Finding All Hops k-Shortest Paths

Gang Cheng and Nirwan Ansari

Advanced Networking Lab
Department of Electrical and Computer Engineering
New Jersey Institute of Technology,
Newark, NJ 07102, U.S.A
{gc2, Ansari}@njit.edu

Abstract— **In this paper, we introduce and investigate a new problem referred to as the All Hops k-shortest Paths (AHKP) problem. An efficient solution, the All Hops k-shortest Paths Extended Bellman-Ford (AHKPEB) algorithm, is proposed. Especially, when $k=1$, AHKPEB is an optimal comparison-based solution to the All Hops Optimal Path (AHOP) problem in terms of the worst-case computational complexity, i.e., it is impossible to find another comparison-based solution to Add-AHOP having the worst-case computational complexity lower than that of AHKPEB.**

Index Terms— **Quality of Service (QoS), Bellman-Ford algorithm, All Hops k-shortest Paths Problem (AHKP).**

I. Introduction

One of the challenging issues for high-speed packet switching networks to facilitate various applications is to select feasible paths that satisfy different quality-of-service (QoS) requirements. This problem is known as QoS routing. However, it has been proved that multiple additively constrained QoS routing is NP-complete [1]. Many proposed source routing algorithms tackle this problem by transforming it into the shortest path selection problem or the k-shortest paths selection problem, which are P-complete, with an integrated cost function that maps the multi-constraints of each link into a single cost. However, since the solutions are computed by finding the shortest path, one of their common problems is that they cannot minimize the number of hops of their solutions. As a result, the network resource is wasted. Given a set of constraints $(\alpha_1,\alpha_2,...,\alpha_M)$ and a network that is modeled as a directed graph $G(N,E)$, where N is the set of all nodes and E is the set of all links, assume each link connected from node u to v, denoted by $e_{u,v}=(u,v)\in E$, is associated with M randomly distributed additive parameters: $w_i(u,v)\ge 0$, $i=1,2,...,M$, and define $P_r\{W_1(p)\le\alpha_1, W_2(p)\le\alpha_2, ..., W_M(p)\le\alpha_M \mid C(p)=u,\ H(p)=n\}$ as the probability that a path p is a feasible path with $C(p)=u$, and its hop count, $H(p)=n$, where $C(p)$ is the cost of p, which is a function of the weights of the links on p, and $W_i(p)=\sum_{e_{u,v}\in p} w_i(u,v)$. The probability of the shortest path to be a feasible path may not be the largest in all paths. Therefore, computing a feasible path among all hops shortest paths, instead of only the shortest path, can increase the success ratio of finding a feasible path. In this paper, we introduce and investigate a new problem referred to as all hops k-shortest paths (AHKP) problem, defined below.

Definition 1: All Hops k-shortest Paths (AHKP) Problem: Assume a network is modeled as a directed graph $G(N,E)$, where N is the set of all nodes and E is the set of all links. Each link connected from node u to v, denoted by $e_{u,v}=(u,v)\in E$, is associated with an additive weight $c(u,v)$. Given a source node $s\in N$ and maximal hop count H, $H<n$, find, for each hop count value h, $1\le h\le H$, and a destination node $u\in N$, the k-shortest h-hop constrained paths from s to u. In this paper, we will refer to the length of a path as the sum of its link weights.

We propose an efficient solution, the All Hops k-shortest Paths Extended Bellman-Ford (AHKPEB) algorithm, to AHKP and prove that, when $k=1$, AHKPEB is an optimal comparison-based solution to Add-AHOP in terms of the worst-case computational complexity, i.e., it is impossible to find another comparison-based solution having the worst-case computational complexity lower than that of AHKPEB. The definitions of comparison-based algorithms and Add-AHOP can be found in [2] and [3], respectively.

II. A Solution To AHKP

The relaxation procedure of our proposed AHKPEB algorithm is shown in Fig. 1. Here, $p_1^h(s,i)$, $p_2^h(s,i)$,…, $p_k^h(s,i)$ represent k h-hop paths from source s to i computed by AHKPEB (we will prove later that $p_1^h(s,i)$, $p_2^h(s,i)$,…, $p_k^h(s,i)$ are h-hop k-shortest paths among all the h-hop paths from source s to i. If, in reality, the total number of h-hop paths from s to i is less than k, we assume that there exist virtual h-hop paths whose costs are infinity). $D_{i,1}^h$, $D_{i,2}^h$, …, $D_{i,k}^h$ are their costs, and $c(i,j)$ is the weight of link $e(i,j)$. Assume there exists a virtual link

This work has been supported in part by the New Jersey Commission on Higher Education via the NJ I-TOWER project, and the New Jersey Commission on Science and Technology via the NJ Center for Wireless Telecommunications.

0-7803-7978-0/03/$17.00 ©2003

$\hat{e}(s,i)$ between the source node s and any other node i, whose cost is infinity (in reality, it does not exist). $\forall i \in N$, $p_1^1(s,i) = e(s,i)$ and $p_g^1(s,i) = \hat{e}(s,i)$, $g = 2,3,...,k$, (if, in reality, no link between the source s and node i exists, $p_1^1(s,i) = \hat{e}(s,i)$). Set $D_{i,1}^1$ as $c(s,i)$ and $D_{i,g}^h$ ($g = 2,3,...,k$) as infinity. The initial values of $D_{i,1}^h$, $D_{i,2}^h$, …, $D_{i,k}^h$ ($h \geq 2$) are infinity.

```
Relax(j,i)
set m = 1, n = 1
for l := 1 to k
    P̂_l = p_l^{h+1}(s,i)   /* P̂_l is a temporary path */
    β̂_l = D_{i,l}^{h+1}     /* β̂_l is a temporary variable */
    if D_{i,m}^{h+1} ≥ D_{j,n}^h + c(i,j) then
        D_{i,l}^{h+1} = D_{j,n}^h + c(i,j)
        p_l^{h+1}(s,i) = p_n^h(s,j) + e(i,j)
        n = n + 1
    else
        D_{i,l}^{h+1} = β̂_m
        p_l^{h+1}(s,i) = P̂_m
        m = m + 1
    end if
end for
```

Fig. 1. The relaxation procedure of AHKPEB

We will prove later that $D_{i,1}^h \leq D_{i,2}^h \leq ... \leq D_{i,k}^h$, i.e., $p_1^h(s,i)$, $p_2^h(s,i)$,…, $p_k^h(s,i)$ are sorted in increasing order of their lengths. The relaxation procedure of AHKPEB is illustrated by the example shown in Fig. 2.

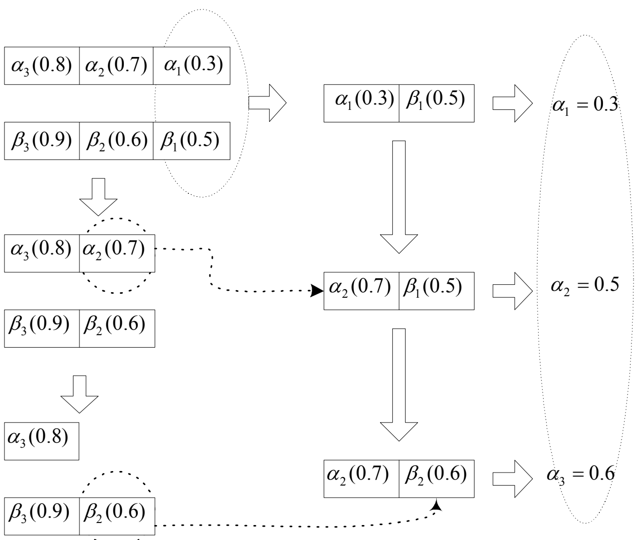

Fig. 2. An illustration of the relaxation procedure of AHKPEB.

Given two sorted path sets ($k = 3$), $\{\alpha_1, \alpha_2, \alpha_3\}$ and $\{\beta_1, \beta_2, \beta_3\}$, whose costs are $\{0.3, 0.7, 0.8\}$ and $\{0.5, 0.6, 0.9\}$, respectively. Note that

$$\min\{\min_{j=1,2,3}\{C(\alpha_j)\}, \min_{j=1,2,3}\{C(\beta_j)\}\} = \min\{C(\alpha_1), C(\beta_1)\}. \quad (1)$$

Let $\boldsymbol{\varphi}_1 = \{\alpha_1, \beta_1\}$; the least cost path in the two sets is the least cost path in $\boldsymbol{\varphi}_1$. In this example, it is α_1. Furthermore, since the two path sets are sorted by their costs and α_1 is the least cost path, the second least cost path in the two sets must be the least cost path between α_2 and β_1, i.e., let $\boldsymbol{\varphi}_2 = (\boldsymbol{\varphi}_1 \cap \overline{\{\alpha_1\}}) \cup \{\alpha_2\}$, the second least cost path in the two sets is the least cost path in $\boldsymbol{\varphi}_2$. Similarly, the j th least cost path in the two sets is the least cost path in $\boldsymbol{\varphi}_j$, which can be proved by deduction, where $\boldsymbol{\varphi}_j = (\boldsymbol{\varphi}_{j-1} \cap \overline{\{\pi_{j-1}\}}) \cup \{\nu_j\}$, π_{j-1} is the least cost path in $\boldsymbol{\varphi}_{j-1}$ and ν_j is the next path to π_{j-1} in the corresponding set. Moreover, $1 \leq j < \nu \leq k$, $C(\pi_j) \leq C(\pi_\nu)$, i.e., the paths (π_j) are sequentially computed in the increasing order of their costs. Following the relaxation procedure, it can be observed that the 3-shortest paths of the two sets are obtained in increasing order of their lengths. Hence, given two sorted sets of k paths in increasing order of their lengths, the outputs of the relaxation procedure of AHKPEB are the sorted k shortest paths of the two sets. Therefore, as shown in Fig. 3, if $\forall d \in \{1,2,...,d_i\}$, $p_1^h(s,i_d)$, $p_2^h(s,i_d)$, …, $p_k^h(s,i_d)$ are sorted in increasing order of their costs, the k-shortest $(h+1)$-hop paths among the paths $p_g^h(s,i_d) + e(i_d,i)$, $d = 1,2,...,d_i$, $g = 1,2,...,k$, are iteratively computed resulting in $p_1^{h+1}(s,i)$, $p_2^{h+1}(s,i)$,…, $p_k^{h+1}(s,i)$, where d_i denotes the degree of node i and i_1, i_2,…, i_{d_i} are its neighboring nodes. Furthermore, $D_{i,1}^{h+1} \leq D_{i,2}^{h+1} \leq ... \leq D_{i,k}^{h+1}$.

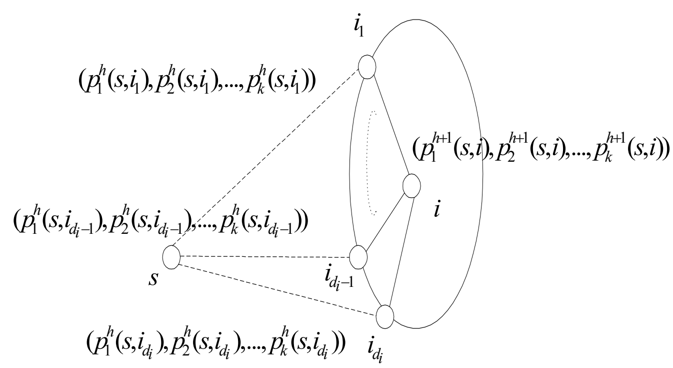

Fig. 3. The AHKPEB algorithm

[Proposition] $p_1^h(s,i)$, $p_2^h(s,i)$,…, $p_k^h(s,i)$ are h-hop k shortest paths among the all h-hop paths from source s to node i and $D_{i,1}^h \leq D_{i,2}^h \leq ... \leq D_{i,k}^h$.

Proof: When $h=1$, from the definition of the initial values of $D_{i,g}^{1}$ ($i \neq s$), $p_g^1(s,i)$, $g=1,2,...,k$, are the one hop k-shortest paths from s to i. Moreover, it can be observed that $D_{i,1}^{1} \leq D_{i,2}^{1} \leq ... \leq D_{i,k}^{1}$.

We assume that the proposition is correct for $n=m$. We shall prove by deduction that it is true for $n=m+1$.

Assume when $n=m+1$, if $\exists j \neq s$, $p_g^{m+1}(s,i)$, $1 \leq g \leq k$, is not one of the k-shortest paths in all $(m+1)$-hop paths from s to i ($D_{i,g}^{m+1}$ is larger than the cost of any $(m+1)$-hop k-shortest path from s to i). Further assume that path $\hat{p}^{m+1}(s,i)$ is not one of $p_g^{m+1}(s,i)$, $g=1,2,...,k$, and has smaller length than that of $p_g^{m+1}(s,i)$ in all $(m+1)$-hop paths from s to i. The predecessor node of node i in $\hat{p}^{m+1}(s,i)$ is d, the path from s to d in $\hat{p}^{m+1}(s,i)$ is $\hat{p}^{m}(s,d)$ (note that $\hat{p}^{m}(s,d)$ may not be one of the k-shortest m-hop paths from s to d, and by the earlier assumption, $p_g^m(s,d)$, $g=1,2,...,k$, generated by AHKPEB are the k-shortest m-hop paths from s to d), the cost of $\hat{p}^{m+1}(s,i)$ is c, and the cost of $\hat{p}^{m}(s,d)$ is c'. Thus,

$$c < D_{i,g}^{m+1}. \tag{2}$$

If $\hat{p}^{m}(s,d)$ is one of $p_g^m(s,d)$, $g=1,2,...,k$, or $\exists g \in \{1,2,...,k\}$ such that $c' = D_{d,g}^{m}$, $\hat{p}^{m+1}(s,j)$ is resulted by concatenating $\hat{p}^{m}(s,d)$ with link $e(j,d)$, i.e., $\hat{p}^{m+1}(s,i)$ is one of $p_g^m(s,i_d)+e(i_d,i),\ d=1,2,...,d_i,\ g=1,2,...,k$. Since $\hat{p}^{m+1}(s,i)$ is not one of $p_g^{m+1}(s,i)$, $g=1,2,...,k$, and $p_g^{m+1}(s,i)$, $g=1,2,...,k$, are the k-shortest paths of $p_g^m(s,i_d)+e(i_d,i),$ $d=1,2,...,d_i,$ $g=1,2,...,k$, $\forall g \in \{1,2,...,k\}$,

$$c = D_d^m + c(i,d) \geq D_{i,g}^{m+1}, \tag{3}$$

which contradicts (1). Hence, $\hat{p}^{m}(s,d)$ is not one of $p_g^m(s,d)$, i.e., $\forall g \in \{1,2,...,k\}$,

$$c' \geq D_{d,g}^{m}. \tag{4}$$

So, $\forall g,u \in \{1,2,...,k\}$, the cost of $\hat{p}^{m+1}(s,i)$ is

$$c = c' + c(j,d) \geq D_{d,g}^{m} + c(j,d) \geq D_{i,g}^{m+1}, \tag{5}$$

which contradicts (2). So, when $n=m+1$, $\forall i \in \{1,2,...,N\}$, $p_g^{m+1}(s,i)$, $g=1,2,...,k$, are the k-shortest paths in all $(m+1)$-hop paths from s to i. Moreover, since $p_g^m(s,i_d)+e(i_d,i),\ g=1,2,...,k$, is in increasing order of their lengths for any $d \in \{1,2,...,d_i\}$, $p_g^{m+1}(s,i)$, $g=1,2,...,k$, is also in increasing order of their lengths, i.e., $D_{i,1}^{m} \leq D_{i,2}^{m} \leq ... \leq D_{i,k}^{m}$.

Thus, for any node $i \in \{1,2,...,N\}$, $p_g^m(s,i)$, $g=1,2,...,k$, generated by AHKPEB must be the k-shortest paths in all m-hop paths from s to i. ■

Computational Complexity: Since the computational complexity in each relaxation procedure is k, the computational complexity of AHKPEB is $O(kHE)$, where E is the number of links. When $k=1$, the computational complexity of AHKPEB is $O(HE)$.

Guerin and Orda [3] proved that $O(N^3)$ is a (tight) lower bound on the computational complexity of any comparison-based solution to the Add-AHOP problem (a case of AHOP where link weights are additive). Note that $HE < N^3$ because $E < N^2$ and $H < N$. Hence, AHKPEB ($k=1$) is an optimal comparison-based solution to Add-AHOP.

III. SIMULATIONS

We conduct our simulations in the network topology presented in [4]. To evaluate the effect of AHKPEB on minimizing the number of hops of feasible paths, two QoS algorithms based on the Dijkstra algorithm and AHKPEB ($k=1$) are designed, respectively, both of which are associated with the cost function $f(x,y)=x+y$. In the simulation, the link weights are independent and uniformly distributed from 0 to 1, two QoS constraints are set to be equal. All data are obtained by running 1,000,000 requests. The algorithm that can always locate the least hops feasible path as long as a feasible path exists is refereed to as the optimal algorithm. Here, it is achieved simply by hop-by-hop flooding which can always locate the least hops feasible path as long as it exists. Hence, its average hops are the lower bound of the average hops of all feasible paths. As shown in Fig. 4, the algorithm based on AHKPEB achieves near optimal average hops, i.e., the hops of its solutions are minimized.

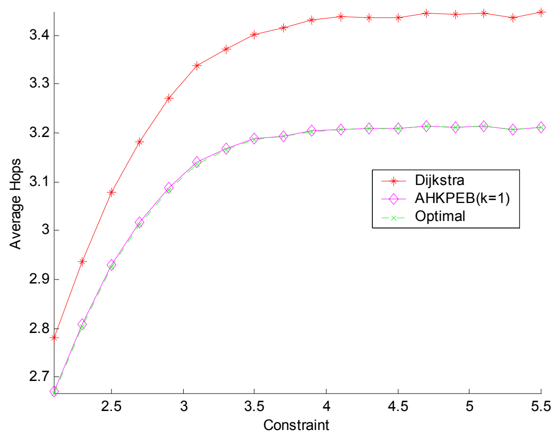

Figure 4. Average hops of the algorithms

Note that AHKPEB ($k=1$) is capable of computing all hops shortest path from a source to a destination. Whenever the algorithm based on the Dijkstra algorithm can find a feasible path, the one based on AHKPEB can also find one. The reverse is not true. As a result, the success ratio (SR) of the algorithm based on AHKPEB must be higher than that of the one based on the Dijkstra algorithm, where SR is defined as follow:

$$SR = \frac{Total\ number\ of\ success\ request\ of\ the\ algorithm}{Total\ number\ of\ success\ request\ of\ the\ optimal\ algorithm}. \quad (6)$$

Note that, in [4] and [5], SR is defined as

$$SR = \frac{Total\ number\ of\ success\ request}{Total\ number\ of\ request}. \quad (7)$$

Since there may not exist a feasible path if the given constraints are tight, in which case, the success ratio (7) cannot truly reflect algorithms' capability in finding a feasible path. Therefore, we adopt (6) as the definition of the success ratio of an algorithm in finding a feasible path, instead of (7). The success ratios of two algorithms are shown in Fig. 5. It can be observed that the SR of the algorithm based on AHKPEB is always higher than that of the algorithm based on Dijkstra algorithm.

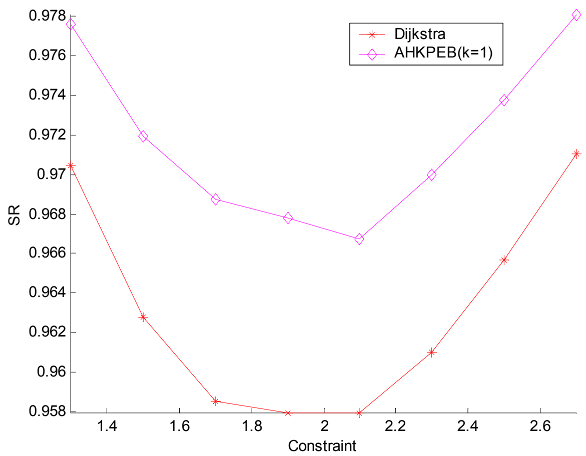

Figure 5. Success ratios in finding a feasible path of algorithms

IV. Conclusions

In this paper, we have introduced and investigated a new problem referred to as the all hops k-shortest paths (AHKP) problem. An efficient solution, the All Hops k-shortest Paths Extended Bellman-Ford (AHKPEB) algorithm, has been proposed. When $k=1$, AHKPEB is an optimal comparison-based solution to the all hops optimal path (AHOP) problem in terms of the worst-case computational complexity, i.e., it is impossible to find another comparison-based solution having the worst-case computational complexity lower than that of AHKPEB.

References

[1] Z. Wang and J. Crowcroft, "Quality of Service routing for supporting multimedia applications," *IEEE Journal on Selected Areas on Communications*, 1996, **14**, (7), vol. 14, pp. 1228-1234.

[2] D.R. Karger, D. Koller, and S. J. Phillips, "Finding the hidden path: Time bounds for all-pairs shortest paths," *SIAM J. Computing*, 1993, vol. 22, pp. 1199-1217.

[3] R. Guerin and A. Orda, "Computing shortest paths for any number of hops," *IEEE/ACM Trans. on Networking*, 2002, **10**, (5), pp. 613-620.

[4] S. Chen and K. Nahrsted, "On finding multi-constrained path," Proceedings of IEEE ICC'98, vol. 2, pp. 874-899, 1998.

[5] T. Korkmaz , M. Krunz, and S. Tragoudas, "An efficient algorithms for finding a path subject to two additive constraints," Proceedings of the ACM SIGMETRICS'2000, pp. 318-327, June 2000.