorithm, we use scaling and rounding described in [4]. Without loss of generality, assume $U \gg n$ and $\varepsilon < 1$.

Lemma 1: Let $G(N, E, \mu, \delta, D)$ denote a network with two metrics μ and δ on the link set E . Let $G_\tau(N, E, \mu, \delta', \tau)$ be the network transformed from G such that

$$\forall (i, j) \in E, \delta'^2(i, j) = \lfloor \tau \delta^2(i, j) / upper \rfloor + 1,$$

where $\tau (= O(n))$ is some integer (to be discussed later) and $lower < upper \leq U$.

Let p_{opt} be the optimal solution to the MP-DCP problem on G and p_τ be any path such that

$$(D - \mu(p_\tau)) / \delta'(p_\tau) \geq (D - \mu(p_{opt})) / \delta(p_{opt}).$$

If $lower \leq \delta^2(p_{opt}) \leq upper$, then

$$\frac{D - \mu(p_\tau)}{\delta(p_\tau)} \geq \left(1 + \frac{upper}{lower} \cdot \frac{L(p_{opt})}{\tau}\right)^{-1/2} \frac{D - \mu(p_{opt})}{\delta(p_{opt})},$$

where $L(p)$ is the number of links of path p .

Proof: We have

$$\delta^2(i, j)(\tau / upper) \leq \delta'^2(i, j) \leq \delta^2(i, j)(\tau / upper) + 1.$$

So

$$\frac{D - \mu(p_\tau)}{\delta(p_\tau)} \geq \frac{D - \mu(p_\tau)}{(upper / \tau)^{1/2} \delta'(p_\tau)}$$

$$\geq \frac{D - \mu(p_{opt})}{(upper / \tau)^{1/2} \delta'(p_{opt})}$$

$$\geq \left(1 + \frac{upper}{\delta^2(p_{opt})} \cdot \frac{L(p_{opt})}{\tau}\right)^{-1/2} \frac{D - \mu(p_{opt})}{\delta(p_{opt})}$$

$$\geq \left(1 + \frac{upper}{lower} \cdot \frac{L(p_{opt})}{\tau}\right)^{-1/2} \frac{D - \mu(p_{opt})}{\delta(p_{opt})} \quad \blacksquare$$

We next present an approximation algorithm *Procedure approx-mp-dcp* for the MP-DCP problem. In each iteration of *approx-mp-dcp*, the algorithm computes a path whose objective is no less than the optimum values among all the paths whose delay variance lies between given values of *lower* and *upper*. This is achieved by calling *approx-max-mp-dcp(lower, upper, \tau, \chi_{opt})* which applies *Procedure exact-mp-dcp* on an appropriately scaled network.

Procedure approx-mp-dcp

```

1   $\tau \leftarrow 2n / \varepsilon$ ,  $upper \leftarrow U$ ,  $lower \leftarrow U / 2$ 
2   $\chi_{opt} \leftarrow -\infty$ 
3  while  $lower \geq 1$ 
4     $p^* \leftarrow \text{approx-max-mp-dcp}(lower, upper, \tau, \chi_{opt})$ 
5     $\chi_{opt} \leftarrow \max\{\chi_{opt}, (D - \mu(p^*)) / \delta(p^*)\}$ 
6     $upper \leftarrow lower - 1$ 
7     $lower \leftarrow upper / 2$ 
8  end while
9  return  $\chi_{opt}$  and the corresponding path
end procedure

```

Procedure approx-max-mp-dcp(lower, upper, τ , χ_{opt})

```

1   $\delta^2(i, j) \equiv \lfloor \tau \delta^2(i, j) / upper \rfloor + 1$  for all link  $(i, j)$ 
2   $L = (lower / upper) \tau$ ,  $\Delta \leftarrow \tau + n$ 
3  while  $(\Delta \geq L)$ 
   /* Using CSP Algorithm on  $\delta'$  */
4   $p_\Delta \leftarrow \text{argmin}\{\mu(p) \mid \delta'^2(p) \leq \Delta\}$ 
   /* Using  $\delta(p_\Delta)$  not  $\delta'(p_\Delta)$  */
5   $\chi_{opt} \leftarrow \max\{\chi_{opt}, (D - \mu(p_\Delta)) / \delta(p_\Delta)\}$ 
6   $\Delta \leftarrow \delta'^2(p_\Delta) - 1$ 
7  end while
8  return  $\chi_{opt}$  and the corresponding path
end procedure

```

Lemma 2: If p_{opt} is the optimal solution to the MP-DCP problem and $lower \leq \delta^2(p_{opt}) \leq upper$, where $lower = upper / 2$, $\tau = 2n / \varepsilon$, *approx-max-mp-dcp* finds a solution p_Δ such that $\chi_D(p_\Delta) \geq \chi_D(p_{opt}) / (1 + \varepsilon)^{1/2}$ in time $O(mn / \varepsilon)$.

Proof: The complexity is easy to show (See Theorem 1).

We next prove the first part of this lemma.

Observe that for any path p in G with $lower \leq \delta^2(p) \leq upper$, $\tau / 2 \leq \delta'^2(p) \leq \tau + n$, where δ' is the metric defined in the line 1 of algorithm *approx-max-mp-dcp*.

Let $p_\tau = \text{argmax}\{(D - \mu(p)) / \delta'(p) \mid \tau / 2 \leq \delta'^2(p) \leq \tau + n\}$. We have

$$\begin{aligned} (D - \mu(p_\Delta)) / \delta(p_\Delta) &\geq (D - \mu(p_\tau)) / \delta(p_\tau) \text{ and,} \\ (D - \mu(p_\tau)) / \delta'(p_\tau) &\geq (D - \mu(p_{opt})) / \delta'(p_{opt}) \end{aligned}$$

This first inequality holds because p_τ is among all the paths delivered by the CSP algorithm invoked in line 4 of *approx-max-mp-dcp* and at termination, p_τ must have been compared with p_Δ (p_Δ is the winner at termination) in updating χ_{opt} (line 5 in *Procedure approx-max-mp-dcp*).

By the first inequality and Lemma 1, we obtain

$$\begin{aligned} \chi_D(p_\Delta) &= (D - \mu(p_\Delta)) / \delta(p_\Delta) \geq (D - \mu(p_\tau)) / \delta(p_\tau) \\ &\geq (1 + L(p_{opt}) upper / (\tau lower))^{-1/2} (D - \mu(p_{opt})) / \delta(p_{opt}) \\ &\geq (1 + \varepsilon)^{-1/2} (D - \mu(p_{opt})) / \delta(p_{opt}) = \chi_D(p_{opt}) / (1 + \varepsilon)^{1/2}. \blacksquare \end{aligned}$$

Theorem 2: *Procedure approx-mp-dcp* finds in time $O((mn / \varepsilon) \log U)$ a path p^* such that $\chi_D(p^*) \geq \chi_D(p_{opt}) / (1 + \varepsilon)^{1/2}$, where p_{opt} is the optimal path for the MP-DCP problem.

Proof: Obviously, the procedure terminates in $O(\log U)$ iterations of *Procedure approx-max-mp-dcp*. Next, we can see that *approx-mp-dcp* must have searched the

interval containing the optimal path before termination as shown in Lemma 2 and thus the theorem is proven. ■

Notice that with proper value of ε , our algorithm can be seen as fully polynomial ε -approximation algorithm defined in [2], [4]. An interesting question is whether we can adopt the techniques in [2], [4] to derive a strongly polynomial algorithm (the time complexity does not depend on U). Unfortunately, (due to the nonseparable nature of objective function), optimality conditions for the MP-DCP problem are not known. So, we are not able to design the *test* or ε -*test* procedures which are critical for the methods in [2], [4].

4. A strongly polynomial heuristic algorithm: case 1

In this section, using parametric search we design a strongly polynomial heuristic algorithm for the MP-DCP problem. The solution obtained by this heuristic can be used to achieve considerable speed up of the *Procedure approx-mp-dcp* presented in the previous sections.

We notice that the objective function of the MP-DCP problem is close to the form of fractional optimization problems that can be solved by Newton method or parametric search [11], [12]. For the MP-DCP problem, the only difficulty is the nonadditive nature of $\delta(p)$. In order to remove this barrier, we change the objective function and consider the following modified problem.

$$\mathbf{H-MP-DCP:} \quad \text{Max } H_{\chi_D}(p) = (D - \mu(p)) / \delta^2(p),$$

$$\text{where } \delta(p) \equiv \left(\sum_{(i,j) \in p} \delta^2(i, j) \right)^{1/2}.$$

Let p_H be the optimal path to the H-MP-DCP problem. Assume $H_{\chi_D}(p_H) = OPT$. In parametric search, for any given λ , we need an oracle test to determine whether OPT is greater or less than λ [11], [12]. Even though the value of OPT is unknown, this can still be achieved by applying Dijkstra's algorithm on the weights $\mu(i, j) + \lambda \delta^2(i, j)$ for all links $(i, j) \in E$. Let p_λ denote the shortest path with respect to $W_\lambda(i, j) = \mu(i, j) + \lambda \delta^2(i, j)$. For the sake of brevity, we present our heuristic algorithm *h-mp-dcp*(G, s, t) using Bellman-Ford-Moore shortest path algorithm instead of Dijkstra's algorithm. For node u , define $N(u) = \{v \mid (u, v) \in E\}$. Each node v of the network is associated with a pair $M_v = (x_v, y_v)$, where x_v and y_v keep track of the mean delay and delay variance of some s - v path during the execution of the *h-mp-dcp* algorithm. M is initialized as $M_s = (0, 0)$ and $M_v = (\infty, \infty)$ for $v \neq s$. The algorithm computes the path p_H without knowing OPT . By the assumption that there always exists a path such that $\mu(p) \leq D$, it can be seen that $OPT \geq 0$.

Algorithm *h-mp-dcp*(*G*, *s*, *t*)

```

1  $M_v = (x_v, y_v) = (\infty, \infty)$  for all nodes
2  $M_s = (0, 0)$ 
3 for  $i \leftarrow 1$  to  $n - 1$  do
4   for each node  $u$  in the network
5     for each  $v$  such that  $v \in N(u)$ 
6       /*oracle test*/
       [ if  $(x_v + OPT y_v \geq x_u + OPT y_u + \mu(u, v) + OPT \delta^2(u, v))$  ]
          $M_v \leftarrow (x_u + \mu(u, v), y_u + \delta^2(u, v))$ 

```

In *h-mp-dcp*, extra steps are required to implement the following oracle test with unknown *OPT*.

$$x_v + OPT y_v \geq x_u + OPT y_u + \mu(u, v) + OPT \delta^2(u, v).$$

If $x_v = \infty$, $y_v = \infty$, then the inequality holds. Assume x_v and y_v are finite (non-negative) values. Then it suffices to evaluate the following Boolean expression.

$$x_u + \mu(u, v) - x_v + OPT(y_u + \delta^2(u, v) - y_v) = p + q OPT \leq 0,$$

where $p = x_u + \mu(u, v) - x_v$ and $q = (y_u + \delta^2(u, v) - y_v)$.

We then only need to determine the sign of $p + q OPT$ (> 0 , < 0 and $= 0$). If $p \cdot q \geq 0$, the sign of $p + q OPT$ is the same as that of p or q recalling that $OPT \geq 0$. In this case implementing the oracle test is obvious.

Consider $p \cdot q < 0$, i.e., $-p / q > 0$. Let $\lambda = -p / q$ and let $p_\lambda = Dijkstra(s, t, W_\lambda)$, where $Dijkstra(s, t, W_\lambda)$ computes the minimal s - t path with respect to W_λ . Now three cases arise.

1. $\mu(p_\lambda) + \lambda \delta^2(p_\lambda) < D$: This implies that $\lambda < (D - \mu(p_\lambda)) / \delta^2(p_\lambda) \leq OPT$.

2. $\mu(p_\lambda) + \lambda \delta^2(p_\lambda) = D$: This implies that $\mu(p_\lambda) + \lambda \delta^2(p_\lambda) = D \leq \mu(p_H) + \lambda \delta^2(p_H)$.

Thus $(D - \mu(p_\lambda)) / \delta^2(p_\lambda) = \lambda \geq (D - \mu(p_H)) / \delta^2(p_H) = OPT$ which implies $\lambda = OPT$.

3. $\mu(p_\lambda) + \lambda \delta^2(p_\lambda) > D$: Then $\mu(p_H) + \lambda \delta^2(p_H) > D$ and $\lambda > (D - \mu(p_H)) / \delta^2(p_H) = OPT$.

With the path p_λ , we can easily decide the sign of $p + q OPT$ by the above three cases.

Theorem 3: (a). The time complexity of algorithm *h-mp-dcp* is $O(mn(m + n \log n))$. If *h-mp-dcp* is based on Dijkstra's algorithm, then the complexity of *h-mp-dcp* is $O((m + n \log n)^2)$.

(b) Let p_{opt} be the optimal solution to the MP-DCP problem (the original problem). Then

(i) $\chi_D(p_H) \geq (\delta(p_{opt}) / \delta(p_H))^{-1/2} \chi_D(p_{opt})$, and

(ii) $\mu(p_{opt}) \leq \mu(p_H)$, $\delta(p_{opt}) \geq \delta(p_H)$. ■

If p_H does not meet the requirements of the applications, we may need a second phase to close or reduce the gap between the heuristic solution and the optimal solution by applying the approximation algorithm with proper approximation factor or the exact algorithm if necessary. On the other hand, the solution obtained by the heuristic algorithm can be used to reduce the computational time of the approximation and exact

algorithms. According to (b) in Theorem 3 we know that $\delta^2(p_H) \leq \delta^2(p_{opt}) \leq U$. So the *Procedure approx-mp-dcp* (resp. *Procedure exact-mp-dcp*) can terminate safely once $upper < \delta^2(p_H)$ (resp. $T_i < \delta^2(p_H)$). Note that using *h-mp-dcp* as an initial pruning step does not affect the polynomial time complexity of these algorithms. The number of invocations of Dijkstra's algorithm in the parametric search can also be greatly reduced using techniques described in [12].

Table 1. Numeric simulation results on two classes of graph topologies

$ V $	OPT	$\Phi(OPT)$	H-OPT	$\Phi(H-OPT)$	Error (%)
1000	0.835	0.7981	0.826	0.7956	0.313
1500	1.043	0.8515	1.036	0.8499	0.188
2000	1.209	0.8867	1.196	0.8842	0.282
2500	1.341	0.9100	1.327	0.9077	0.253
3000	1.456	0.9273	1.437	0.9246	0.291

(a) Regular graph (out degree = 6)

$ V $	OPT	$\Phi(OPT)$	H-OPT	$\Phi(H-OPT)$	Error (%)
1000	0.643	0.7399	0.628	0.7350	0.662
1500	0.526	0.7006	0.515	0.6967	0.557
2000	0.505	0.6932	0.492	0.6886	0.664
2500	0.418	0.6620	0.413	0.6602	0.274
3000	0.459	0.6769	0.459	0.6769	0.000

(b) Waxman's random graph

$|V|$, OPT, $\Phi(OPT)$, H-OPT and $\Phi(H-OPT)$ denote the number of nodes of the network, the optimal $\chi_D(p)$, the optimal $\pi_D(p)$, the solution for $\chi_D(p)$ obtained by Algorithm *h-mp-dcp* and the corresponding $\pi_D(p)$. The Error column is computed as

$$100 (\Phi(OPT) - \Phi(H-OPT)) / \Phi(OPT).$$

We present in Table 1 numerical simulation results for this heuristic. The experiments are carried out on two different classes of graphs: regular graph [13], and Waxman's random graph [14]. In these classes of graphs, for each link (i, j) , $\mu(i, j)$ is randomly independently generated integers uniformly distributed in $[1, 20]$ and $\delta^2(i, j)$ is randomly independently generated integers uniformly distributed in $[1, 200]$. The value of D is 115% of $\mu(p^*)$ where p^* is the s - t path with minimum mean delay. (Now, MP-DCP problem can be seen to be defined on a deterministic network with two independent metrics: mean delay μ and delay variance δ^2). It can be seen that the optimal values and the approximate values of $\pi_D(p)$ are very close.

5. MP-DCP Problem: Case 2

In this section we consider the MP-DCP problem in the case when $\forall p, \mu(p) > D$.

Theorem 4: If $\forall p, \mu(p) > D$, the MP-DCP problem is NP-hard.

Proof: Let us consider an instance of the longest path problem on graph $G(V, E)$. It is known that finding the longest simple path in terms of the number of links is NP-hard and it can also be seen that finding the longest simple path from a given node s to a node t is also NP-hard [15].

To prove the NP hardness of MP-DCP problem in Case 2, it suffices to show that the longest path problem is a subclass of the MP-DCP problem.

Define an MP-DCP problem instance on G with given bound $D = 1$ as follows:

Let $\delta^2(i, j) = 1$ for each link $(i, j) \in E$ (now $\delta^2(p)$ is equal to the number of hops of path p).

Let $M = n / ((1 + 1/n)^{1/2} - 1) = O(n^2)$

Assign the $\mu(i, j)$ on each link $(i, j) \in E$ as follows:

$$\mu(i, j) = \begin{cases} 1 + M, & j = t \\ 1, & \text{otherwise} \end{cases}$$

We next show that the optimal path for the above MP-DCP problem is the longest s - t path in G .

Let p_{opt} and p_l denote the optimal MP-DCP s - t path and a longest s - t path, respectively.

We obtain

$$(1 - \mu(p_l)) / \delta(p_l) \leq (1 - \mu(p_{opt})) / \delta(p_{opt})$$

Assume that $\delta^2(p_{opt}) < \delta^2(p_l) (< n)$.

Then we have the following contradiction.

$$1 + 1/n < \delta^2(p_l) / \delta^2(p_{opt}) \leq ((\mu(p_l) - 1) / (\mu(p_{opt}) - 1))^2 < ((n + M) / M)^2 = 1 + 1/n. \quad \blacksquare$$

Theorem 5: No pseudo polynomial exact algorithm or fully polynomial constant factor approximation algorithm can be obtained for Case 2 of the MP-DCP problem unless $P = NP$.

Proof: According to Theorem 4, the longest path problem is a subclass of the MP-DCP problem with $D = 1$ (Case 2) and thus a pseudo polynomial exact algorithm for this problem, which involves only numbers bounded by polynomial function of n , is also applicable to the longest path problem. This would then contradict the fact that there is no pseudo polynomial algorithm for the longest path problem unless $P = NP$.

If there exists a fully polynomial constant factor

approximation algorithm for the MP-DCP problem for Case 2, then let $\varepsilon < 1$ be the approximation factor, and let p_ε and p_l be the approximate solution to MP-DCP problem and the longest s - t path, respectively. By the definition of approximation factor for maximum problem, we have

$$|\chi_D(p_{opt}) - \chi_D(p_\varepsilon)| / \chi_D(p_{opt}) \leq \varepsilon$$

Hence

$$\chi_D(p_\varepsilon) \geq (1 - \varepsilon) \chi_D(p_{opt})$$

So, $(1 - u(p_\varepsilon)) / \delta(p_\varepsilon) \geq (1 - \varepsilon)(1 - u(p_l)) / \delta(p_l)$.

$$\text{Hence } \delta^2(p_l) / \delta^2(p_\varepsilon) \leq ((1 - \varepsilon)(u(p_l) - 1) / (u(p_\varepsilon) - 1))^2 \leq (1 - \varepsilon)^2 (1 + 1/n) \leq 2(1 - \varepsilon)^2.$$

So, p_ε is a constant factor approximate solution to the longest path problem. This leads to the contradiction of the fact that no constant factor polynomial time approximation algorithm exists for the longest path problem [15]. \blacksquare

The barrier to extend the heuristic algorithm of section 4 is that the optimum value OPT may be negative under the assumption that $\forall p, \mu(p) > D$. Dijkstra's algorithm is not applicable due to the likely presence of negative link weights. So we need to use the Bellman-Ford-Moore algorithm. Even this algorithm will fail if there is a directed circuit of negative weight in the network.

6. Path protection under inaccurate state information

Path protection requires selection of a pair of disjoint paths such that the individual delays of the two paths or the total delay of the two paths is bounded by some specified value [16]. In this section, we only consider the latter case. In this scenario, from a pair of paths, we usually choose the quicker path (with smaller delay) as the primal path and use the slower path as the backup path. We use upper case letter, e.g. P , to denote a pair of disjoint paths and use $P(1)$ and $P(2)$ to denote the two individual paths in P . With accurate state information, Suurballe and Tarjan [17] proposed an efficient algorithm requiring only $O(m + n \log n)$ time for computing a shortest pair of link disjoint s - t paths (We can use link splitting to get node disjoint paths). Most recently, Orda and Sprintson [16] presented approximation algorithms for computing a delay constrained shortest pair of disjoint paths and provided several insightful results. In this section, we shall consider extension of our algorithms developed in the previous sections to the path protection problem under inaccurate state information.

Suurballe and Tarjan's algorithm is given next for the sake of completeness.

Algorithm shortest-link-disjoint-path(s, t)

- Step 1)** Compute a shortest path tree from node s to all the other nodes. Let d_i represent the shortest path length from node s to node i .
- Step 2)** Replace the length of all links $l = (i, j)$ with $w(l) - d_j + d_i$, where $w(l)$ is the original link weight.
- Step 3)** Let p_1 be a shortest s - t path. Reverse the directions of all the links on p_1 without changing the newly computed weights.
- Step 4)** Compute the shortest s - t path in the transformed network.
- Step 5)** Let p_2 be the shortest s - t path obtained in step 4. Discard every link in one path whose reversal appears in the other and group the remaining links into two paths.

When the state information is inaccurate, we need to generalize the MP-DCP problem to the Most Probable Delay Constrained Pair of Disjoint Paths (MP-DCPDP) problem defined below.

MP-DCPDP Problem: Given a maximum total delay requirement D for a pair of disjoint paths and source s and destination $t \neq s$, the MP-DCPDP problem is to find a pair of disjoint s - t paths P^* such that for any pair of disjoint s - t paths P , $\chi_D(P^*) \geq \chi_D(P)$, where $\chi_D(P) = (D - \mu(P)) / \delta(P)$ and $\mu(P) = \mu(P(1)) + \mu(P(2))$, $\delta^2(P) = \delta^2(P(1)) + \delta^2(P(2))$.

In the following, to simplify the writing, cost and delay of a path in the context of the MP-DCPDP problem, refer, respectively, to mean delay (μ) and mean variance (δ^2) of the path or pair of paths, i.e., we call $\mu(p)$ (*resp.* $\mu(P)$) and $\delta^2(p)$ (*resp.* $\delta^2(P)$) as the cost and delay of path p (*resp.* a pair of paths P), respectively.

In this section, we only consider the case that there exists a disjoint pair of paths P such that $\mu(P) \leq D$ (corresponding to Case 1 for the single path version). Notice that the MP-DCPDP problem defined here only consider the total cost and total delay of the two disjoint paths. We don't consider the individual delay of the disjoint paths as in [18], [19] since this problem is much more intractable.

We first consider the heuristic Algorithm *h-mp-dcp* presented in section 4 for extension to the MP-DCPDP problem. Define

$$D(P) = (D - \mu(P)) / \delta^2(P) \text{ and}$$

$$OPT = \max \{H\chi D(P) \mid P \text{ is a pair of disjoint } s\text{-}t \text{ paths}\}.$$

As in the single path case, if OPT is unknown, we can still compute the path using the same algorithm as if OPT were known, but we need an oracle test to decide the relationship ($<$, $=$, $>$) between OPT and any given value λ in computing the two shortest paths in the Algorithm shortest-link-disjoint-path. The oracle test procedure is based on Algorithm shortest-link-disjoint-path on the link

weight $\mu(i, j) + \lambda \delta^2(i, j)$ for all links $(i, j) \in E$. The oracle test procedure works in exactly the same way as in the single path version.

Since we are not aware of any pseudo-polynomial algorithm for the delay constrained shortest pair of disjoint paths problem (which requires identifying a shortest pair of disjoint paths with the total delay of the two paths bounded by a specified value) we do not attempt to extend our algorithms of sections 2 and 3 for the MP-DCPDP problem on general networks.

6.1. Algorithms on DAG

Even on DAG determining a delay constrained shortest pair of disjoint paths is NP-hard, since adding a link with zero cost and delay from source node s to destination node t reduces the CSP problem on DAG to delay constrained shortest pair of disjoint paths on DAG. The key issue in designing exact and approximation algorithms on DAG is first to find a pseudo polynomial algorithm for the Constrained Shortest Pair of Node Disjoint Paths (CSPNDP-DAG) problem which requires identifying a shortest pair of node disjoint paths with the total delay of the two paths bounded by a specified value. Link disjoint path algorithm can be designed similarly. We can then derive exact and approximation algorithms for the MP-DCPDP-DAG problem by replacing the CSP algorithm with the CSPNDP-DAG algorithm in the algorithms of sections 2 and 3.

Our exact algorithm for the CSPNDP-DAG problem is the extension of the methods first proposed by Perl and Shiloach [19] and enhanced by Li, et al. [18] and Eppstein [20]. It consists of two phases. The first phase is the DAG transformation.

6.2. DAG transformation

Given a DAG $G = (V, E)$, we construct a larger DAG $G^* = (V^*, E^*)$, such that there is a one to one correspondence between a pair of node disjoint paths in G and a single path in G^* . Assuming that the nodes of G are labeled $1, 2, \dots, n = |V|$ in topological order such that if $(u, v) \in E$, then $u < v$. Without loss of generality, assume $(s, t) \notin E$ (we can introduce a dummy node between link (s, t) if the assumption is violated). The construction of G^* is as follows:

$$V^* = \{(u, v) \mid u, v \in V \text{ and}$$

$$u \neq v \text{ unless } u = v = s \text{ or } u = v = t\}$$

$$E^* = \{((u, v), (u, w)) \mid (v, w) \in E \text{ and}$$

$$v \leq u\} \cup \{((v, u), (w, u)) \mid (v, w) \in E \text{ and } v \leq u\}$$

Following the argument in [18] and [19] with straight forward extension, it can be seen that there are two node

disjoint s-t paths in G with total cost C and total delay T if and only if there exists a directed path from (s, s) to (t, t) in G^* with cost C and delay T . On the other hand, the corresponding node disjoint pair of paths in G can be constructed from the (s, s) - (t, t) path in G^* [18]-[20].

Now, the CSPNDP-DAG problem is equivalent to finding the minimal cost path from (s, s) to (t, t) with delay bounded by a specified quantity T . It can be seen that $|V^*| = O(n^2)$ and $|E^*| = O(mn)$. Then we can apply the second phase to complete the algorithm. The second phase is just to apply the CSP algorithms given in section 2 on G^* to compute the constrained shortest path with delay bounded by T .

Theorem 6: CSPNDP-DAG problem is solvable in $O(Tmn)$ steps. ■

The ε -approximation algorithm for the CSPNDP-DAG problem can be derived as in section 3.

7. Summary

We have studied the stochastic shortest path problem aimed at identifying the most probable delay constrained path (MP-DCP problem). Our work is based on the formulation given in [9]. The work in [9] focused on developing heuristic approaches using the Lagrangian relaxation or line search techniques. In contrast, our focus has been on developing polynomial time ε -approximation and heuristic algorithms. For the case (Case 1) when there is a path whose mean delay is less than or equal to the specified delay bound D , we have presented an exact algorithm of pseudo polynomial time complexity, a fully polynomial time ε -approximation algorithm and a strongly polynomial time heuristic algorithm. In the unlikely case (Case 2) when every path violates this assumption we have shown that the problem is NP-hard. We have also shown that for this case no pseudo polynomial time exact algorithm or fully polynomial time constant factor approximation algorithm is possible unless $P = NP$. The difficulty in this case arises because we need to find a path minimizing one path metric and maximizing another path metric simultaneously. See [9].

We have considered application of our results to the path protection problem under inaccurate state information. For this problem we have shown how to develop a strongly polynomial time heuristic algorithm. We have also shown for this problem how to develop exact and ε -approximation algorithms for DAGs.

Acknowledgment: The works of K. Thulasiraman and Guoliang Xue have been supported by NSF ITR grants ANI-0312435 and ANI-0312635, respectively.

REFERENCES

- [1] A. Goel, K. G. Ramakrishnan, D. Kataria and D. Logothetis, "Efficient computation of delay-sensitive routes from one source to all destinations," *IEEE INFOCOM*, 2001.
- [2] R.Hassin, "Approximation schemes for the restricted shortest path problem," *Math. of Oper. Res.*, 17(1), 1992, pp.36-42.
- [3] A. Jüttner, B. Szviatovszki, I. Mécs and Z. Rajkó, "Lagrange relaxation based method for the QoS routing problem," *IEEE INFOCOM-2001*, pp. 859-868.
- [4] D. Lorenz and D. Raz, "A simple efficient approximation scheme for the restricted shortest paths problem," *Oper. Res. Lett.*, vol. 28, pp. 213-219, June 2001.
- [5] K. Mehlhorn and M. Ziegelmann, "Resource constrained shortest paths," *Proc. 8th European Symposium on Algorithms (ESA2000)*.
- [6] Y. Xiao, K. Thulasiraman and G. Xue, "Equivalence, unification and generality of two approaches to the constrained shortest path problem with extension", Allerton conference, Oct., 2003, Urbana-Champaign, IL.
- [7] G. Xue, "Minimum-cost QoS multicast and unicast routing in communication networks," *IEEE Trans. on Comm.*, vol. 51, no.5, May 2003, pp.817-824.
- [8] R. Guerin and A. Orda, "QoS routing in networks with inaccurate information: theory and algorithms," *IEEE/ACM Trans. on Networking*, vol. 7, pp. 350-364, June 1999.
- [9] T. Korkmaz and M. Krunz, "Bandwidth-delay constrained path selection under inaccurate state information," *IEEE/ACM Trans. on Networking*, June 2003.
- [10] D. H. Lorenz and A. Orda, "QoS routing in networks with uncertain parameters," *IEEE/ACM Trans. Networking*, vol. 6, pp. 768-778, Dec. 1998.
- [11] N. Megiddo, "Combinatorial optimization with rational objective functions," *Math. of Oper. Res.*, vol. 4, no. 4, Nov. 1979.
- [12] T. Radzik, Fractional combinatorial optimization, in handbook of combinatorial optimization, Editors DingZhu Du and Panos Pardalos, vol. 1, Kluwer Academic Publishes, Dec. 1998.
- [13] K. Thulasiraman and M. N. Swamy, *Graphs: Theory and Algorithms*, Wiley Interscience, New York, 1992
- [14] M. Waxman, "Routing of multipoint connections," *IEEE J. Selected Areas in Comm.* 6(9), Dec. 1988.
- [15] D. Karger, R. Motwani and G. D. S. Ramkumar, "On approximating the longest path in a graph", *Algorithmica* 18, pp. 82-98, 1997.
- [16] A. Orda and A. Sprintson, "Efficient algorithms for computing disjoint QoS paths," *IEEE INFOCOM*, 2004.
- [17] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, pp. 325-336, 1984.
- [18] C.-L. Li, S. T. McCormick, and D. Simchi-Levi, "The complexity of finding two disjoint paths with min-max objective function," *Discrete Applied Math.* 26, 1990, pp. 105-115.
- [19] Y. Perl and Y. Shiloach, "Finding two disjoint paths between two pairs of vertices in a graph," *J. ACM* 25, 1978, pp. 105-115.
- [20] D. Eppstein, "Finding common ancestors and disjoint paths in DAGs," Tech. Report 95-52, Dec. 1995.