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 the delay-cost-constrained routing, i.e., finding a route in the network 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 number. In the rest of the paper, it is assumed that the QoS metrics are real numbers. Two–constrained path routing, which is a special case of multi–constrained path routing, is also NP–hard. Extended Bellman–Ford shortest path algorithm has been proposed to solve two–constrained path routing problems [1, 8]. In [8], a straight-forward extension of the Bellman-Ford shortest path algorithm is used to find paths that satisfy the QoS constraints. This algorithm 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 that can be applied to the extended Bellman–Ford algorithm to solve two–constraints QoS routing problems efficiently. The two heuristics are called the *limited granularity heuristic* and the *limited path heuristic*. The idea of the limited granularity heuristic is to use finite domains, such as bounded ranges of integer numbers, to approximate the infinite number of values that QoS metrics can take. When all QoS metrics except one are taken from finite domains, the multi–constrained path routing problem is solvable in polynomial time [1]. This heuristic is a generalization of the algorithm in [1] in the sense that it does not fix the approximation scheme. This paper proves an optimal approximation scheme that yields optimal worst case guarantee in finding paths that satisfy the QoS constraints among all limited granularity heuristic schemes. 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 summarized as follows.

- 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, the limited granularity heuristic and the limited path heuristic.
- The paper proves the *optimal scheme* for the limited granularity heuristic. The scheme is optimal in the sense that it provides optimal worst case guarantee among all limited granularity schemes.
- 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.
- The paper evaluates and compares the two heuristics through simulations and demonstrates that the limited path heuristic is a very effective heuristic.

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. This paper considers centralized algorithms and assumes that each node has precise information about the link states in the whole network.

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 $1 \leq l \leq k$, $w_l(q) < w_l(p)$.

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, because $w_1(p_1) = 40.0 > 2.0 = w_1(p_2)$ and $w_2(p_2) = 40.0 > 2.0 = w_2(p_1)$.

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. $\square$

Lemma 1 establishes that for an 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.

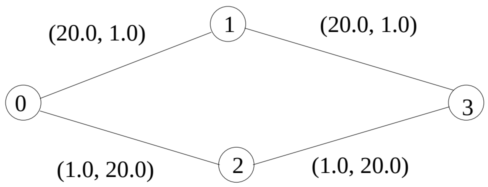


Figure 1: Optimal QoS paths

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

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

Figure 2: Extended Bellman–Ford algorithm for the multi–constrained QoS routing problem

Definition 2 *Multi–constrains path problem*: 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 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 k*–constrained* problem. In particular, when $k = 2$, the problem is call a *two*–constrained problem.

2.3 Extended Bellman–Ford Algorithm

The Bellman–Ford shortest path algorithm can be extended to solve the multi–constrained QoS path routing problem. One such algorithm is called Constrained Bellman–Ford (CBF) algorithm [8]. CBF guarantees to find a solution for this problem when one exists. The running time of this algorithm may grow exponentially with respect to the graph size. Thus, heuristics must be developed on top of the algorithm to reduce the time and space complexity. Since the heuristics considered in this paper are applied to this algorithm, a variation of the CBF algorithm, which will be called $EBFA$ in the rest of the paper, is presented in Figure 2 for the

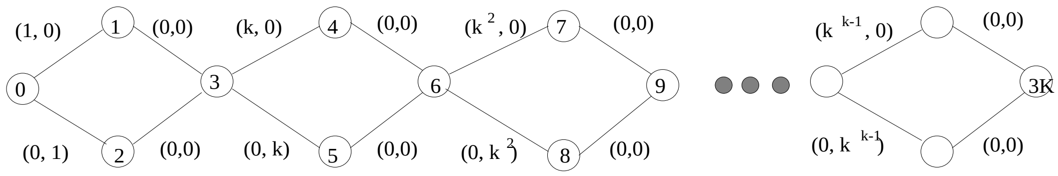


Figure 3: The number of optimal QoS paths between two nodes

completeness of the paper. For simplicity, the algorithm in Figure 2 only checks whether there exists a path that satisfies the QoS constraints from src to dst. The algorithm can easily be modified to find the exact QoS path from src to dst.

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. The first three lines in the main routine (BELLMAN_FORD) initialize the variables. Lines (4) to (6) perform the relax operations as in the original Bellman–Ford shortest path algorithm. After the relax operations all optimal QoS paths from node src to node dst are stored in the set $PATH(dst)$. Lines (7) and (8) check whether there exists an optimal QoS path that satisfies the QoS constraints. The $RELAX(u, v, w)$ operation is a little more 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 $PATH(v)$.

Consider the time and space complexity of the algorithm. 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) depends 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. Figure 3 shows an example that the number of QoS paths between two nodes grows exponentially with respect to the size of the network. In Figure 3, the number of optimal QoS paths from node 0 to node $d = 3k$ is equal to 2^k because from each node $3i$ where $0 \leq i \leq k$, taking the link $3i \rightarrow 3i + 1$ or $3i \rightarrow 3i + 2$ will results in different optimal QoS paths. Thus, in the worst case, time and space requirement of the algorithm can grow exponentially with respect to the network size.

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. When the time complexity of the RELAX routine is reduced to polynomial, the whole algorithm will have a polynomial time complexity in the worst case since the algorithm executes the RELAX routine $O(|V||E|)$ times. Limiting the size of the PATH set in each node may miss some optimal QoS paths and may only find approximate solutions. Thus, the challenge of the heuristics is how to use the limited resources to achieve good performance. 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

As shown in [1], if all QoS metrics except one take bounded integer values, the multi-constrained QoS routing problem is solvable in polynomial time. 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. Thus, $aw_1(p) \in \{r_1, r_2, ..., r_X\}$. The choice of the values r_1, r_2, ..., r_X is called a mapping scheme.

The limited granularity heuristic is shown in Figure 4. 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 heuristics is simpler than that in $EBFA$ and has a time complexity of $O(X)$. Besides the RELAX function, the initialization in the BELLMAN_FORD routine is also changed as well. Specifically, all $d[v, i]$, $0 \leq v \leq |V| - 1$, $v \neq src$ and $1 \leq i \leq X$ must be set to infinity to indicate that no QoS path has been found. All $d[src, i]$, $1 \leq i \leq X$ must set to be 0.

By limiting the granularity of the QoS metrics, the limited granularity heuristic has a time complexity of $O(X|V||E|)$. The major concern about this heuristic is the effectiveness of this heuristic in finding paths that satisfy the QoS constraints. 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 scheme in [1] uses $r_i = i * \frac{c_1}{X}$ while [4] proposes other mapping schemes. Which mapping scheme can yield better result? 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

RELAX(u, v, w)
for each $d[v,i] \in PATH(v)$
  Let j be the largest j such that $r_j < r_i - w_1(u,v)$
  if $(d[v,i] > d[u,j] + w_2(u,v))$ then
    $d[v,i] = d[u,j] + w_2(u,v)$

Limited_Granularity_Heuristic(G, w, c, src, dst)
For $i = 0$ to $|V(G)| - 1$
  For $j = 1$ to X
    $d[i,j] = \infty$
For $j = 1$ to X
  $d[src,j] = 0$
For $i = 1$ to $|V(G)| - 1$
  For each edge $(u,v) \in E(G)$
    RELAX(u, v, w)
  if $(d[dst,X] < c_2)$ then return "yes"
return "no"

Figure 4: The limited granularity heuristic

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: Let $p = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_L$ be a path that satisfies the QoS constraints $w_1(p) \leq c_1$ and $w_2(p) \leq c_2$. Let the range $[0, c_1]$ be approximated by X discrete 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 $r_{i-1} < x \leq r_i$.

Let $p(i)$ denote the path $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_i$. By induction, it can be shown that $aw_1(p(i)) \geq r_i$. Base case, when $i = 1$, since r_1 is the smallest value that can be used to approximate, $aw_1(v_0 \rightarrow v_1) \geq r_1$. Assuming that $aw_1(p(i-1)) \geq r_{i-1}$, $aw_1(p(i)) = aw_1(p(i-1)) + aw_1(v_{i-1} \rightarrow v_i) \geq r_{i-1} + w_1(v_{i-1} \rightarrow v_i) > r_{i-1}$. Thus, $aw_1(p(i)) \geq r_i$. $aw_1(p(X)) \geq r_X = c_1$. When $L > X$, $aw_1(p) > c_1$ and the heuristic cannot recognize that the path p satisfies the QoS constraints. $\square$

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

Proof: Let us first consider the first half of the theorem: the limited granularity heuristic with mapping $r_i = i * \frac{c_1}{X}$, for all $1 \le i \le X$, can guarantee finding a path that satisfies the QoS constraints. The mapping scheme is called *uniform mapping scheme.*

When $r_i = i * \frac{c_1}{X}$, $aw_1(u \to v) < w_1(u \to v) + \frac{c_1}{X}$. Consider the path p, we can prove that the limited granularity with uniform mapping scheme can guarantee finding a QoS path by showing that $aw_1(p) \le c_1$.

$$aw_1(p) = \sum_{(u \to v)\ on\ p} aw_1(u \to v) < \sum_{(u \to v)\ on\ p} (w_1(u \to v) + \frac{c_1}{X})$$

$$= \sum_{(u \to v)\ on\ p} w_1(u \to v) + \sum_{(u \to v)\ on\ p} \frac{c_1}{X} \le c_1 - \frac{L-1}{X} * c_1 + \frac{L}{X} = c_1 + \frac{c_1}{X}$$

Thus, $aw_1(p) < c_1 + \frac{c_1}{X}$. Since $aw_1(p) = \sum_{(u \to v)\ on\ p} aw_1(u \to v)$ only takes values in the set $\{\frac{c_1}{X}, 2 * \frac{c_1}{X}, 3 * \frac{c_1}{X}, ...\}$, $aw_1(p) \le c_1$. Thus, the limited granularity heuristic with the uniform mapping scheme can guarantee finding a path that satisfies the QoS constraints.

Now, consider the second half of the theorem, the limited granularity heuristic using other mapping schemes cannot guarantee to find a path that satisfies the 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. $w_1(v_i \to v_{i+1}) = \frac{c_1}{2*X^2}$, for all $0 \le i \le X-1$. Consider the path $p = v_0 \to v_1 \to ... \to v_X$. $length(p) = X$. $w_1(p) = X * \frac{c_1}{2*X^2} = \frac{c_1}{2X} \le c_1 - \frac{X-1}{X} * c_1$. Thus, using the uniform mapping, the limited granularity heuristic can detect this path. When using other mapping scheme, $aw_1(p) \ge X * a_1 > c_1$ 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 $w_1(v_i \to v_{i+1}) = \frac{j*\frac{c_1}{X} - a_j}{2*(X-j)}$, for all $1 \le i \le X - j$. $w_1(v_0 \to v_1) = \frac{a_j + j * \frac{c_1}{X}}{2}$. Consider the path $p = v_0 \to v_1 \to ... \to v_{X-j+1}$, $w_1(p) = \frac{a_j + j*\frac{c_1}{X}}{2} + (X - j) * \frac{j*\frac{c_1}{X} - a_j}{2*(X-j)} = \frac{j}{X} c_1 \le c_1 - \frac{X-j+1-1}{X} c_1$. In the non-uniform mapping scheme, $aw_1(v_0 \to v_1) = a_{j+1}$. 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.

In all cases, the limited heuristics algorithm cannot guarantee finding the path p that satisfies

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

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 we maintain $100 * N$ entries in each node, we can guarantee to find a path p when $w_1(p) \leq 0.99c_1, w_2(p) \leq c_2$, which is very close to the original QoS constraints $w_1(p) \leq c_1, w_2(p) \leq c_2$. The precision of the algorithm grows together with the number of entries maintained. 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. 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. If the heuristic can record all optimal QoS paths in each node, it can perform as good as $EBFA$. 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: Let path p be the path whose w_1 weight is the smallest among all the P paths from src to dst. The path p is an optimal QoS path because no other paths can have smaller w_1 weights. All paths whose w_2 weights are larger than $w_2(p)$ are not optimal QoS paths. Let SR be the set of the paths whose w_2 weights are smaller than $w_2(p)$. SR may contain other optimal QoS paths. If $w_2(p)$ is ranked as the pth smallest among the P paths, the size of SR will be $p - 1$. Since w_2 weight is independent of the w_1 weight and all paths are of equal lengths, $w_2(p)$ can be ranked the anywhere in the ranking among the P paths with equal probability. Thus, after p is selected, the size of SR can be 0, 1, ..., P-1, with equal probability $\frac{1}{P}$. Let $T(P)$ be the average number of optimal QoS paths in the P paths. The following formulae can be obtained:

$$T(1) = 1, T(P) = 1 + \frac{T(P-1)}{P} + \frac{T(P-2)}{P} + ... + \frac{T(1)}{P}$$

From the above formulae, we can prove by induction that

$$T(P) \leq \sum_{i=1}^{P} \frac{1}{i}.$$

Base case $P = 1$,

$$T(1) = 1 \leq \sum_{i=1}^{1} \frac{1}{i}.$$

Induction case, assuming that

$$T(N) \leq \sum_{i=1}^{N} \frac{1}{i}, for\ all\ N < P.$$

$$\begin{aligned}
T(P) &= 1 + \tfrac{T(P-1)}{P} + \tfrac{T(P-2)}{P} + ... + \tfrac{T(1)}{P} \\
&\leq 1 + \tfrac{(1+\frac{1}{2}+...+\frac{1}{p-1})}{P} + ... + \tfrac{1}{P} \\
&= 1 + \tfrac{P-1}{1*P} + \tfrac{P-2}{2*P} + \tfrac{P-3}{3*P} + ... + \tfrac{1}{(P-1)*P} \\
&= 1 + (\tfrac{1}{1} - \tfrac{1}{P}) + (\tfrac{1}{2} - \tfrac{1}{P}) + (\tfrac{1}{P-1} - \tfrac{1}{P}) \\
&= 1 + \tfrac{1}{2} + \tfrac{1}{3} + ... + \tfrac{1}{P-1} + 1 - \tfrac{P-1}{P} \\
&= 1 + \tfrac{1}{2} + \tfrac{1}{3} + ... + \tfrac{1}{P-1} + \tfrac{1}{P} = \textstyle\sum_{i=1}^{P} \tfrac{1}{i}
\end{aligned}$$

As shown in [3], $\sum_{i=1}^{P} \frac{1}{i} = ln(P) + O(1)$. Thus, $T(P) \leq ln(P) + O(1)$. □

Given an N–node graph with 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 weight 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?

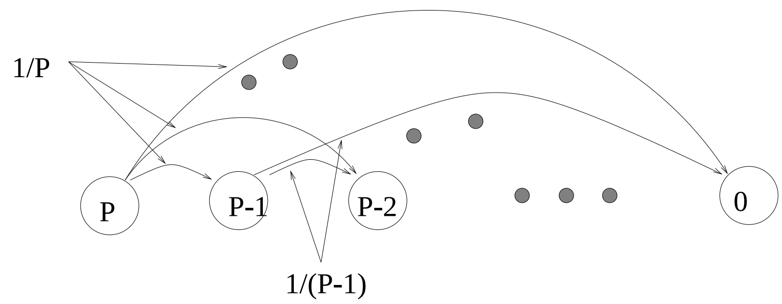


Figure 5: The Markov Chain

This problem can be modeled as a Markov process with $P+1$ states, 0, 1, ..., P, as shown in Figure 5. 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 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. $A^1(P,0) = \frac{1}{P}$, $A^2(P,0) = \frac{1}{P}(\frac{1}{P-1} + \frac{1}{P-2} ... + \frac{1}{1}) = \frac{1}{P}\sum_{i=1}^{P-1}\frac{1}{i}$ and

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

Lemma 4:

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

Proof: First we prove the following formula that will be used in the proof of the lemma.

$$\sum_{i=2}^{N}\frac{(ln(i))^k}{i} \leq \frac{(ln(N))^{k+1}}{k+1}$$

$\sum_{i=2}^{N}\frac{(ln(i))^k}{i} \leq \int_1^N \frac{(ln(x))^k}{x}dx = \int_1^N \frac{d(ln(x))^{k+1}}{k+1} = \frac{1}{k+1}(ln(x))^{k+1}|_1^N = \frac{(ln(N))^{k+1}}{k+1} - 0 = \frac{(ln(N))^{k+1}}{k+1}$

Armed with this formula, we will now prove the theorem.

$$
\begin{aligned}
A^i(P,0) &= \frac{1}{P}\sum_{l_1=i}^{P-1}\frac{1}{l_1}\sum_{l_2=i-1}^{l_1}\frac{1}{l_2}\sum\cdots\sum_{l_i=1}^{l_{i-1}}\frac{1}{l_i}\\
&\le \frac{1}{P}\sum_{l_1=i}^{P-1}\frac{1}{l_1}\sum_{l_2=i-1}^{l_1}\frac{1}{l_2}\sum\cdots\sum_{l_{i-1}=2}^{l_{i-2}}\frac{ln(l_{i-1})+1}{l_{i-1}}\\
&\le \frac{1}{P}\sum_{l_1=i}^{P-1}\frac{1}{l_1}\sum_{l_2=i-1}^{l_1}\frac{1}{l_2}\sum\cdots\sum_{l_{i-1}=2}^{l_{i-2}}\frac{2ln(l_{i-1})}{l_{i-1}}\\
&= \frac{2}{P}\sum_{l_1=i}^{P-1}\frac{1}{l_1}\sum_{l_2=i-1}l_1\frac{1}{l_2}\sum\cdots\sum_{l_{i-1}=2}^{l_{i-2}}\frac{ln(l_{i-1})}{l_{i-1}}\\
&= \frac{2}{P}\times\frac{1}{2\times 1}\sum_{l_1=i}^{P-1}\frac{1}{l_1}\sum_{l_2=i-1}^{l_1}\frac{1}{l_2}\sum\cdots\sum_{l_{i-2}=3}^{l_{i-3}}\frac{(ln(l_{i-2}))^2}{l_{i-2}}\\
&= \ldots\\
&= \frac{2}{P}\times\frac{1}{(i+1)!}\times(ln(P))^{i+1} = \frac{(ln(P))^{i+1}*2}{P*(i+1)!}
\end{aligned}
$$

□

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

Proof: This proof use the following formula $n! \ge \sqrt{2\pi n}(\frac{n}{e})^n$ from [3]. When $i > e^2 ln(P)$,

$$
\begin{aligned}
A^i(P,0) &= \frac{(ln(P))^{i+1}*2}{P*(i+1)!} \le \frac{2}{P}\frac{(ln(P))^{i+1}}{\sqrt{2\pi(i+1)}(\frac{i+1}{e})^{i+1}} = \frac{2}{P\sqrt{2\pi(i+1)}}\left(\frac{eln(P)}{i+1}\right)^{i+1}\\
&\le \frac{2}{P\sqrt{2\pi(i+1)}}\left(\frac{1}{e}\right)^{ln(P^{e^2})} \le \frac{2}{P\sqrt{2\pi(i+1)}}\frac{1}{P^{e^2}} \le \frac{1}{P^{e^2+1}}.
\end{aligned}
$$

□

Theorem 2: Given an N node graph with nodal degree of $d \le N$, 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 $i > e^2 lg(N^L)$,

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

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. □

Theorem 2 states that the probability is high, but it does not specify how high it would be. As can be seen from the proof, the probability also depends on P. For example, if $P = 100$, the probability is more than 99.9999999%. 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.

Corollary 1: Given an N–node graph with nodal degree of $d \le N$, the limited path heuristic has a very high probability to find paths of all possible lengths that satisfy the QoS constraints, when each node maintains $O(N^2 lg(N))$ paths.

Proof: Theorem 2 states to by maintaining $O(Llg(N))$ entries in the limited path heuristic, it has a very high probability to find all QoS paths of length L. In an N–node graph, the length of any QoS path must be between 1 and N. Thus, maintaining $\sum_{L=1}^{N} O(Llg(N)) = O(N^2lg(N))$ paths in each node will give very high probability for the limited path heuristic to record all optimal QoS paths. Thus, the limited granularity heuristic has a very high probability to find a path that satisfies the QoS constraints when such a path exists, when each node maintains $O(N^2lg(N))$ paths. $\Box$

When $X = O(N^2lg(N))$, the time complexity of the heuristic is $O(N^5(lg(N))^2|E|)$. Notice that the bound $O(N^2lg(N))$ is not tight. However, it establishes that the limited path heuristic can perform almost as good as $EBFA$ in polynomial time.

5 Experiments

This section studies and compares the two heuristics through simulation. The mesh topology is used in the study. One example of the mesh topology is shown in figure 6. Since the performance of the two heuristics is closely related to the distance of the paths, the study chooses the source and the destination to be furthest apart in the mesh topology as shown in Figure 6. 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 $[0.0, 20.0]$. We compare the two heuristics 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 simulation results. The two concepts are defined as follows.

$$existence\ percentage = \frac{total\ \#\ of\ requests\ satisfied\ using\ the\ exhaustive\ algorithm}{total\ \#\ of requests}$$

$$competetive\ ratio\ = \frac{total\ \#\ of\ requests\ satisfied\ using\ the\ heuristic}{total\ \#\ of\ requests\ satisfied\ using\ the\ exhaustive\ algorithm}$$

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. By definition, both existence percentage and competitive ration are in the range of $[0.0, 1.0]$.

Figure 7 shows the results for 8×8 mesh. The source and the destination is fixed as the two nodes in the diagnose in the mesh as shown in Figure 6. The QoS paths between these two nodes are the most difficult paths to be found in this topology. 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

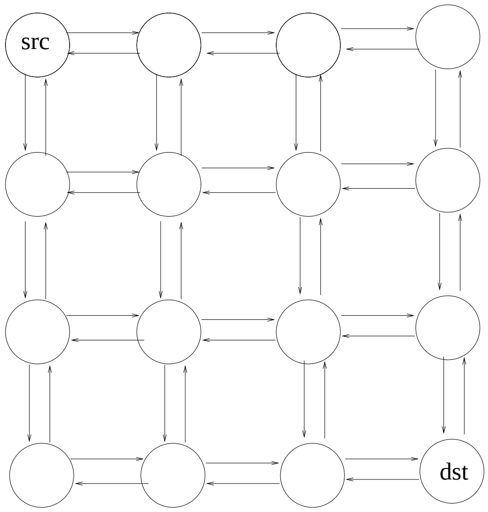
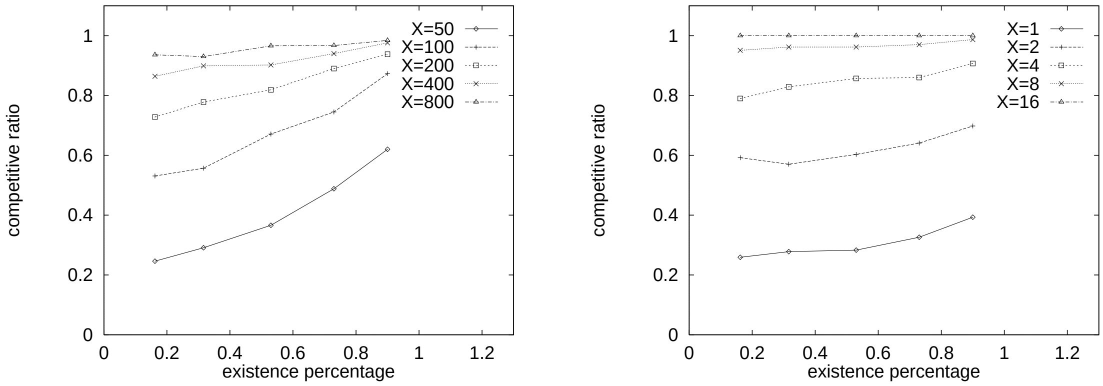


Figure 6: A 4×4 mesh



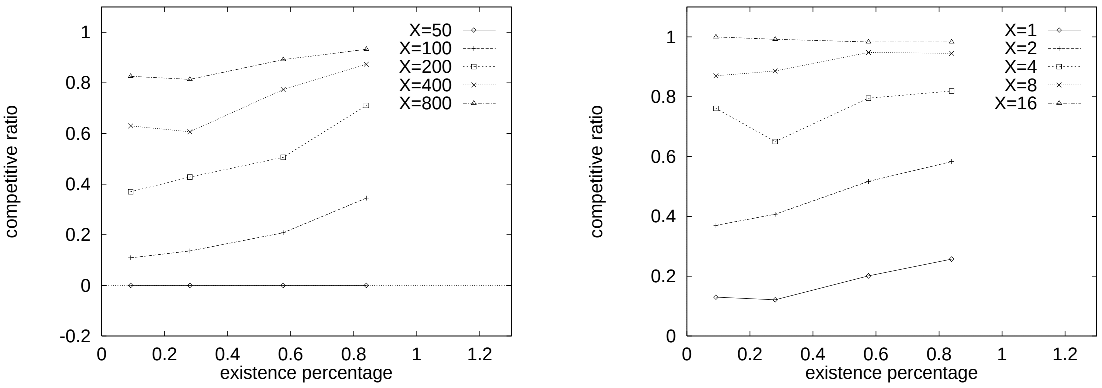
(a) Limited granularity heuristic

(b) Limited path heuristic

Figure 7: Results for 8×8 mesh

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 execution time than the limited granularity heuristic.

Figure 7 also shows that the performance of the limited path heuristic is affected mainly by the number of the entries maintained in each node but not by how difficult the paths can be found, while the performance of the limited granularity heuristic is affected by both. This is indicated by the flatter curves for the limited path heuristic in Figure 7 (b) than those for the limited granularity heuristic in Figure 7 (a). The limited granularity heuristic tends to find paths that can be found with ease while missing the ones that are hard to find.



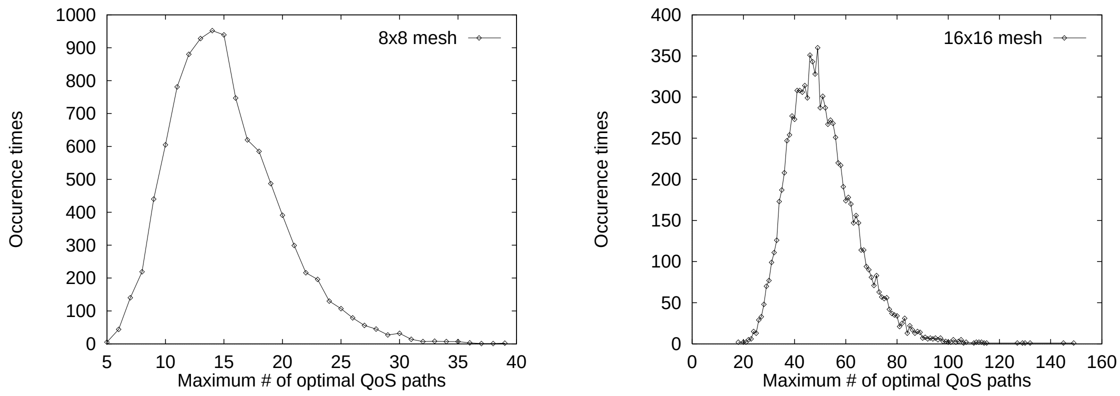
(a) Limited granularity heuristic

(b) Limited path heuristic

Figure 8: Results for 16×16 mesh

Figure 8 shows the results for 16×16 mesh. 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.

The previous two experiments show that the limited path heuristic performs very well in the random mesh topology. This experiment tries to reveal the exact number of entries to be maintained in each node for the limited path heuristic to achieve high performance by showing the maximum number of entries actually maintained in each nodes in EBFA. If the limited path heuristic can maintain the same number of entries, it will perform exactly like EBFA. In the experiment, the QoS constraints are set to be very large values so that all optimal QoS paths are maintained. Figure 9 shows the results for 8×8 and 16×16 meshes. Note that when counting the maximum number of entries, all nodes in the graph are considered, not



(a) 8×8 mesh

(b) 16×16 mesh

Figure 9: Results for 8×8 mesh with random destination

just the destination. The x–axis shows the maximum number of entries, that is, the maximum number of optimal QoS paths to any node in the graph. The y–axis shows the number of times the maximum number of optimal QoS paths occur in 10000 trials. As shown in the figure, most of the 8×8 mesh has a maximum number of optimal QoS paths between 5 to 30 and most of the 16×16 mesh has a maximum number of optimal QoS paths between 20 to 100. By maintaining these numbers (30 for 8×8 mesh and 100 for 16×16 mesh) of paths in the limited path heuristics, the heuristic can perform both effectively (like EBFA) and efficiently. Actually, the limited path heuristic does not need to maintain all the optimal QoS paths to be effective in practice as shown in Figure 8 (b).

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 does not consider the cases when the mapping schemes are not fixed. 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 the two–constrained QoS path routing problem. 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 the two–constrained QoS path routing problem. The study also shows that in order for the limited granularity heuristic to be effective, the number of entries to be maintained in each node should be fairly large, which renders high time and space complexity for the limited granularity heuristic. 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, University of California at Berkeley.