

# Precomputation for Finding Paths with Two Additive Weights<sup>\*</sup>

Yong Cui, Ke Xu, Jianping Wu, Mingwei Xu

Department of Computer Science, Tsinghua University, Beijing, P.R.China, 100084

{cy, xuke}@csnet1.cs.tsinghua.edu.cn; jianping@cernet.edu.cn; xmw@csnet1.cs.tsinghua.edu.cn

**Abstract**—As the most challenging problems of the upcoming next-generation networks, 2-constrained quality-of-service routing (QoSR) is NP-complete problem, for which we propose a novel precomputation algorithm, LEFPA. This algorithm converts two additive weights to a single metric with linear energy functions (LEFs) and pre-computes QoS routing table with multiple ( $B$ ) LEFs to further enhance its scalability. We first analyze the performance of LEFs and give a method to determine the feasible and unfeasible areas in the metric space for a QoS request. We then introduce the proposed LEFPA, whose computation complexity is  $O(B(m+n\log n))$ . Furthermore, we use three methods to evaluate the routing performance. Extensive simulations show that our LEFPA has both absolutely and competitively high performance.

**Keywords**—Linear energy function, QoS routing, scalability, precomputation, performance evaluation

## I. INTRODUCTION

Providing different quality-of-services (QoS) for different applications in the Internet is a challenging issue [1], of which QoS Routing (QoSR) is the most pivotal problem [2]. The main function of QoSR is to find a feasible path that satisfies multiple constraints for QoS applications.

For the NP-completeness [3], [4] of multi-constrained QoSR, many heuristics have been proposed. However, these algorithms have three limitations [2]: (1) High time complexity prohibits their applications. (2) Low performance means that these algorithms sometimes cannot find a feasible path even when it does exist. (3) Some algorithms work only for a specific network. Furthermore, in next-generation high-speed networks, a high rate of packet arrival prohibits the online computation scheme, to which most heuristics belong. Based on the analyses of linear energy function (LEF), we propose a novel approach with the name LEF based precomputation algorithm (LEFPA). Using precomputation with LEFs of two dimensions, LEFPA can deal with not only 2-constrained problems, but also 1-constrained optimal problems, such as the Delay-Constrained Least-Cost (DCLC) problem. Both the theoretical analysis and experimental results show that our easily implemented LEFPA is highly scalable and has high performance. Furthermore, LEFPA is consistent with the routing architecture of the current Internet. LEFPA can serve the next-generation high-speed networks.

This paper has three contributions: (1) We give a mathematical model in the QoS metric space to decide whether a QoS request can be determined by the continuous change of LEFs. (2) We propose the precomputation algorithm LEFPA for QoSR problem based on LEFs. (3) We

propose a novel evaluation method for QoSR algorithms, named the unknown-area proportion method.

The rest of this paper is organized as follows. In Part II we analyze the relation of LEFs and constraint space. LEFPA is proposed in Part III, and extensive simulations show its performance in Part IV. Finally, conclusions appear in Part V.

## II. LINEAR ENERGY FUNCTION ANALYSIS

### A. Problem Formulation

A directed graph  $G(V, E)$  presents a network.  $V$  is the node set and the element  $v \in V$  is called a node representing a router.  $E$  is the set of edges representing links that connect the routers. The element  $e_{ij} \in E$  represents the edge  $e = v_i \rightarrow v_j$  of  $G$ . In QoS, each link has a group of independent weight  $w(e) = (w_1(e), w_2(e))$ , which is also called QoS metric. For a path  $p = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n$  and  $l \in \{1, 2\}$ , the weight  $w_l(e) \in R^+$  satisfies the additive character if  $w_l(p) = \sum_{i=1}^n w_l(v_{i-1} \rightarrow v_i)$ .

#### Definition 1. Multi-Constrained Path

For a given graph  $G(V, E)$ , source node  $s$ , destination node  $t$  and constraint vector  $c = (c_1, c_2)$ , the path  $p$  from  $s$  to  $t$  is called Multi-Constrained Path, if  $w_1(p) \leq c_1$  and  $w_2(p) \leq c_2$ . We write  $w(p) \leq c$  in brief.

Note:  $w(e)$  and  $c$  are also 2-dimensional vectors. For a given QoS request and its constraints  $c$ , QoS seeks to find the path  $p$  satisfying  $w(p) \leq c$  based on the network-state information.

### B. Linear Energy Function

Dijkstra gave the Shortest Path Tree (SPT) algorithm, which has a low time complexity. However, QoSR problem is related to multiple weights simultaneously. Thus the problem is changed to the one, in which the complexity is NP-complete, and the original Dijkstra's algorithm cannot be used to solve it, so we convert the multiple weights to a single weight.

#### Definition 2. Linear Energy Function (LEF) $g_a$

The LEF of link  $e$  is defined as the linear function  $g_a(e) = a_1 w_1 + a_2 w_2$ , which represents the "cost" of  $e$ . Here, the coefficients  $a_1, a_2 \in [0, 1]$  are independent of  $e$  and satisfy  $a_1 + a_2 = 1$ . The vector  $a = (a_1, a_2)$  that satisfies the above condition is called energy coefficient.

Based on LEF, we can convert the original multi-constrained path problem to a least-energy path problem. Each coefficient of LEF represents the importance of the

<sup>\*</sup> Supported by: (1) the National Natural Science Foundation of China (No. 90104002; No. 69725003); (2) the National High Technology Research and Development Plan of China (No. 2002AA103067).

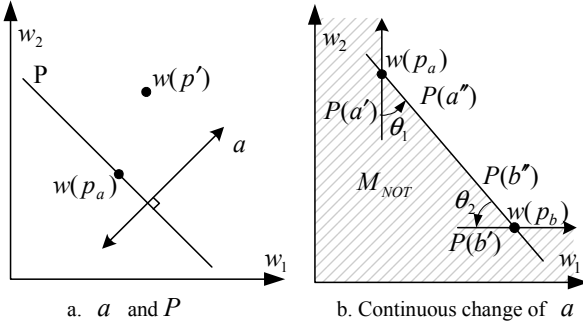


Fig. 1. Relation between paths and lines

corresponding weight to that of other weights when calculating the SPT.

**Theorem 1:** Dijkstra's algorithm with the keyword  $g_a(e)$  can create a least-energy tree  $T_a$  rooted by  $s$ . A path  $p_a$  from  $s$  to an arbitrary node  $t$  along the tree  $T_a$  can satisfy  $g_a(p_a) = \min_{p(s,t) \in G} g_a(p(s,t))$ .

*Proof:* Because  $g_a(e)$  is a linear function, which satisfies  $g_a(e_1 + e_2) = \sum_{l=1}^2 a_l w_l(e_1 + e_2) = \sum_{l=1}^2 a_l w_l(e_1) + \sum_{l=1}^2 a_l w_l(e_2) = g_a(e_1) + g_a(e_2)$ , we can calculate  $g_a(e)$  for each link  $e$ , and then run the Dijkstra's algorithm with  $g_a(e)$  being the keyword instead of the original cost. Because the original Dijkstra's algorithm can guarantee that the path from  $s$  to an arbitrary node  $t$  along the tree has the least cost, the algorithm that uses  $g_a(e)$  as the keyword can guarantee that  $p_a$  is the least-energy path from  $s$  to  $t$ , viz.  $g_a(p_a) = \min_{p(s,t) \in G} g_a(p(s,t))$ .

**Definition 3. Path-weight space**

$W^2 = W_1 \times W_2$  is called the path-weight space, if  $w_l(p) \in W_l$  for any  $p \in G$  and  $l \in \{1, 2\}$ .

For the common condition of  $w_l(e) \in R^+$ , we can take  $W_l = R^+$ , so  $w_l(p) \in W_l$  for any path  $p$ . Thus,  $w(p)$  is a point in the space  $W^2$ , viz.  $w(p) \in W^2$ , and the above set  $\{w(p_a) | \forall a\}$  is a point set in  $W^2$ .

**Theorem 2:** For a given  $G$ , a source-destination pair  $(s, t)$ , energy coefficient  $a = (a_1, a_2)$  and shortest path  $p_a$ , an arbitrary path  $p$  from  $s$  to  $t$  must be on the upside of the line  $P = \{w | a_1 w_1 + a_2 w_2 = g_{opt}\}$  in space  $W^2$ .

*Proof:* We use the reduction to absurdity. If a point  $w(p')$  exists on the downside of line  $P$ , we have  $g_{opt} = a_1 w_1(p_a) + a_2 w_2(p_a) > a_1 w_1(p') + a_2 w_2(p')$ , which is contrary to  $g(p_a) = \min_{p(s,t) \in G} g(p(s,t))$  in Theorem 1. Thus, if there is any path  $p'$  from  $s$  to  $t$ ,  $w(p')$  must be on the upside of the line  $P$  in space  $W^2$ .

For example, for a given vector  $a$  shown in Fig. 1.a, we use Dijkstra's algorithm to find the shortest path  $p_a$  from  $s$

to  $t$  w.r.t.  $g_a$ . Drawing the perpendicular  $P$  of vector  $a$  crossing the point  $w(p_a)$ , we get a partition of space  $W^2$ : all points  $w(p')$  of paths  $p'$  from  $s$  to  $t$  must be on the upside of  $P$ . We should note that, because of the discreteness of network topology graph, the corresponding  $p_a$  is changed discretely in the space  $W^2$  with the continuous change of vector  $a$ . Thus, the mapping  $a \mapsto p_a$  is not an injection. For the common sense where multiple least-energy points exist, we should consider that multiple vectors  $a$  map a single point  $p_a$  as shown in Fig. 1.b. When the vector changes from  $a'$  to  $a''$  continuously, the least-energy point keeps the same value, i.e.  $p_a$ . It is the same case that when the vector changes from  $b'$  to  $b''$ , the corresponding least-energy point is  $p_a$  without change. However, the vector  $a''$  that is equal to  $b'$  has two least-energy paths, i.e.  $p_a$  and  $p_b$ . In this case, a discrete change of least-energy paths occurs.

**C. Feasible area analysis of  $W^2$**

We give a partition of the path-weight space  $W^2$ , including unfeasible area  $M_{NOT}$ , feasible area  $M_{FEASIBLE}$  and unknown area  $M_{UNKNOWN}$ . For a given QoS request, we can judge its feasibility and give a feasible path for a feasible request.

**Definition 4. Unfeasible area**

The point set  $M(a) = \{w | w \in W^2\}$ , where  $w$  is in the lower side of  $P(a)$ , is called unfeasible area determined by a given vector  $a$ .  $M_{NOT} = \bigcup_{a_1+a_2=1, a \geq 0} M(a)$  is called the unfeasible area.

**Theorem 3:** If the constraints  $c \in M_{NOT}$  of a QoS request from  $s$  to  $t$ , a feasible path  $p$  satisfying  $w(p) \leq c$  does not exist.

*Proof:* According to the definition of  $M_{NOT}$ ,  $M_{NOT}$  is the union of all  $M(a)$  with the continuous change of vector  $a$ . For a given QoS request  $c \in M_{NOT}$ , there must be a line  $P(a)$  according to the vector  $a$ , so that  $c$  is on the downside of  $P(a)$ . According to Theorem 2, any path  $p'$  from  $s$  to  $t$  must be the upside of  $P(a)$ . Therefore, there is no feasible path  $p$  that satisfies  $w(p) \leq c$ .

**Definition 5. Available area**

The point set  $M_{AVL} = \overline{M_{NOT}} = \overline{\bigcup_{a_1+a_2=1, a \geq 0} M(a)}$  is called available area in space  $W^2$ , where the complement of  $M_{NOT}$  is  $\overline{M_{NOT}} = W^2 \setminus M_{NOT}$ .

For example, when there are multiple least-energy paths with the different energy functions, the available area  $M_{AVL}$  must be a convex set and each vertex of  $M_{AVL}$  must be a

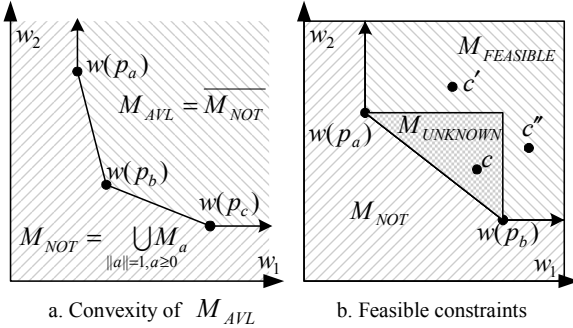


Fig. 2.  $M_{FEASIBLE}$ ,  $M_{AVL}$ ,  $M_{UNKNOWN}$  and  $M_{NOT}$

least-energy path as shown in Fig. 2a.

We further divide  $M_{AVL}$  into two parts: feasible area  $M_{FEASIBLE}$ , and unknown area  $M_{UNKNOWN}$ .

**Definition 6.** Feasible area  $M_{FEASIBLE}$

$M_{FEASIBLE} = \{p \mid \exists a, p \geq p_a\}$  is called feasible area.

**Definition 7.** Unknown area  $M_{UNKNOWN}$

$M_{UNKNOWN} = M_{AVL} - M_{FEASIBLE}$  is called unknown area.

**Theorem 4:** For the arbitrary constraints  $c$  of a QoS request, if  $c \in M_{FEASIBLE}$ , there must be a feasible path.

*Proof:* From the definition of  $M_{FEASIBLE}$ ,  $\exists a$  that satisfies  $p_a \leq c$ , where  $p_a$  is a feasible path.

Therefore, we divide  $W^2$  into 3 parts:  $M_{NOT}$ ,  $M_{UNKNOWN}$  and  $M_{FEASIBLE}$ . Fig. 2.b shows the relation between them. Although some papers have pointed out that LEFs cannot guarantee to find a feasible path even when it does exist [5], [6], they have not given a clear mathematical model. We here use the computational geometry to describe the area ( $M_{UNKNOWN}$ ) successfully, in which the constraint cannot be judged feasible by LEFs.

### III. PROPOSED HEURISTIC LEFPA

#### A. The Idea of LEFPA

Based on the above theory, we can precompute the set  $\{p_a\}$  of least-energy paths with the energy coefficient  $a$  changing continuously over all possible vectors. This path set  $\{p_a\}$  is the QoS routing table. Thus, the computation of the routing table can be independent of QoS requests. When the QoS request arrives, we only need to look up a feasible path in the routing table. According to the position of the QoS constraint in space  $W^2$ , there are three possibilities. (1) We know that the feasible path does not exist for  $c \in M_{NOT}$ . (2) We don't know whether a feasible path exists for  $c \in M_{UNKNOWN}$ . (3) We know that the feasible path exists for  $c \in M_{FEASIBLE}$ .

For the first case, in practice a router can refuse the QoS request or start the QoS negotiation. For the second case, we then expect that the probability to occur this case is very small. In the performance evaluation part of this paper (Part IV),

```

LEFPA (B, G, s)
1)  $x = 0$ 
2) WHILE (  $x \leq 1$  )
3)    $a_1 = x$ 
4)    $a_2 = 1 - a_1$ 
5)   FOR EACH edge  $e$  IN  $G$ 
6)      $g_a(e) = a_1 w_1(e) + a_2 w_2(e)$ 
7)   dijkstra( $G, s$ )
8)   FOR EACH node  $t$  IN  $G$ 
9)     IF  $p_a(s, t) \neq \text{last } p_b(s, t)$ 
10)      store  $p_a(s, t)$ 
11)   $x = x + (1/B)$ 

```

Fig. 3. Proposed heuristic LEFPA

extensive simulations show that it is really very small and most (95%) QoS requests will be in the other two areas. Furthermore, we also demonstrate that a QoS request in the area  $M_{UNKNOWN}$  has very small opportunity to have a feasible path. Therefore, we can regard this area as the unfeasible area without affecting the routing performance, and refuse this kind of QoS request. For the third case, we can select an element from  $\{p_a\}$  as the feasible path.

Because a practical algorithm cannot implement a continuous change of vector  $a$ , we cannot guarantee to find all of the elements in  $\{p_a\}$ . As a result, the unknown area  $M_{UNKNOWN}$  is enlarged. It is necessary to validate that, based on few discrete vectors  $a$ , the enlarged  $M_{UNKNOWN}$  decreases the overall performance of this algorithm very little.

Because the discrete values of  $a$  must be independent of networks, we normalize the weights of each link first. The maximum possible weight of each metric on a link is equal; i.e. the weight  $\max_{e \in E} w_l(e)$  is a constant that is independent of  $l$  where  $l \in \{1, 2\}$ .

#### B. Description of LEFPA

We propose the precomputation heuristic LEFPA for 2-constrained problems and 1-constrained optimal problems in Fig. 3. The heuristic, running on node  $s$ , includes the following steps. (a) Create a number ( $B$ ) of vectors  $a = (a_1, a_2)$  according to configuration (Line 3-4). (b) For the given network graph  $G$  and vector  $a$ , calculate the energy  $g_a(e)$  for each link (Line 5-6). (c) Use Dijkstra's algorithm to create least-energy tree  $T(a)$  with source node  $s$  and keyword  $g_a$  (Line 7). Save the least-energy path from  $s$  to each node along  $T(a)$  to QoS routing table (Line 8-10). When saving the current least-energy path  $p_a(s, t)$ , the last path  $p_b(s, t)$  from  $s$  to  $t$  must be checked to see if these two paths are identical. If  $p_b(s, t) = p_a(s, t)$ , then it is not necessary to save the current path  $p_a(s, t)$ . This process can guarantee that there are no identical paths in the routing table. Because the linear function  $g_a(e)$  has isotonicity, the

routing table generated by each node with the same vector  $a$  will not have a routing loop based on the consistent network state information [7].

### C. Complexity of LEFPA

Now we analyze the complexity of LEFPA. In a graph  $G$ , the node number is  $n = |V|$ , while the edge number  $m = |E|$ . Step (b) has the complexity of  $O(m)$ . Step (c) has the complexity of  $O(n \log n + m)$  with the improved Dijkstra's algorithm. Step (d) is  $O(n)$ . Thus, the overall time complexity of LEFPA is  $O(B(m + n \log n + n))$ , which is as  $B$  times as the original Dijkstra's algorithm with a single metric in the graph.

LEFPA creates the QoS routing table by saving least-energy paths. Because it avoids saving identical paths, the size of the QoS routing table is less than or equal to  $B$  times that of the original routing table with a single-metric network. For the 2-constrained problems, when a QoS request arrives, we only need to look for a feasible path satisfying  $p_a(s, t) \leq c(s, t)$  in the (at most  $B$ ) paths from  $s$  to  $t$ . If such a path is found, then return the path  $p_a(s, t)$ , and otherwise refuse the QoS request. For the one constraint optimal problem (e.g. DCLC), we then need to look for the optimal path that satisfies the constraint in all s-to-t paths (at most  $B$  paths). Assume that it needs to look up entries in the original routing table for  $L$  times in a single-metric network; the times in the QoS routing table is  $L + \lceil \log_2 B \rceil$  with bisearch.

For the smaller  $B$ , it is more likely to fail to find a feasible path in  $\{p_a\}$ , while the larger  $B$  makes LEFPA impractical. We will illustrate the relation between  $B$  and the performance with extensive simulations, which show that LEFPA performs well when  $B$  is small (e.g.  $B = 7$ ).

## IV. PERFORMANCE EVALUATION

We first propose the method of unknown-area proportion to evaluate the absolute performance of LEFPA. We then simulate QoS requests to compare its performance with other algorithms directly.

### A. Unknown-area proportion

We propose the method of unknown-area proportion, which is independent of QoS requests, to evaluate the absolute performance of LEFPA. We take the unknown area as the inefficiency of LEFPA. Thus, the simulation experiment is to analyze the relation between the inefficiency probability  $\text{Pr} = M_{\text{UNKNOWN}} / (M_{\text{NOT}} + M_{\text{AVL}})$  and  $B$ . Because  $M_{\text{UNKNOWN}}$  is a limited area while  $M_{\text{NOT}}$  and  $M_{\text{AVL}}$  are unlimited areas, we take the uppermost point  $p_a$  and take the rightmost point  $p_b$  as shown in Fig 2.b as the border points.

### B. Evaluation of absolute performance

Based on the pure random network graph with node number  $N$  [8], we generate two weights for each link, where  $w_1(e) \sim [1, 1000]$ ,  $w_2(e) \sim [1, 1000]$  and  $w_1(e)$  and  $w_2(e)$  have no correlation. We simulate the instances with  $N$  being 50, 100, 200 and 500, respectively. We generate 10

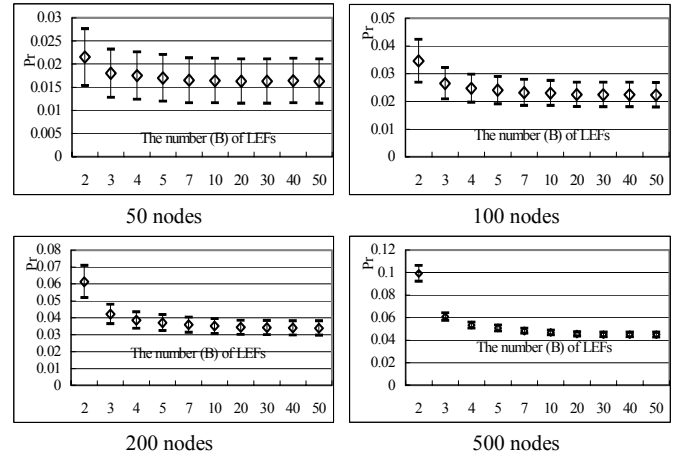


Fig. 4. The absolute performance of LEFPA

graphs in each instance, and in each graph we select source nodes 100 times (a particular node can be selected more than one time). Each time, the source node  $s$  uses LEFPA with a number ( $B$ ) of LEFs  $g_{a_i}$  to compute a least-energy tree,

where  $a^i = (i/B, (B-i)/B)$  and  $i = 0, 1, \dots, B-1$ . On each source node  $s$ ,  $\text{Pr}_s = \frac{1}{N} \sum_{t \in G} \text{Pr}_s(t)$  is calculated by taking each node in the graph as the destination  $t$  once. Then, for all the 100 source nodes, the average  $\text{Pr} = \frac{1}{100} \sum \text{Pr}_s$  is calculated. At last, we analyze the 10 graphs with the same number of nodes statistically and obtain the average  $\text{Pr}$  and the 95% confidence interval for each  $B$ .

Figure 4 shows the average inefficiency probability  $\text{Pr}$  with the 95% confidence interval versus  $B$  for random graphs with 50, 100, 200 and 500 nodes. When the node number is relatively small, the proportion of unknown area is very little. With the increase of node number, the proportion  $\text{Pr}$  increases. This shows the larger the network, the more the path number between a particular source-destination pair. With the increase of path number, the ability to optimize a path increases and optimization of a particular weight increases. This leads to the situation in which the uppermost point  $p_a$  is closer to the Y-coordinate and the rightmost point  $p_b$  is closer to the X-coordinate. However, the 95% confidence interval decreases with the increase of network scale. This shows the adaptability of LEFPA for large-scale network.

Although  $\text{Pr}$  decreases with  $B$  increasing, Fig. 4 shows that when  $B$  is large enough, the inefficiency probability  $\text{Pr}$  is insignificant. For example, when  $B = 7$  and  $N = 500$ ,  $\text{Pr} = 0.0482 \pm 0.0022$  with a 95% confidence interval. If we regard that algorithms are practical if  $\text{Pr}$  is less than 5%,  $B = 7$  is enough. This shows that in practice, to ensure high performance, we need only 7 uniform LEFs to find enough paths of different characteristics. With larger  $B$ , most of the newly found paths are reduplications and  $\text{Pr}$  nearly goes down no more. This is consistent with the conclusion in [9]. Therefore, the  $\text{Pr}$  in LEFPA with  $B = 7$  is close to that in the mathematical model with continuous change of the coefficient. That is to say,  $M_{\text{UNKNOWN}}$  with  $B = 7$  in practical is close

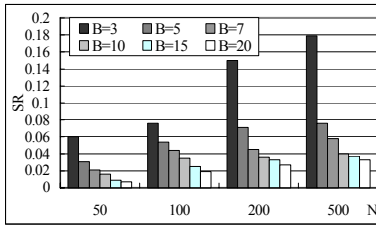


Fig. 5. Unfeasibility of requests in unknown area

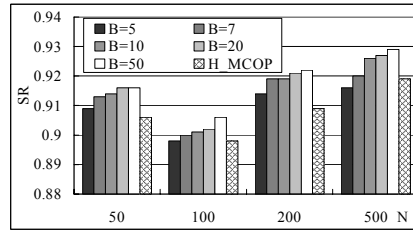


Fig. 6. Comparison with random constraints

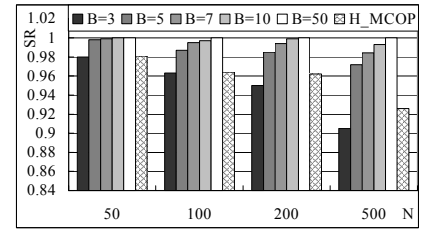


Fig. 7. Comparison with simulated constraints

to that in theory.

Although we demonstrated that the unknown area is small ( $Pr$  is little), we now demonstrate that the feasibility for a request  $c \in M_{UNKNOWN}$  is also small. First, we generate the constraints for requests within the unknown area in Fig. 2.b randomly. Then, we use H\_MCOP [6], one of the most efficient QoS algorithms, to calculate the feasible paths for each request. Fig. 5 shows the SR for each  $B$ . The experiment shows that when  $B = 7$ , the SR is less than 5% and most requests do not have a feasible path.

Having established that (1) the unknown area is small, and (2) the request within the unknown area has a low feasibility, LEFPA can refuse the request within the unknown area with a small probability of misjudgment (refuse the requests that have a feasible path), i.e., about  $0.05 \times 0.05 = 0.25\%$ . As a result, LEFPA achieves a high absolute performance.

#### C. Comparison between LEFPA and H\_MCOP

Current precomputation algorithms tend to have a prohibitive computing complexity, and most are based on distance vectors, so they are not fit for large-scale networks. In order to show the performance of LEFPA, we compare LEFPA with H\_MCOP [6], one of the most efficient algorithms, which is also based on Dijkstra's algorithm.

To compare the routing performance of LEFPA and H\_MCOP, we use two methods to generate constraints for QoS requests, including random constraints used in [6] and simulated constraints. These two methods are omitted for brief in this paper and detailed in [10]. Showing the SR with the random constraints, Fig. 6 demonstrates that when  $B \geq 7$ , the SR of LEFPA is higher than that of H\_MCOP. Fig. 7 shows the SR with simulated constraints. When  $B \geq 7$ , LEFPA overmatches H\_MCOP and is more insensitive to the network scale than H\_MCOP.

As a summary, the next-generation networks that provide QoS can follow the current precomputation routing architecture. When calculating the SPT, it only needs to change the current cost to the energy with a number ( $B = 7$ ) of uniform LEFs. The computation complexity of QoS is only  $B$  times the current complexity. Our extensive simulations below will show that LEFPA has a higher SR than H\_MCOP, although H\_MCOP has to compute a feasible path pertinently for each individual request. On the aspect of complexity, H\_MCOP is  $O(km \log kn + n \log n + (k^2 + 1)m)$  with  $k$ -shortest path algorithm, while LEFPA is  $O(B(m + n \log n + n))$ . When  $B = 7$ , LEFPA is better than H\_MCOP, not only on computation complexity but also on the

running time in practical experiments. For example, the running time of LEFPA is 51.8 millisecond in a 500-node graph, while that of H\_MCOP is 15.3 millisecond. Considering the difference between precomputation and on-line computation, if on a given source node, there are 50 requests to the other 50 nodes in the network for instance, the running time of LEFPA keeps fixed while that of H\_MCOP will increase 50 times of origin to 765 millisecond.

#### V. CONCLUSION

Based on the theoretical analysis of LEF, this paper proposes a novel precomputation heuristic, LEFPA, for QoS. In this algorithm, a node can use a number ( $B$ ) of uniform LEFs to precompute the QoS routing table with the time complexity  $O(B(m + n \log n + n))$ , which is only  $B$  times original Dijkstra's algorithm. The size of the QoS routing table is less than or equal to  $B$  times that of the current routing table with a single weight named cost. When requests arrive at a high speed, the router only needs routing table look-up rather than online computation. Further enhancing the scalability, the precomputation mode is also fit for the current routing architecture of the Internet. Extensive simulations show that our LEFPA performs well in complexity, performance and scalability. Therefore, we conclude that the precomputing LEFPA promises to be a valuable QoS algorithm for high-speed next-generation networks.

#### REFERENCE

- [1] X. Xiao and L. M. Ni, Internet QoS: A big picture, IEEE Network, vol. 13, no. 2, pp. 8–18, March-April 1999.
- [2] Y. Cui, J. P. Wu, K. Xu, et al. Research on Internetwork QoS Routing Algorithms: A Survey. Chinese Journal of Software, vol.13, no.11, pp.2065–2075, 2002.
- [3] M. S. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, New York, 1979.
- [4] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. IEEE JSAC, vol. 14, no. 7, pp. 148–154, 1996.
- [5] T. Korkmaz, M. Krunz, S. Tragoudas. An efficient algorithm for finding a path subject to two additive constraints. ACM. Performance Evaluation Review, vol.28, no.1, pp.318-27, June 2000.
- [6] T. Korkmaz, M. Krunz, Multi-Constrained Optimal Path Selection. IEEE INFOCOM'01, pp. 834 -843, vol.2, 2001.
- [7] J. L. Sobrinho, Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet, IEEE INFOCOM'01, pp. 727 -735 vol.2, 2001.
- [8] E. W. Zegura, K. L. Calvert, M. J. Donahoo, A quantitative comparison of graph-based models for Internet topology. IEEE/ACM Transactions on, Networking, vol. 5, no. 6, pp. 770–783, Dec. 1997.
- [9] X. Yuan, X. Liu, Heuristic Algorithms for Multi-Constrained Quality of Service Routing, IEEE INFOCOM'01, pp. 844 -853 vol.2, 2001.
- [10] Y. Cui, J. P. Wu, K. Xu. LEFPA: A Linear Energy Function Based Precomputation Algorithm for QoS Routing. Technique Report. available at <http://netlab.cs.tsinghua.edu.cn/~cui/report/>.