On the Extended Bellman–Ford Algorithm to Solve Two–Constrained Quality of Service Routing Problems

Xin Yuan
Department of Computer Science
Florida State University
Tallahassee, FL 32306
xyuan@cs.fsu.edu

Abstract

Two–constrained quality of service (QoS) routing finds a route in the network that satisfies two independent quality of service constraints. This problem is NP–hard and a number of heuristic algorithms have been proposed to solve the problem. This paper considers two heuristics, namely the limited granularity heuristic and the limited path heuristic, that can be applied to the extended Bellman–Ford algorithm to efficiently solve two–constrained QoS routing problems. Analytical and simulation studies are conducted to compare the effectiveness of the heuristics in finding the paths that satisfy the QoS constraints and the time/space requirements of the heuristics. The major results of this paper include the followings. First, the paper proves an optimal limited granularity heuristic scheme. The scheme is optimal in the sense that it provides optimal worst case guarantee in finding the paths that satisfy the QoS constraints among all limited granularity heuristic schemes. Second, the paper shows that, in polynomial time, the limited path heuristic has very high probability in finding a path in a randomly generated graph that satisfies the QoS constraints if such a path exists.

1 Introduction

The migration to integrated networks for voice, data and multimedia application introduces new challenges in supporting predictable communication performance. Multimedia applications require the communication to meet stringent requirement on delay, delay–jitter, cost and/or other quality of service (QoS) metrics. QoS routing, which identifies paths that meet the QoS requirement and selects one that leads to high overall resource efficiency, is the first step toward achieving end–to–end QoS guarantees.

The QoS requirement of a connection is typically specified as a set of *constraints*. For example, a delay constraint requires the total delay of a path to be no greater than a given upper bound. There are two types of QoS constraints, *link constraints* and *path constraints* [2]. The link constraints are the QoS metrics that can be determined by the state of the bottleneck link. Examples of link constraints include bandwidth constraints and buffer space constraints. The path constraints are the QoS metrics that can be determined by the combined state of the whole path. Examples of path constraints include delay constraints and cost constraints.

The multi-constrained path routing is to find a path that satisfies multiple independent path constraints. One example is to find a route with bounded end–to–end delay and bounded end–to–end cost. The multi–constrained path routing is known to be NP–hard[4, 7] under the assumption that the QoS metrics are real numbers or unbounded integer numbers. In the rest of the paper, it is assumed that the QoS metrics are real numbers. Two–constrained path routing, a special case of the multi–constrained path routing, is also NP–hard. The Bellman–Ford shortest path algorithm has been extended to solve two–constrained path routing problems [1, 8]. The algorithm in [8] guarantees to find a path that satisfies the QoS constraints if such a path exists. In the worst case, the time complexity of the algorithm may grow exponentially with respect to the network size. Algorithms in [1] find approximate solutions in polynomial time.

This paper considers two polynomial time heuristics, the *limited granularity heuristic* and the *limited path heuristic*, that can be applied to the extended Bellman–Ford algorithm to solve two–constraints QoS routing problems. The limited granularity heuristic obtains approximate solutions in polynomial time by using finite domains, such as bounded ranges of integer numbers, to approximate the infinite number of values that QoS metrics can take. The limited path heuristic focuses on the cases that occur most frequently in general and solves these cases efficiently and effectively. For a randomly generated graph, the limited path heuristic

can efficiently and with very high probability find a path that satisfies the QoS constraints when such a path exists. The main contributions of the paper are as follows. First, the paper summarizes two types of heuristics that can be applied to the extended Bellman-Ford algorithm to efficiently solve two–constrained QoS path routing problems. Second, the paper proves that the algorithm proposed in [1] is optimal among all the limited granularity heuristics in the sense that the algorithm provides optimal worst case guarantee in finding paths that satisfy the QoS constraints. Third, the paper proposes the limited path heuristic and shows that the heuristic can efficiently find paths that satisfy the QoS constraints with very high probability when such paths exist. Finally, the paper evaluates and compares the two heuristics.

The rest of the paper is organized as follows. Section 2 formally describes the multi-constrained QoS path problem and discusses the extended Bellman–Ford algorithm that can solve this problem. Section 3 studies the limited granularity heuristic for two–constrained problems. Section 4 presents the limited path heuristic. Section 5 summarizes the simulation results. The related work is discussed in Section 6. Section 7 concludes the paper.

2 Background

2.1 Notation and Assumptions

The network is modeled as a directed graph $G(V, E)$, where V is the set of nodes representing routers and E is the set of edges representing links that connect the routers. Each edge $e = u \rightarrow v$ is associated with k independent weights, $w_1(e)$, $w_2(e)$, ..., $w_k(e)$, where $w_l(e)$ is a positive real number ($w_l(e) \in R^+$) for all $1 \leq l \leq k$. The notation $w(e) = w(u \rightarrow v) = (w_1(e), w_2(e), ..., w_k(e))$ is used to represent the weights of a link. It is assumed that the weight functions are additive [7], that is, the weight of a path is equal to the summation of the weights of all edges on the path. Thus, for a path $p = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_k$, $w_l(p) = \sum_{i=1}^{k} w_l(v_{i-1} \rightarrow v_i)$. Notation $w(p) \leq w(q)$ denotes $w_l(p) \leq w_l(q)$ for all $1 \leq l \leq k$. Other comparative operators $<, =, >, \geq$ and arithmetic operators $+, -$ on the weight vectors are defined similarly. Let a path $p = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_k$ and a link $e = v_k \rightarrow v_{k+1}$. The notation $p + e$ or $p + v_k \rightarrow v_{k+1}$ denotes the path $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_k \rightarrow v_{k+1}$. Given a set S, the notation $|S|$ denotes the size of the set S.

2.2 Multi–Constrained Path Problem

Definition 1: Given a directed graph $G(V, E)$ with $k \geq 2$ weight functions $w_1 : E \rightarrow R^+$, $w_2 : E \rightarrow R^+$, ..., $w_k : E \rightarrow R^+$, a path $p = src \rightarrow v_1 \rightarrow v_2 \rightarrow ... \rightarrow dst$ is said to be an *optimal QoS path* from node src to node dst if there does not exist another path q from node src to node dst such that $w(q) < w(p)$, that is, for all $1 \leq l \leq k$, $w_l(q) < w_l(p)$.

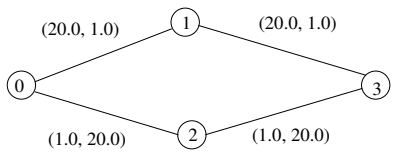


Figure 1. Optimal QoS paths

When $k = 1$, the optimal QoS path is the same as the shortest path. When $k > 1$, however, there can be multiple optimal QoS paths between two nodes. For example, in Figure 1, both path $p_1 = 0 \rightarrow 1 \rightarrow 3$ and path $p_2 = 0 \rightarrow 2 \rightarrow 3$ are optimal QoS paths from node 0 to node 3.

Lemma 1: If there exists a path from node src to node dst that satisfies the QoS constraints, then there exists an optimal QoS path from src to dst that satisfies the same QoS constraints.

Proof: Straight forward from the definition of the optimal QoS path. □

Lemma 1 establishes that for a QoS routing algorithm to guarantee finding a path that satisfies the QoS constraints when one exists, the algorithm only needs to cover the optimal QoS paths. The optimal QoS paths are interesting because each optimal QoS path can potentially satisfy particular QoS constraints that no other path can satisfy. On the other hand, when there exists a path that satisfies the QoS requirement, there always exists an optimal QoS path that satisfies the same QoS requirement.

Definition 2: Given a directed graph $G(V, E)$, a source node src, a destination dst, $k \geq 2$ weight functions $w_1 : E \rightarrow R^+$, $w_2 : E \rightarrow R^+$, ..., $w_k : E \rightarrow R^+$, k constants c_1, c_2, ... c_k; the *multi–constrained path problem* is to find a path p from src to dst such that $w_1(p) \leq c_1$, $w_2(p) \leq c_2$, ..., $w_k(p) \leq c_k$. For a problem with k constraints, we call the problem a *k–constrained* problem. In particular, when $k = 2$, the problem is called a *two*–constrained problem.

2.3 Extended Bellman–Ford Algorithm

Figure 2 presents an Extended Bellman–Ford Algorithm (EBFA) that solves the multi–constrained QoS routing problem. The algorithm is a variation of the Constrained Bellman–Ford algorithm [8]. EBFA extends the original Bellman-Ford shortest path algorithm [3] by having each node u to maintain a set $PATH(u)$ that records all optimal QoS paths found so far from src to u. Routine $BELLMAN_FORD$ determines whether there exists a path that satisfies the QoS constraints from src to dst. In $BELLMAN_FORD$, lines (1) to (3) initialize the vari-

```
RELAX(u, v, w)
(1)     For each w(p) in PATH(u)
(2)       flag = 1
(3)       For each w(q) in Path(v)
(4)         if(w(p) + w(u,v) ≥ w(q)) then
              flag = 0
(5)         if (w(p) + w(u,v) < w(q)) then
              remove w(q) from PATH(v)
(6)       if (flag = 1) then
              add w(p) + w(u,v) to PATH(v)

BELLMAN-FORD(G, w, c, src, dst)
(1)     For i = 0 to |V(G)| − 1
(2)       PATH(i) = φ
(3)     PATH(src) = {0}
(4)     For i = 1 to |V(G)| − 1
(5)       For each edge (u,v) ∈ E(G)
(6)         RELAX(u, v, w)
(7)     For each w(p) in PATH(dst)
(8)       if (w(p) < c) then return "yes"
(9)     return "no"
```

Figure 2. Extended Bellman–Ford Algorithm

ables and lines (4) to (6) perform the relax operations. After the relax operations, all optimal QoS paths from node src to node dst are stored in $PATH(dst)$. Lines (7) and (8) check whether there exists an optimal QoS path that satisfies the QoS constraints. The $RELAX(u, v, w)$ operation is complicated since all the elements in $PATH(u)$ and $PATH(v)$ must be considered. For each element $w(p)$ in $PATH(u)$, line (4) in the RELAX routine checks whether there exists an old path q from src to v that is better than path $p + (u \rightarrow v)$. If such q exists, then $p + (u \rightarrow v)$ is not an optimal QoS path. Line (5) checks whether path $p + (u \rightarrow v)$ is better than any old path q from src to v. If such q exists, then path q is not an optimal QoS path and is removed from the set $PATH(v)$.

The algorithm executes the RELAX operation $O(|V||E|)$ times, where V is the set of nodes and E is the set of edges in the graph. The time and space needed to execute the RELAX(u, v, w) depend on the sizes of the $PATH(u)$ and $PATH(v)$ sets, which are the number of optimal QoS paths from node src to nodes u and v respectively. Since the number of the optimal QoS paths from src to u or v can be exponential with respect to the size of V and E, the time and space requirement of $EBFA$ may also grow exponentially.

Both the limited granularity heuristic and the limited path heuristic aim at reducing the time and space requirement of $EBFA$ while maintaining its effectiveness in finding paths that satisfy the QoS constraints. The idea of both heuristics is to limit the number of paths maintained in each node, that is, the size of the set $PATH$, and thus, limit the time and space requirement to execute the $RELAX$ operation. The next few sections will discuss the limited granularity heuristic and the limited path heuristic in details.

3 The Limited Granularity Heuristic for Two–Constrained Problems

The idea of the limited granularity heuristic is to use bounded finite ranges to approximate the QoS metrics, which reduces the original NP–hard problem to a simpler problem that can be solved in polynomial time. This heuristic is a generalization of the algorithm in [1].

Let $w_1 : E \rightarrow R^+$ and $w_2 : E \rightarrow R^+$ be the two weight functions associated with each link in the graph $G = (V, E)$. The goal of the two–constraints QoS routing is to find a path p that satisfies the two QoS constraints, $w_1(p) \leq c_1$ and $w_2(p) \leq c_2$.

The limited granularity heuristic limits the number of paths maintained in each node to be X and uses a bounded finite range that contains X elements to approximate one of the two QoS metrics. Since approximating w_1 is essentially the same as approximating w_2, in the rest of the discussion, it is assumed that w_1 is approximated. The limited granularity heuristic maps the range $[0, c_1]$ (with infinite elements) into a finite range with X elements, r_1, r_2, ..., r_X, where $0 < r_1 < r_2 < ... < r_X = c_1$. Any w_1 weight $x \in [0, c_1]$ is approximated by r_i if and only if $r_{i-1} < x \leq r_i$. Notation $aw_1(p)$ will be used to denote the approximated w_1 weight of p in the bounded finite domain. The choice of the values r_1, r_2, ..., r_X is called a *mapping scheme*.

The limited granularity heuristic is shown in Figure 3. Each node u maintains a table $d[u, i]$, $1 \leq i \leq X$, that records the best QoS paths (smallest w_2 weight paths) with w_1 weights less than or equal to r_1, r_2, ..., r_X. In the $RELAX(u, v, w)$ operation, to compute $d(v, i)$, only $d(u, j)$, where j is the largest k such that $r_k \leq r_i - w_1(u, v)$, needs to be checked. The choice of j guarantees the w_1 weight of the considered path to be less than r_i. The $RELAX$ routine in the limited granularity heuristic has a time complexity of $O(X)$. By limiting the granularity of the QoS metrics, the limited granularity heuristic has a time complexity of $O(X|V||E|)$.

A number of important questions related to the performance of this heuristic have yet to be answered. How large (small) should we set the value X for the algorithm to be effective? How to choose the values for r_l, $1 \leq l \leq X$, to achieve best results? The rest of the section will attempt to answer these questions.

Lemma 2: Let X be the size of the table maintained in each node. When w_1 weight for each link is a positive real number, the limited granularity heuristic cannot find any path that meets the two conditions: (1) the path satisfies the QoS constraints, and (2) the length of the path is larger than X.
Proof: See [9]. □

```
RELAX(u, v, w)
(1)     for each d[v, i] ∈ PATH(v)
(2)       Let j be the largest j such that r_j < r_i − w_1(u, v)
(3)       if (d[v, i] > d[u, j] + w_2(u, v)) then
(4)         d[v, i] = d[u, j] + w_2(u, v)

Limited_Granularity_Heuristic(G, w, c, src, dst)
(1)     For i = 0 to |V(G)| − 1
(2)       For j = 1 to X
(3)         d[i, j] = ∞
(4)     For j = 1 to X
(5)       d[src, j] = 0
(6)     For i = 1 to |V(G)| − 1
(7)       For each edge (u, v) ∈ E(G)
(8)         RELAX(u, v, w)
(9)     if (d[dst, X] < c_2) then return "yes"
(10)    return "no"
```

Figure 3. The limited granularity heuristic

Lemma 2 shows that in order for the limited granularity heuristic to be effective in finding paths of length L, the number of entries in each node should be at least L. For an N–node graph, paths can potentially be of length N. Thus, the limited granularity heuristic should at least maintain $O(N)$ entries in each node to be effective.

Theorem 1: Let X be the number of table entries in each node. If there exist a path p such that

$$w_1(p) \leq c_1 - \frac{L-1}{X} * c_1, w_2(p) \leq c_2,$$

the limited granularity heuristic with the *uniform mapping scheme*, that is, $r_i = i * \frac{c_1}{X}$, for all $1 \leq i \leq X$, can guarantee finding a path q that satisfies the QoS constraints $w_1(q) \leq c_1, w_2(q) \leq c_2$. Limited granularity heuristics with other mapping schemes **cannot** guarantee to find a path that satisfies the QoS constraints.

Proof: The limited granularity scheme with the uniform mapping scheme is similiar to the algorithm in [1]. The proof of the first half theorem can be found in [1].

Consider the second half of the theorem, the limited granularity heuristic using other mapping schemes cannot guarantee to find a path that satisfies the same QoS constraints. Let the other mapping scheme maps $[0, c_1]$ to X numbers, a_1, a_2, ...,a_X where $0 < a_1 < a_2 < ... < a_X = c_1$. Let j be the largest number such that $a_j < j * \frac{c_1}{X}$. Consider the two cases.

- *Such j does not exist.* $a_1 > \frac{c_1}{X}$. Consider a linear graph with $X + 1$ nodes. $w_1(v_i \rightarrow v_{i+1}) = \frac{c_1}{2*X^2}$, for all $0 \leq i \leq X - 1$. Consider the path $p = v_0 \rightarrow v_1 \rightarrow ... \rightarrow v_X$. $length(p) = X$. $w_1(p) = X * \frac{c_1}{2*X^2} = \frac{c_1}{2X} \leq c_1 - \frac{X-1}{X} * c_1$. Thus, using the uniform mapping, the limited granularity heuristic can detect this path. When using other mapping scheme, $aw_1(p) \geq X * a_1 > c_1$ and the heuristics cannot find the path.

- *Such j exists.* Consider a linear graph with $X - j + 2$ nodes, v_0, v_1, ..., v_{X-j+1}. Let the weight function be $w_1(v_i \rightarrow v_{i+1}) = \frac{j*\frac{c_1}{X} - a_j}{2*(X-j)}$, for all $1 \leq i \leq X - j$. $w_1(v_0 \rightarrow v_1) = \frac{a_j + j*\frac{c_1}{X}}{2}$. Consider the path $p = v_0 \rightarrow v_1 \rightarrow ... \rightarrow v_{X-j+1}$, $w_1(p) = \frac{a_j + j*\frac{c_1}{X}}{2} + (X - j) * \frac{j*\frac{c_1}{X} - a_j}{2*(X-j)} = \frac{j}{X}c_1 \leq c_1 - \frac{X-j+1-1}{X}c_1$. In the non-uniform mapping scheme, $aw_1(v_0 \rightarrow v_1) = a_{j+1}$. Using similar proving technique as in the proof of Lemma 2, it can be shown that after considering $X - j$ more links, $aw_1(p) > a_X = c_1$ and the heuristic cannot find the path. □

This theorem shows that by maintaining $X = O(N)$ entries in each node, the limited granularity heuristic has a good chance of finding a path that meets the QoS constraints if one exists. However, the constant factor in the $O(N)$ should be fairly large to achieve good performance. For example, if $X = 100 * N$, the heuristic guarantees to find a path p with $w_1(p) \leq 0.99c_1, w_2(p) \leq c_2$, which is very close to the original QoS constraints $w_1(p) \leq c_1, w_2(p) \leq c_2$. This theorem also shows that the uniform mapping scheme can achieve the optimal worst case guarantee in finding paths that satisfy the QoS constraints among all limited granularity schemes.

4 Limited Path Heuristic for Two–Constrained Problems

The limited path heuristic ensures the worst case polynomial time complexity by maintaining a limited number of optimal QoS paths, say X optimal QoS paths, in each node. The heuristic is basically the same as the extended Bellman–Ford algorithm in Figure 2 except that before a path is inserted into $PATH$, the size of $PATH$ is checked. When $PATH$ already contains X elements, the new path will not be inserted. Note that the performance of the heuristic can be improved by using more flexible path replacement schemes that replace existing optimal QoS paths with newly found optimal QoS paths. However, to simplify the analysis, it is assumed that new path will not be inserted when the size of the set reaches the limit. By limiting the size of $PATH$ to X, the time complexity of the $RELAX$ operation is reduced to $O(X^2)$. Thus, the time complexity of the heuristic is $O(X^2|V||E|)$. This heuristic may miss some optimal QoS paths that satisfy the QoS constraints because it does not record all the optimal QoS paths. Thus, one important question to be answered is the following. How large (small) should we set the value X in

order for the heuristic to be effective? When X is large, the heuristic has a high probability in finding paths that satisfy the QoS constraints because each node has a good chance to record all optimal QoS paths. On the other hand, large X also results in large time and space requirement. The rest of the section will show what is the value of X in order for each node to have a good chance to record all optimal QoS paths and for the limited path heuristic to be effective.

Lemma 3: Let P be the number of paths of the same length from node src to node dst in a graph where the weights are randomly generated with uniform distribution, the average number of optimal QoS paths in the P paths is less than or equal to $ln(P) + O(1)$.

Proof: See [9]. □

Given an N–node graph with a maximum nodal degree of d, the average number of optimal QoS paths of length L from a node to any other node is $O(ln(d^L))$ = $O(L \times ln(d))$. For a graph with small nodal degree, the average number of optimal QoS paths between two nodes is O(L). For a graph with large nodal degree, that is, the nodal degree is O(N), the number is O(Llg(N)).

Lemma 3 concerns about the average number of optimal QoS paths. A more interesting question is how many optimal QoS paths each node should record in order to have a very high probability to record *all* optimal QoS paths. This question directly relates to the performance of the limited path heuristic because if each node has a very high chance to record all optimal QoS paths, the heuristic will have a very high probability to find a path that satisfies the QoS constraints if such a path exists.

To determine the number of optimal QoS paths in a set of P paths of the same length from a source to a destination. Let us consider the following process. First, the path that has the smallest w_1 weight is chosen from the set. The path p is an optimal QoS path because its w_1 weight is the smallest among all the paths. All paths whose w_2 weights are larger than $w_2(p)$ are not optimal QoS paths. The set SR of paths whose w_2 weights are smaller than $w_2(p)$ may contain other optimal QoS paths. The process is then repeated on the set SR. Since w_1 weight and w_2 weight is independent and the length of the paths are the same, the probability of the size of the remaining set SR may be 0, 1, ..., $P-1$, each with probability $\frac{1}{P}$. The question to be answered is the followings. What is the probability that after selecting k optimal QoS paths, the remaining set is empty? In other words, what is the probability that there exists k optimal QoS paths in a set of P paths?

This problem can be modeled as a Markov process with $P+1$ states, 0, 1, ..., P, as shown in Figure 4. Each state i represent is state that there are still i paths to choose the optimal QoS paths from. Each time an optimal QoS path is chosen from a set of N paths, the remaining set size is 0 to N-1, each with probability $\frac{1}{N}$. The transition matrix of the

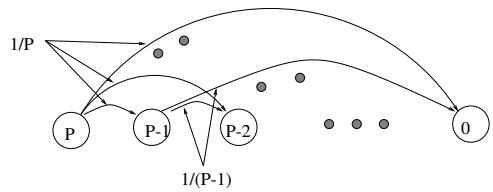


Figure 4. The Markov Chain

Markov chain is as follows.

$$A = \begin{pmatrix} 0 & \frac{1}{P} & \frac{1}{P} & \cdots & \frac{1}{P} \\ 0 & 0 & \frac{1}{P-1} & \cdots & \frac{1}{P-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{1} \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Let us define $A^1 = A$ and $A^k = A^{k-1}A$ for $k > 1$. $A^k(i,j)$ represents the probability of the state transferring from node i to node j in exactly k steps. For example, $A^1(P,0)$ represents the probability of a set of size P became empty after one optimal QoS path is chosen. The possibility is $\frac{1}{P}$. $A^k(P,0)$ is the probability that the set of size P becomes empty after selecting exactly k optimal QoS paths. By manipulating the matrix A^k, the following equations can be obtained. $A^1(P,0) = \frac{1}{P}$, $A^2(P,0) = \frac{1}{P}(\frac{1}{P-1} + \frac{1}{P-2} \cdots + \frac{1}{1}) = \frac{1}{P}\sum_{i=1}^{P-1}\frac{1}{i}$ and

$$A^k(P,0) = \frac{1}{P}\sum_{i_1=k-1}^{P-1}\frac{1}{i_1}\sum_{i_2=k-2}^{i_1}\frac{1}{i_2}\sum_{i_3=k-3}^{i_2}\cdots\sum_{i_k=1}^{i_{k-1}}\frac{1}{i_k}.$$

Lemma 4 and Lemma 5 obtain the numerical results for $A^i(P,0)$. The proofs of these lemmas can be found in [9].

Lemma 4: $A^i(P,0) \leq \frac{(ln(P))^{i+1} * 2}{P*(i+1)!}$. □

Lemma 5: $A^i(P,0) \leq \frac{1}{P^{e^2+1}}$ when $i > e^2 ln(P)$. □

Theorem 2: Given an N node graph with a nodal degree of $d \leq N$, the limited path heuristic has a very high probability to record all optimal QoS paths and thus, has a very high probability to find a path that satisfies the QoS constraints when one exists, when each node maintains $O(Llg(N))$ paths, where L is the length of the paths.

Proof: The total number of paths of length L between any two nodes in the graph is at most N^L. The probability that there exist no more than i optimal QoS paths among the $P = N^L$ paths is

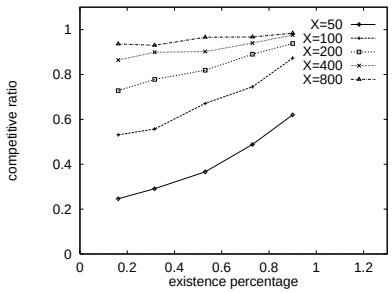
$$p = 1 - \sum_{k=i+1}^{P} A^k(P,0)$$

Apply Lemma 5 to the equation. When $i > e^2 lg(N^L)$,

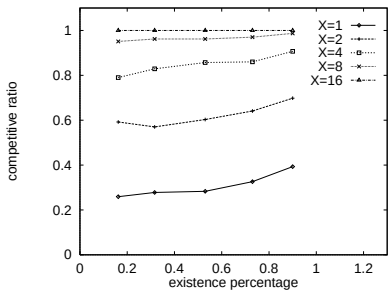
$$p = 1 - \sum_{k=i+1}^{P} A^{k+1}(P,0) \geq 1 - \sum_{k=i+1}^{P}\frac{1}{P^{e^2+1}} \geq 1 - \frac{1}{P^{e^2}}$$

When each node maintains $e^2 lg(N^L) = O(Llg(N))$ paths, the probability is very high, that is, more than $1 - \frac{1}{P^{e^2}}$, where $P = N^L$, that each node can record all optimal QoS paths, and thus the heuristic can find a path that satisfies the QoS constraints when one exists. □

To give an idea of how high the probability is, assuming $P = 100$, the probability is larger than 99.9999%. Another way to look at this issue is that we do not need to maintain $e^2 Llg(N)$ paths in each node to achieve high performance.



(a) Limited granularity heuristic



(b) Limited path heuristic

Figure 5. Results for 8×8 meshes

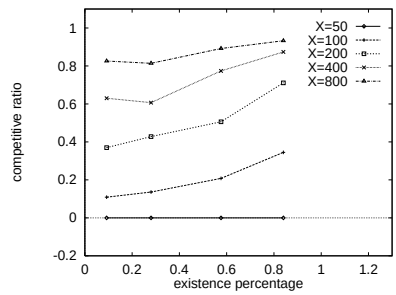
5 Experiments

This section studies and compares the two heuristics through simulation. The mesh topology is used in the study. Since the performance of the two heuristics is closely related to the length of the paths, the study chooses the source and the destination to be furthest apart in the mesh topology. In all the simulation, the w_1 weight of each link is randomly generated in the range of $(0.0, 10.0)$ and w_2 weight in the range of $(0.0, 20.0)$. The two heuristics are compared with the exhaustive algorithm, $EBFA$, that can guarantee find a path that satisfies the QoS constraints if such a path exists. Two concepts, the *existence percentage* and *competitive ratio*, are used to describe the performance. The two concepts are defined as follows.

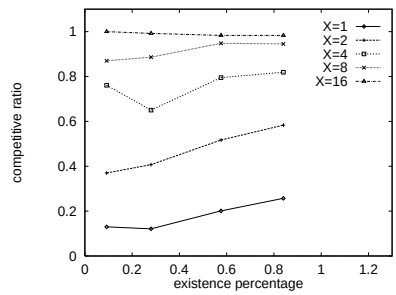
$$existence\ percentage = \frac{\#\ of\ satisfiable\ requests}{\#\ of requests}$$

$$comp.\ ratio\ = \frac{\#\ of\ requests\ satisfied\ by\ a\ heuristic}{\#\ satisfiable\ of\ requests}$$

The existence percentage indicates how difficult the paths that satisfy the QoS constraints can be found. The larger the existence percentage, the easier the paths can be found. The competitive ratio indicates how well the heuristic performs with respect to the exhaustive algorithm.



(a) Limited granularity heuristic



(b) Limited path heuristic

Figure 6. Results for 16×16 meshes

Figure 5 shows the results for 8×8 meshes. In both figures, the x-axis represents the existence percentage and the y–axis represents the competitive ratio. Different curves are for different values of X in the two heuristics. The data are obtained by running the two heuristics and the exhaustive algorithm using requests with the same QoS constraints on 500 randomly generated 8×8 meshes. The general trend in the figure is that both the limited granularity heuristics and the limited granularity heuristics can have close to 1 competitive ratio when sufficiently large number of entries are maintained in each node. However, to achieve high competitive ratio, the limited granularity heuristic requires to maintain a very large number of entries, e.g. 800 in this experiment, while the limited path heuristic only requires a small number of entries in each node, e.g. 16 in the experiment. Due to the large difference in the number of entries maintained in each node between the two heuristics, the limited path heuristic is also much more efficient in terms of execu-

tion time than the limited granularity heuristic.

Figure 6 shows the results for 16×16 meshes. The general trend in the 16×16 mesh is similar to that in the 8×8 mesh except that maintaining same amount entries in the larger mesh results in lower performance. For example, in the 8×8 mesh, the limited granularity heuristics has about 95% competitive ratio when maintaining 800 entries in each node while in the 16×16 mesh, it can only achieve about 80% competitive ratio when the existence percentage is close to 0.1. The degradation in performance for the limited path heuristic is not so severe as that for the limited granularity heuristic. Maintaining 8 entries in each node can still achieve a competitive ratio of 90% in the 16×16 mesh as oppose to 95% in the 8×8 mesh. Maintaining 16 entries has close to 1.0 competitive ratio for both 8×8 and 16×16 meshes. Notice that the performance of the limited path heuristic can be further improved when a path replacement algorithm is used to allow newly–found optimal QoS paths to replace existing optimal QoS paths.

6 Related Work

Much work has been done in QoS routing recently, an extensive survey can be found in [2]. Among the proposed QoS routing schemes, the ones that deal with multi-constrained QoS path routing are more related to the work in this paper. In [6], a distributed algorithm is proposed to find QoS paths that satisfy end–to–end delay while minimizing the cost. Although this algorithm considers two QoS metrics, it does not solve two–constrained problems because the cost metric is not bounded. Ma [5] showed that when weighted fair queuing algorithm is used, the metrics of delay, delay–jitter and buffer space are not independent and all of them become functions of bandwidth. In this case, the multi-constrained problem is solvable in polynomial time. Jaffe [4] proposed a distributed algorithm that solves two–constrained problems with a time complexity of $O(V^5 b log V b)$, where b is the largest number of the weights. This algorithm is pseudo-polynomial in that the execution time depends on the value of the weights (not just the size of the graph). Chen [1] proposed a heuristic algorithm that is close to the limited granularity heuristic. However, it only considers a fixed mapping schemes. This paper considers a more general heuristic and proves that the mapping scheme in [1] results in optimal worst case guarantee in finding paths that satisfy the QoS constraints among all the limited granularity schemes. Widyono [8] proposed an algorithm that performs exhaustive search on the QoS paths in exponential time. The limited path heuristic runs in polynomial time and performs as good as the exhaustive algorithm in most of the cases.

7 Conclusion

This paper studies two heuristics, the limited granularity heuristic and the limited path heuristic, that can be applied to the extended Bellman–Ford algorithm and efficiently solve two–constrained QoS path routing problems. The paper proves that the uniform mapping scheme, which is used in the algorithm in [1], can provide optimal worst case guarantee in finding paths that satisfy the QoS constraints among all the limited granularity schemes. The paper also shows that the limited path scheme can efficiently achieve very high probability in finding a path that satisfies the QoS constraints when such a path exists. The simulation study further demonstrates that the limited path heuristic is a very effective and efficient heuristic to solve two–constrained QoS path routing problems. In conclusion, although the two–constrained routing problem is NP–hard in the worst case, it can be solved efficiently in practice by using the heuristics discussed in this paper.

References

[1] S. Chen and K. Nahrstedt "On Finding Multi-Constrained Paths." *International Conference on Communications* (ICC'98), June 1998.

[2] Shigang Chen and Klara Nahrstedt " An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions," *IEEE Network, Special Issue on Transmission and Distribution of Digital Video,* Nov./Dec. 1998.

[3] Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms.", *The MIT Press*, 1990.

[4] J.M. Jaffe "Algorithms for Finding Paths with Multiple Constraints." *Networks*, 14:95–116, 1984.

[5] Q. Ma and P. Steenkiste "Quality–of–Service Routing with Performance Guarantees", *IWQos'97*, May 1997.

[6] H. F. Salama, D. S. Reeves and Y. Viniotis, "A Distributed Algorithm for Delay–Constrained Unicast Routing." *IEEE INFOCOM'97*, Japan, April 1997.

[7] Z. Wang and Jon Crowcroft "Qos Routing for Supporting Resource Reservation." *IEEE JSAC*, September 1996.

[8] Ron Widyono, "The Design and Evaluation of Routing Algorithms for Real–time Channels." *TR–94–024*, International Computer Science Institute, UC Berkeley.

[9] X. Yuan, "On the Extended Bellman–ford Algorithm to Solve Two–Constrained Quality of Service Routing Problems", *Technical Report, TR–990701*, Department of Computer Science, Florida State University, July 1999.