Least-cost High-quality Object Retrieval for Distributed Multimedia Collaborations *

K. Selçuk Candan Yifeng Yang
Computer Science and Engineering Department,
Arizona State University
Email: {candan, yifeng.yang}@asu.edu

Abstract

In a collaborative network, retrieval of media objects from corresponding servers introduces many challenges. A major challenge is the multi-objective route establishment requirement, where various QoS and object quality parameters should be simultaneously optimized. This task requires new quality-based routing algorithms. In this paper, we present optimistic, pessimistic, and hybrid algorithms that optimize the media delivery costs while maximizing the quality of the transmitted objects. We also report on simulation results to show applicability of these algorithms for varying sizes of collaborative networks. These algorithms can be implemented either as sub-application-layer solutions in an information-centric network or as application-layer information sharing/communication protocols between collaborators.

1. Introduction

One of the most important opportunities introduced by the increasing connectivity around the world is the *wide area collaborative work*. On the other hand, moving to a media based collaborative environment is not a trivial task. One reason for this is that collaborations mostly occur in heterogeneous environments: collaborators may have different needs, preferences; they may have different hardware/software capabilities; or they may observe different network loads. Consequently, in a collaborative environment, retrieving non-local objects through a network is challenging. For this purpose, network characteristics of multimedia objects need to be fully explored and efficient protocols need to be developed for providing fast and high quality multimedia communication. Such protocols must address the heterogeneous capabilities and needs of different users, and they need to shape the multimedia object to fit into the available resources. We expect that, as the distinction between data and information blurs through new media standards, such as MPEG4, object delivery systems will serve as information rather than data networks.

Above challenges lead us to the observation that *a collaborative multimedia system should be flexible enough to handle the heterogeneity which is inherent to a multi-user distributed system*. This flexibility must be present at all levels of the system: specification of the multimedia documents, storage and retrieval of multimedia objects from the sources, and delivery of the objects.

In this paper, we develop a set of algorithms that, in a collaborative network, find media object delivery routes that minimize the delivery cost as well as maximize the *quality* of the delivered objects. The proposed algorithms can either be implemented as part of the network layer or can be implemented as an application layer solution within collaborating softwares. For this purpose, we introduce an extended network model capable of capturing the effect of object size changes and quality transformations on the transmission cost. This model can also capture the effects of transformations required at the network boundaries (between high/low bandwidth networks), though, in general, such boundary transformations is expected to be limited to a few network nodes.

2. Collaborative Environment

A collaborative environment consists of a set of users (U), a set of sources (S), a set of media-objects (O^{med}), and a network ($G = (V, E, \wp)$) which connects the users (collaborators) with each other and the sources. In G, $U \cup S \subseteq V$, E is the connectivity, and $\wp : E \rightarrow \mathbf{R}^+$ specifies the cost of sending a byte of information along an edge in the graph.

In a collaborative network, user and server nodes, as well as the intermediary network nodes, can per-

*This work was supported by an ASU-FGIA Grant F-014-98. We thank Arunabha Sen and Prasad Chalasani for their helpful comments and inputs.

form various conversions on media objects. These conversions consist of various types of operations, such as *format conversion*, *quality reduction/ enhancement*, or *compression/ decompression*. We use the mapping $\mathcal{CAP}: V \rightarrow 2^{\mathcal{HC}}$ to specify what media processing and conversion capabilities are available at a given node.

Note that operations performed on media objects may change sizes as well as qualities of media objects. Given a conversion function, $cf_i \in \mathcal{HC}$, $cost_i$, $size_i$, and $quality_i$, denote the unit cost, size change ratio, and the quality change ratio associated with cf_i. Given an object, o, of size s_o and quality q_o, the result of the conversion is another object of size $size_i \times s_o$, quality $quality_i \times q_o$. The cost of the conversion operation is $cost_i \times s_o$.

If a user u_i wishes to access some object o, he must be able to (1) retrieve this object from sources through the network, (2) ensure that the object is in a form that is compatible with the viewing/editing resources he has, and (3) ensure that the object has the desired (minimum) quality (such as image size and resolution). Furthermore, the user must be able to achieve these goals at the lowest possible cost/delay.

To account for media related requirements, we are extending the network layer with various capabilities, including *single/multi-client single/multi-server routing*, *immediate and advance resource reservation*, *adaptive re-routing and reservation modification*, and *multi-criteria route establishment*. When these services are not provided by the underlying network, application level counterparts can be implemented. Such application level solutions to media communication include [1], in which Amir, McCanne, and Zhang propose to decompose a given video transmission into multiple sessions, each with different bandwidth requirements. Other techniques for scalable media transmission include source channel coding. In [6], Ferrari *et al.* discusses various issues related with network support for multimedia.

In this paper, we focus on the multi-criteria route establishment problem.

3. Related Work

Open Shortest Path First (OSPF)[11] has the ability to detect any change in the network and recalculate a new and more efficient source-destination path. Although OSPF does not provide quality of service guarantees, it provides different types of service. The cost of a route is given by a single metric which can be *delay, throughput*, or *reliability*. To guarantee a specified QoS to a source-destination flow, the network needs to set aside some of its resources for allocation to that flow. Recognizing the need for routing techniques that guarantee certain quality of service, there has been significant activity in recent years [12, 8, 9]. If the aim is to create a single plan which optimizes all of these parameters simultaneously, the optimal route construction problem must be viewed as a multi-criteria optimization problem.

The problem of finding an optimal route satisfying such a diverse set of criteria is often intractable (NP-Complete) [13]. The two-criteria decision problem is known to be NP-complete [7]. Loui [10] proposed algorithms for identifying optimal paths in stochastic graphs. Daellenbach and De Kluvyer discussed how dynamic programming can be used to solve multi-objective problems [5]. Warburton [14] proposed approximation algorithms for multi-objective shortest-path problems. Earlier, as part of the Collaborative Heterogeneous Interactive Multimedia Platform (CHIMP) [2, 3], we proposed a pseudopolynomial time algorithm for a problem related to multi-objective route establishment.

4. Collaborative Network

In a media-intensive environment, object retrieval parameters include acceptable ranges for delivery time as well as QoS and object quality parameters. Network must establish a delivery plan such that the object transmission cost is minimized, and the final object quality (e.g., frame rate, resolution) is maximized. Consequently, routing algorithms must take into account these objectives and perform multi-objective routing. In Sections 5 and 6, we propose algorithms for multi-objective route establishment.

Given a collaborative network $G = (V, E, \wp)$ and node capabilities $\mathcal{CAP}: V \rightarrow 2^{\mathcal{HC}}$, and two nodes s (source) and d (destination), we create a new graph $G' = (V', E', cost', size\_ratio', qual')$, such that

- $V' = (V \times \mathcal{HC}) \bigcup \{s', d'\}$,
- $E_1 = \{\langle v_1, cap_1\rangle \rightarrow \langle v_2, cap_2\rangle |$ $(v_1 \rightarrow v_2 \in E) \wedge (cap_1 \in CAP(v_1)) \wedge (cap_2 \in CAP(v_2))\}$
- $\forall e_j \in E_1, \wp'(e_j) = \wp(e_i) + cost(cap_1)$, where $(e_i = v_1 \rightarrow v_2 \in E) \wedge \langle e_j = \langle v_1, cap_1\rangle \rightarrow \langle v_2, cap_2\rangle \rangle$,
- $\forall e_j \in E_1, size\_ratio'(e_j) = size(cap_1)$, where $(e_i = v_1 \rightarrow v_2 \in E) \wedge \langle e_j = \langle v_1, cap_1\rangle \rightarrow \langle v_2, cap_2\rangle \rangle$.
- $\forall e_j \in E_1, qual'(e_j) = qual(cap_1)$, where $(e_i = v_1 \rightarrow v_2 \in E) \wedge \langle e_j = \langle v_1, cap_1\rangle \rightarrow \langle v_2, cap_2\rangle \rangle$.
- $E_2 = \{\langle s, \perp\rangle \rightarrow \langle s, cap_i\rangle | \langle s, cap_i\rangle \in V'\}$
- $\forall e_j \in E_2, \wp'(e_j) = 0$; $\forall e_j \in E_2, size\_ratio'(e_j) = 1$; and $\forall e_j \in E_2, qual'(e_j) = 1$.
- $E_3 = \{\langle d, cap\_i, \perp\rangle \rightarrow \langle d, \perp\rangle | \langle d, cap_i\rangle \in V'\}$
- $\forall e_j \in E_3, \wp'(e_j) = 0$; $\forall e_j \in E_3, size\_ratio'(e_j) = 1$; and $\forall e_j \in E_3, size\_ratio'(e_j) = 1$.

- $E' = E_1 \bigcup E_2 \bigcup E_3$ and $cost' = \wp'$.
- $s' = \langle s, \perp \rangle$ and $d' = \langle d, \perp \rangle$.

G' describes the alternative transmission and conversion options as well as the corresponding unit cost and size change ratio. Each node in G' corresponds to a conversion function/ node pair. Since, there are multiple nodes in G' which correspond to s and d, we add two virtual nodes, s' and d', that represent all source and destination alternatives. Note that, if there are multiple sources which may deliver a requested object, possibly with different initial sizes and qualities, we can also attach such alternative object sources to s' with edges labeled with initial object size and qualities.

The final size (the quality) of the object delivered to node, n, depends on the $size\_ratio$ ($qual$) weights associated with the edges on the delivery path: the final object size (quality) is calculated by multiplying the corresponding weights of the edges on the path. The total delivery cost, on the other hand, is the sum of the delivery costs of the object over each edge on the path. Note that the cost of delivering an object, o, over an edge, e, depends on the size of the object at the node $source(e)$, which itself depends on the delivery path. *Consequently, since the object delivery cost of a given edge varies depending on the route followed by a particular object, the well-known shortest path algorithms [4] can not be applied to find least cost retrieval plans.* Next, we define terminology that we will use in our algorithms.

Definition 4.1 (Size of a vertex wrt. a path) Let $G' = (V, E, cost, size\_ratio, qual)$ be a graph described in the previous section and let for each edge $e \in E$, $size\_ratio(e) \in R^+$ be the corresponding size change ratio. Let $P = p_0 \rightarrow \ldots \rightarrow p_l$ denote a simple directed path in G' from p_0 to p_l.

The *size of p_l with respect to path P* is defined as

$$\sigma_P(p_l) = \prod_{e \in P} size\_ratio(e).$$

Definition 4.2 (Cost of an edge wrt. a path) Let $G' = (V, E, cost, size\_ratio, qual)$ be a directed graph and let for each $e \in E$, $size\_ratio(e) \in R^+$ be the corresponding size ratio and $cost(e) \in Z^+ \cup \{0\}$ be the unit cost of the edge. Let $P = p_0 \rightarrow \ldots \rightarrow p_l$ denote a simple directed path in G' from p_0 to p_l and $\epsilon \in E$ be an edge from p_l to $p_j \in V$.

The *cost of edge ϵ with respect to path P* is defined as

$$\delta_P(\epsilon) = cost(\epsilon) \times \sigma_P(p_l).$$

The *least cost high quality retrieval problem* can be stated as follows:

Definition 4.3 Given

- a network, $G' = (V, E, cost, size\_ratio, qual)$, where for each $e \in E$
  - $cost(e) \in Z^+ \cup \{0\}$ is a the unit cost,
  - $size\_ratio(e) \in R^+$ is the size change ratio,
  - $qual(e) \in [0.0, 1.0]$ is the quality change ratio,
- source and destination vertices $s, t \in V$, and
- specified vertices $s, t \in V$,

is there a *simple* path, $P = s \rightarrow \ldots \rightarrow t$, in G', of total cost (T_P) and quality (T_Q),

$$T_P = \sum_{e=(p_i \rightarrow p_j) \in E \; and \; on \; P} \delta_{P(p_i)}(e),$$

$$T_Q = \prod_{e=(p_i \rightarrow p_j) \in E \; and \; on \; P} qual(e),$$

such that $T_P \leq K$ and $T_Q \geq Q$. Note that for a given path, P, described as a sequence of edges as $\langle e_0, e_1, \ldots, e_k \rangle$, in G', the total cost is equal to

$$T_P = \sum_{i=0}^{k} cost(e_i) \times \prod_{j=0}^{i} size\_ratio(e_i).$$

Theorem 4.1 The least cost high quality retrieval problem is NP-hard. $\diamond$

The detailed proof of the above theorem is omitted due to space considerations. The proof is by a reduction from the n (≥ 2) shortest path problem.

In the next sections, we describe heuristic algorithms and three different strategies for creating a least-cost and least-cost high quality retrieval plan.

5. Least-cost Routing

The simplest of the two routing algorithms, the least-cost routing algorithm (LCRA), assumes that conversions do not change the quality of the object. Consequently, it annotates each node, n_i, in G', with two variables: size, s_i, which is calculated by multiplying the size change ratios of the edges on the path from the source to n_i ($s_i = \prod_{e_k \; on \; p}(size\_ratio(e_k))$), and cost, c_i, which is calculated by adding the edge traversal costs on this path ($c_i = \sum_{e_k(n_u \rightarrow n_v) \; on \; p}(s_u \times cost(e_k))$).

Next, we describe a heuristic least-cost retrieval algorithm ($LCRA$) and in Section 7, we provide simulation results that show how well the algorithm performs. (Note: we use the "*size of an object at node n_i*" and "*size of node n_i*" interchangeably.)

- Let s' be the start, st, and d' be the destination, $dest$.

- *For every pair of nodes, n_i and n_j, in G', find $\delta^{\ell}_{i,j}$ ($\delta^{s}_{i,j}$), which describes the largest (smallest) possible change in the size of object at node n_j, given a unit increase in the size of object at node n_i.* For this purpose, we use a modified version of Bellman-Ford algorithm([4]) that takes into account multiplicative distances and removes negative cycles.
- *Find the largest (and smallest) cost change, Δ^{ℓ}_i (Δ^{s}_i) at the node dest that a unit increase in the size of an object at a given node, n_i, can cause.* We do this by assigning the value $\delta^{\ell}_{i,j} \times cost_e$ ($\delta^{s}_{i,j} \times cost_e$), where $cost_e$ is the edge cost factor) to each edge on the way from n_i to the terminal node and using the shortest path algorithm described above.
- *Find the smallest cost from the source node, st, to the destination node, dest, using a modified version of the Dijkstra's algorithm*: Dijkstra's algorithm ([4]) is based on a concept called *edge relaxation*, where given two nodes, n_i and n_j, of respective costs, c_i and c_j, and an edge, e_l, from n_i to n_j, with an associated cost, $cost(e_l)$, the edge e_l is traversed if this reduces the cost, c_j, of n_j; i.e., if $c_i + cost(e_l) < c_j$.

If we know that it will be useful in the long run, we would like to relax an edge even if it causes temporary increase in the cost. Similarly, if we know that that it may lead a large cost increase in the long run, we would like to skip relaxing an edge even if the relaxation would cause a temporary decrease in the cost.

To account for this, we modify the relaxation condition of an edge e_l from n_i to n_j as follows: $(c_i + (s_i \times cost(e_l))) < c_j + ((size\_ratio(e_l) \times s_i) - s_j) \times ((w \times \Delta^{\ell}_j) + ((1-w) \times \Delta^{s}_j))$, where c_i and c_j are current costs and s_i and s_j are current sizes of the nodes; $cost_e$ and $size\_ratio_e$ are the cost and size_ratio factors of the edge; Δ^{ℓ}_j and Δ^{s}_j are the smallest and largest cost change factors for the node n_j; and w is a real number between 0.0 and 1.0. After the application of the edge, the c_j changes to $c_i + (s_i \times cost(e_l))$ and s_j changes to $size\_ratio(e_l) \times s_i$.

When w is 1.0, the relaxation considers only the largest (or the worst case) increase in the cost; hence, we call this case the **pessimistic strategy**. When w is 0.0, it considers the smallest (best case) increase, and we call this case the **optimistic strategy**. For any other value of w, the result falls somewhere between the best and worst cases, and we call this case the **w-hybrid strategy**.

Assuming a graph without negative cycles, the complexity of Dijkstra's algorithm is known to be $O((v + e) \times \log v)$, where v is the number of nodes and e is the number of edges. The complexity of the second step of the algorithm (finding the largest and smallest increases for every pair of nodes can be done by a modified version of the Floyd-Warshall's all-pairs shortest path algorithm which is $O(v^3)$. The third step, then, requires Bellman-Ford's shortest-path algorithm which is $O(v \times e)$. Hence, the running time of the above algorithm is $O(v^3 + v \times e)$. In Section 7, we report on the results of this algorithm.

6. Least-cost High-quality Routing

Users may wish to simultaneously achieve two objectives: (1) minimization of the retrieval cost and (2) maximization of the object qualities. However,these two goals may conflict with one another. Below, we define a goal which merges these two conditions:

- *minimize penalty:* the query executed in a way that minimizes a penalty function of the form $\mathcal{P}(c,q) = \alpha \times c - \beta \times q$, where c is the retrieval cost, q is the quality of the results, and α and β are positive integers.

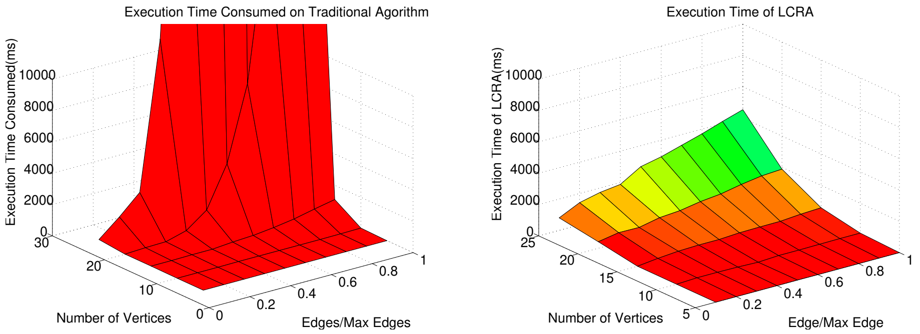
We can also use other penalty functions, such as $\mathcal{P}(c,q) = \alpha \times c + \beta \times \log(q)$, which take into account the exponentially decreasing nature of quality.

We can not directly replace the cost function we used in $LCRA$ algorithm with the above penalty function: cost is additive, whereas the quality is multiplicative. Instead, we annotate each node, n_j, with two new variables:q_j, to account for the quality ($q_j = \prod_{e_k\, on\, the\, path\, to\, n_j}(qual(e_k))$) and $p_j = \alpha \times c_j - \beta \times q_j$. Note that, unlike the edge cost, the quality change ratio, $qual(e_k)$, of an edge, e_k, is independent of the input size.

- We identify the start node, st, and the destination node, $dest$.
- For every pair of nodes, n_i and n_j, in the G', we find $\delta^{\ell}_{i,j}$ ($\delta^{s}_{i,j}$).
- We find the largest (and smallest) cost change, Δ^{ℓ}_i (Δ^{s}_i) at the node $dest$ that a unit increase in the size of an object at a given node, n_i, can cause. Note that, since the quality ratio is not dependent on the input size, this value is equivalent to the largest (and smallest) penalty change, ΔP^{ℓ}_i (ΔP^{s}_i), at $dest$ that a unit increase in the size of a given node, n_i, can cause.
- We apply the modified shortest path algorithm with the following relaxation condition (of a given edge e_l from n_i to n_j):

$$(p_i + (\alpha \times s_i \times cost(e_l) - \beta \times i_i \times qual(e_l))) < (p_j + ((size\_ratio(e_l) \times s_i) - s_j) \times ((w \times \Delta P^{\ell}_j) + ((1-w) \times \Delta P^{s}_j)),$$

where w is a real number between 0.0 and 1.0. We use this constant, as in the case of the $LCRA$, to define *optimistic*, *pessimistic*, and *w-hybrid* $LPRA$ approaches.



(a) (b)

Figure 1: Experiment results: (a) Time taken by the brute-force solution, (b) time taken by $LCRA$

7. Experiment Results

In this section, we report on the simulation results we run to evaluate the performance of the algorithms.

We randomly created a set of connected directed networks, G's. We varied the number of nodes in the graph from 5 to 25 and the number of edges from 10% to the 100% of the maximum possible number of edges.[1] The cost, size, and information factors of the edges range between 0.1 and 10, 0.1 and 10, and 0.5 and 1.0, respectively. We ran the experiments on a LINUX platform with a 300MHz processor and 128 MB memory space. We ran each experiment 50 times and averaged the results.

For comparison purposes, we have implemented a brute-force algorithm which explores all possibilities and returns result with the lowest penalty. In Figures 1(a) and 1(b), we compare the execution times of the brute-force and *pessimistic* $LCRA$ algorithms for varying number of nodes' and edges. In this example, we assumed that the cost is the determining factor in optimization criterion. As expected, the brute-force algorithm becomes intractable very quickly.

The penalty function we used for the simulations is $\mathcal{P}(c, q) = \alpha \times c - \beta \times q$. We can rewrite this function as $\beta \times (\gamma \times c - q)$, where $\gamma = \frac{\alpha}{\beta}$. Intuitively, when γ is low, the cost is not very significant, whereas when γ, cost is more significant.

Figures 2(a,b,c) show the ratio of the penalty returned by LPRA and the best possible penalty (found by brute force search) for various graph configurations and γ values. According to these figures, LPRA performs well for large graphs, but not as well for smaller graphs. The reason for this is that in smaller graphs, the number of conversion options is not large, and a single possible retrieval path not discovered by LPRA may have significant impact on the final quality. In larger graphs, since the number of options is larger, the chance of missing significant paths is not as large.

Figures 2(c,d,e), on the other hand, depicts the rank of the result returned by LPRA among all possible network and conversion paths (enumerated by the brute-force search). According to this figure, the rank of the result returned by LPRA is within 1% of all possible paths. In fact, in most cases, it is very close the best solution. However, when the graph is small and quality is important, the rank degrades (however, it is still within 1%).

These results show that the LCRA and LPRA algorithms perform well for multi-objective routing in information networks.

[1]We limited the number of nodes with 25, because the brute force algorithm that we used for evaluation purposes was performing (in terms of time) very poorly for larger values of nodes.

8. Conclusion and Future Work

Retrieving non-local media objects through a network introduces many challenges. One of these challenges is multi-objective route establishment, where various QoS and object quality parameters are simultaneously optimized. In this paper, we described scalable media-object retrieval algorithms. We presented a set of optimistic, pessimistic, and hybrid algorithms that optimize the media delivery costs while maximizing the quality of the transmitted objects. We, finally, reported on a set of experiments that show that algorithms we introduced achieves the goals.

We are currently in the process of developing pseudopolynomial algorithms that solve the least-cost high-quality object retrieval problem. We are also investigating cheaper approximations with bounded errors to the problem. Furthermore, although the general problem is NP-hard, some of its subsets could have polynomial time solutions. We are currently investigating such specific subsets of the problem.

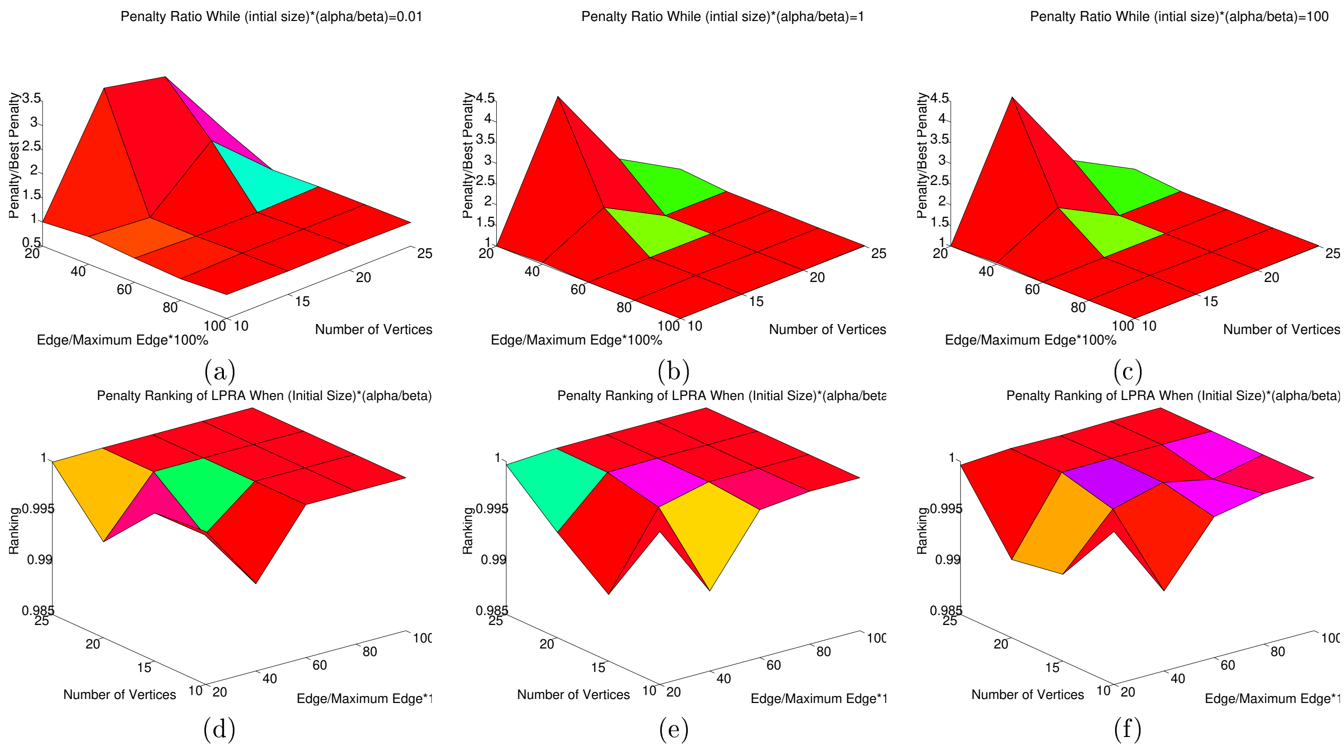


Figure 2: The ratio of penalty returned by LPRA and by brute force search: (a) $\gamma = 0.01$, (b) $\gamma = 1$, and (c) $\gamma = 100$. The rank of the result returned by LPRA: (d) $\gamma = 0.01$, (e) $\gamma = 1$, and (f) $\gamma = 100$

9. References

[1] E. Amir, S. McCanne, and H. Zhang. (1995) *An Application Level Video Gateway*, Proceedings of ACM Multimedia '95, November 5-9, 1995.

[2] K.S. Candan, V.S. Subrahmanian, and P.V. Rangan (1996) *Towards a Theory of Collaborative Multimedia*, IEEE International Conference on Multimedia Computing and Systems, pp. 279-282, Hiroshima, Japan, June 96.

[3] K.S. Candan, V.S. Subrahmanian, and P.V. Rangan (1998) *Collaborative Multimedia Systems: Synthesis of Media Objects.* IEEE Trans. on Knowledge and Data Engineering, Vol. 10, Nr. 3, pp. 433-457, May-June 1998.

[4] T. Cormen, C. Leiserson and R. Rivest. (1990) *Introduction to Algorithms.* MIT Press and McGraw Hill.

[5] H.G. Daellenbach and C.A. De Kluyver. (1980) *Note on Multiple Objective Dynamic Programming*, Journal of Operational Research Society, 31, pp. 591-594, 1980.

[6] D. Ferrari, A. Banerjea, and H. Zhang (1994) *Network Support for Multimedia: A Discussion of the Tenet Approach*, Computer Networks and ISDN Systems, vol. 26, pp. 1267-1280, 1994.

[7] M. Garey and D. Johnson. (1979) *Computer and Intractability: A Guide to the Theory of NP Completeness*, W.H. Freeman, San Fransisco, 1979.

[8] R. Guerin *et al.* (1997) *QoS Path Management with RSVP*, Proceedings of IEEE GLOBECOM'97, Internet Mini-Conference.

[9] R. Guerin *et al.* (1997) *QoS Routing Mechanisms and OSPF Extensions*, Proccedings of IEEE GLOBECOM'97, Internet Mini-Conference.

[10] R. Loui. (1983) *Optimal Paths in Graphs with Stochastic or Multidimensional Weights*, Communications of ACM, 26, pp. 670-676, 1983.

[11] (1994) OSPF version 2, Network Working Group Request for Comments 1583. http://ds.internic.net/rfc/rfc1583.txt.

[12] H. F. Salama, D. S. Reeves, and Y. Viniotis. (1997) *A Distributed Algorithm for Delay-constrainted Unicast Routing*, Proceedings of IEEE INFOCOM'97, pp. 84-91.

[13] Z. Wang and J. Crowcroft. (1996) *Quality of Service Routing for Supporting Multimedia Applications*, IEEE Journal on Selected Areas of Communication, vol. 14, no. 7, September 1996.

[14] A. Warburton. (1987) *Approximation of Pareto Optima in Multiple-Objective, Shortest-Path Problems*, Operations Research, Vol. 35, No.1, pp. 70-79, 1987.

[15] L. Zhang, S. Deering, D. Estrain, S, Shenker and D. Zappala (1993) *RSVP: A New Resource Reservation Protocol*, IEEE Network, September 1993.