Finding a Path Subject to Many Additive QoS Constraints

Guoliang Xue, *SM'99/ACM'93*, Arunabha Sen, *M'xx*, Weiyi Zhang, *S'02*, Jian Tang, *S'04* and Krishnaiya Thulasiraman, *F'90/ACM'90*

***Abstract*— A fundamental problem in Quality-of-Service (QoS) routing is to find a path between a source-destination node pair that satisfies two or more end-to-end QoS constraints. We model this problem using a graph with n vertices and m edges with K additive QoS parameters associated with each edge, for any constant $K \geq 2$. This problem is known to be NP-hard. Fully polynomial time approximation schemes (FPTAS) for the case of $K = 2$ have been reported in the literature. We concentrate on the general case and make the following contributions. (1) We present a very simple $O(Km + n \log n)$ time K-approximation algorithm that can be used in hop-by-hop routing protocols. (2) We present an FPTAS for one optimization version of the QoS routing problem with a time complexity of $O(m(\frac{n}{\epsilon})^{K-1})$. (3) We present an FPTAS for another optimization version of the QoS routing problem with a time complexity of $O(n \log n + m(\frac{\mathcal{H}}{\epsilon})^{K-1})$ when there exists an $\mathcal{H}$-hop path satisfying all QoS constraints. When K is reduced to 2, our results compare favorably with existing algorithms. The results of this paper hold for both directed and undirected graphs. For ease of presentation, undirected graph is used.**

Keywords: QoS routing, multiple additive constraints, efficient approximation algorithms.

I. Introduction

A fundamental problem of routing in a network that provides Quality-of-Service (QoS) guarantees is to find a path between a specified source-destination node pair that simultaneously satisfies multiple QoS constraints, such as cost, delay, and reliability [3], [12], [16], [18], [22]. Such an environment is commonly modeled by a graph with n vertices and m edges where the n *vertices* represent *computers* or *routers* and the m *edges* represent *links*. Each edge has K weights associated with it, representing cost, delay, and reliability, etc. Weights on edges extend to weights on paths in a natural way. If the edge weights represent cost, delay, and reliability, then the corresponding path weight is obtained by adding (multiplying, in the case of reliability) the weights of the edges on the path. For this reason, such QoS parameters are said to be *additive*. QoS parameters such as bandwidth are known as *bottleneck* parameters where the corresponding weight of a path is the smallest of the weights of the edges on the path [8], [22]. Problems involving bottleneck constraints can be easily solved by ignoring all edges whose weights are smaller than a chosen value. Therefore we restrict our attention to additive parameters only.

It is well known that the *multi-constrained path* (MCP) problem is NP-hard, even when the number of constraints is two [22]. Recognizing the need for an efficient solution to this fundamental problem, many researchers have studied this problem in the last few years. Most of the existing works concentrate on the MCP problem with two additive constraints. This special case is known as the *delay constrained least cost* path (DCLC) problem where the two edge weights are *cost* and *delay*, and one seeks a minimum cost path subject to a given delay constraint. Chen *et al.* [3] studied the DCLC problem and proposed a polynomial time heuristic algorithm based on scaling and rounding of the delay parameter so that the delay parameter of each edge is approximated by a bounded integer. Xue [25], [26] proposed to use a linear combination of the two weights and presented a simple algorithm for finding a good linear combination of the two weights. He also proved near optimality properties of the two paths found. These heuristics can find a good solution quickly, but do not provide any *performance guarantee*.

Warburton in [23] first developed a *fully polynomial time approximation scheme* (FPTAS) [5] for the DCLC problem on an acyclic graph. Hassin in [9] presented two improved FPTASs, one with a time complexity of $O(mn(\frac{n}{\epsilon})\log(\frac{n}{\epsilon}))$, where ϵ is the approximation parameter, and the other with a time complexity of $O(\log\log\mathcal{B}(mn/\epsilon + \log\log\mathcal{B}))$ where $\mathcal{B}$ is an upper bound on the optimal solution value which is no more than $n-1$ times the maximum edge-cost. Hassin's algorithm, which has a straightforward extension to general graphs, finds a delay constrained path whose cost is within a factor of $(1+\epsilon)$ of that of the delay constrained least cost path. Lorenz and Raz in [15] presented a faster FPTAS with a time complexity of $O(mn(\log\log n + 1/\epsilon))$. In [7], Goel *et al.* presented an approximation algorithm for the single source all destinations delay sensitive least cost path problem of time complexity $O((m + n\log n)\mathcal{H}/\epsilon)$, where $\mathcal{H}$ is the hop count of the computed path. Note that the path computed by this algorithm does not necessarily satisfy the delay constraint: its delay is at most $(1+\epsilon)$ times the delay constraint and its cost is at most that of the delay constrained least cost path. In [6], Ergun *et al.* presented an FPTAS for the case of acyclic graphs with a time complexity of $O(m(\frac{n}{\epsilon}))$. Orda and Sprintson [19] presented a precomputation scheme for QoS routing with two additive parameters. Guerin and Orda [8] presented efficient approximation algorithms for QoS routing with inaccurate information. More recently [20], Orda and Sprintson presented efficient approximation algorithms for computing a pair of disjoint QoS paths. Applications of QoS in multiservice IP

Guoliang Xue, Arunabha Sen, Weiyi Zhang and Jian Tang are all with the Department of Computer Science and Engineering at Arizona State University, Tempe, AZ 85287-8809. Email: {xue, asen, weiyi.zhang, jian.tang}@asu.edu. Krishnaiya Thulasiraman is with the School of Computer Science at the University of Oklahoma, Norman, OK 73019. The research of Xue, Zhang and Tang was supported in part by ARO grant W911NF-04-1-0385 and NSF grants CCF-0431167 and ANI-0312635. The research of Thulasiraman was supported in part by NSF grant ANI-0312435.

networks and their practical significance may be found in [17].

The MCP problem with three or more constraints has also been studied. In [14], Korkmaz and Krunz proposed a *randomized heuristic* for the MCP problem. Using simulations they showed that their heuristic provides better performance than other algorithms with comparable computational complexity. In [28], Yuan presented a *limited granularity heuristic* and a limited path heuristic. Xue *et al.* [27] presented an FPTAS whose running time depends on the size of the input.

In this paper, we concentrate on the general case with $K \geq 2$ being any fixed integer and make the following contributions. (1) We present a simple $O(Km + n\log n)$ time K-approximation algorithm which can be easily used in hop-by-hop routing protocols. (2) We present an FPTAS for one optimization version of the MCP problem with a time complexity of $O(m(\frac{n}{\epsilon})^{K-1})$. (3) We present an FPTAS for another optimization version of the MCP problem with a time complexity of $O(n\log n + m(\frac{\mathcal{H}}{\epsilon})^{K-1})$ when there exists an $\mathcal{H}$-hop path satisfying all QoS constraints. When reduced to the special case of $K = 2$, the time complexities of our algorithms compare favorably with existing algorithms. Table I summarizes complexities of related algorithms for various versions of the MCP problems.

The rest of this paper is organized as follows. In Section II, we define the problems and some notations. In Section III, we present our simple K-approximation algorithm. In Sections IV and V, we present our two FPTASs. In Section VI, we present computational experiences. We conclude this paper in Section VII.

II. Definition of Problems and Notations

Throughout this paper, K denotes an *integer constant* which is greater than or equal to 2. All other constants, functions and variables are assumed to have *real values* unless specified otherwise.

We model a computer network by an edge weighted undirected graph $G = (V, E, \omega_1, \ldots, \omega_K)$, where V is the set of n *vertices*, E is the set of m *edges* each with K weights, and $\omega_k(e) \geq 0$ is the k^{th} *weight* of edge e, $\forall\, e \in E,\ \forall\, 1 \leq k \leq K$. Let p be a path in G. The k^{th} *weight* of p, denoted by $\omega_k(p)$, is the sum of the k^{th} weights over the edges on p. We assume that G is a connected graph. The *decision version* of the MCP problem (DMCP) is defined in the following.

Definition 2.1 ($\mathsf{DMCP}(G, s, t, K, W)$*):* INSTANCE: an undirected graph $G = (V, E)$, with K ***nonnegative real-valued*** edge weights $\omega_k(e), 1 \leq k \leq K$, associated with each edge $e \in E$; a positive constant W; and a source-destination node pair (s, t). QUESTION: Is there an s–t path p such that $\omega_k(p) \leq W,\ \forall\, 1 \leq k \leq K$?

A path p satisfying all K QoS constraints is called a *feasible path* of $\mathsf{DMCP}(G, s, t, K, W)$. We say that $\mathsf{DMCP}(G, s, t, K, W)$ is *feasible* if it has a feasible path, and *infeasible* otherwise. Note that we could formulate the DMCP problem in a seemingly more general form by replacing W with K independent positive constants $W_1, \ldots, W_K$ and replacing the K constraints $\omega_k(p) \leq W,\ \forall\, 1 \leq k \leq K$ with the following K constraints: $\omega_k(p) \leq W_k,\ \forall\, 1 \leq k \leq K$. However, the two forms are equivalent because we can *scale the k^{th} weight* (on edges, and thereby on paths) from $\omega_k(e)$ to $\omega'_k(e) = \omega_k(e) \times \frac{W}{W_k}$ so that for any path p, $\omega_k(p) \leq W_k,\ \forall\, 1 \leq k \leq K$ if and only if $\omega'_k(p) \leq W,\ \forall\, 1 \leq k \leq K$. Therefore we choose to use the simpler form in this paper. In the following, we define two optimization versions of this NP-hard problem.

Definition 2.2 ($\mathsf{SMCP}(G, s, t, K, W)$*):* INSTANCE: an undirected graph $G = (V, E)$, with K ***nonnegative real-valued*** edge weights $\omega_k(e), 1 \leq k \leq K$, associated with each edge $e \in E$; a positive constant W; and a source-destination node pair (s, t). PROBLEM: find an s–t path p^{opt} such that $\omega_k(p^{opt}) \leq \zeta^{opt} \cdot W,\ \forall\, 1 \leq k \leq K$, where ζ^{opt} is the smallest real number $\zeta \geq 0$ such that there exists an s–t path p satisfying $\omega_k(p) \leq \zeta \cdot W,\ \forall\, 1 \leq k \leq K$.

In the definition of SMCP, we are treating all K constraints equally, where a single parameter ζ^{opt} is applied to all K constraints. This is slightly different from traditional optimization versions of the DCLC problem, where we strictly enforce the delay constraint while approximating the minimum cost. When the number of constraints K is greater than 2, we have to approximate at least $K - 1$ constraints, because *finding a path satisfying two or more additive constraints is itself an NP-hard problem.* This motives us to approximate all K constraints simultaneously.

We call ζ^{opt} the *optimal value* of $\mathsf{SMCP}(G, s, t, K, W)$ and call p^{opt} an *optimal path* or an *optimal solution* of $\mathsf{SMCP}(G, s, t, K, W)$. Note that $\zeta^{opt} \leq 1$ *if and only if* $\mathsf{DMCP}(G, s, t, K, W)$ *is feasible*. Since ζ^{opt} is allowed to be smaller than 1, our optimization problem SMCP also introduces a metric to compare two feasible solutions to DMCP–the one with the smaller corresponding ζ value is regarded as a better solution. *When* $\zeta^{opt} \leq 1$*, any optimal solution of* $\mathsf{SMCP}(G, s, t, K, W)$ *is a feasible path for* $\mathsf{DMCP}(G, s, t, K, W)$*, but the reverse is not true.* Note that when a feasible path for $\mathsf{DMCP}(G, s, t, K, W)$ is not an optimal solution of $\mathsf{SMCP}(G, s, t, K, W)$, we must have $\zeta^{opt} < 1$. Therefore we define another optimization version of the MCP problem, named FMCP.

Definition 2.3 ($\mathsf{FMCP}(G, s, t, K, W)$*):* INSTANCE: an undirected graph $G = (V, E)$, with K ***nonnegative real-valued*** edge weights $\omega_k(e), 1 \leq k \leq K$, associated with each edge $e \in E$; a positive constant W; and a source-destination node pair (s, t). PROBLEM: find an s–t path q^{opt} such that $\omega_k(q^{opt}) \leq \xi^{opt} \cdot W,\ \forall\, 1 \leq k \leq K$, where ξ^{opt} is the smallest real number $\xi \geq 1$ such that there exists an s–t path q satisfying $\omega_k(q) \leq \xi \cdot W,\ \forall\, 1 \leq k \leq K$.

We call ξ^{opt} the *optimal value* of $\mathsf{FMCP}(G, s, t, K, W)$ and call q^{opt} an *optimal path* or an *optimal solution* of $\mathsf{FMCP}(G, s, t, K, W)$. Note that $\xi^{opt} \geq 1$. Also note that $\xi^{opt} = 1$ if and only if $\mathsf{DMCP}(G, s, t, K, W)$ has a feasible path. When $\xi^{opt} = 1$, any optimal solution of $\mathsf{FMCP}(G, s, t, K, W)$ is also a feasible path for $\mathsf{DMCP}(G, s, t, K, W)$. Also, any feasible path for $\mathsf{DMCP}(G, s, t, K, W)$ is guaranteed to be an optimal path for $\mathsf{FMCP}(G, s, t, K, W)$.

We note that every optimal solution to $\mathsf{SMCP}(G, s, t, K, W)$ is also an optimal solution

TABLE I
SUMMARY OF COMPUTATIONAL COMPLEXITIES OF ALGORITHMS FOR THE MCP PROBLEM

paper	K	enforcing	approximating	guarantee	time complexity
Lorenz and Raz [15]	2	delay	cost	$1+\epsilon$	$O(mn(\log\log n + 1/\epsilon))$
Goel *et al.* [7]	2	cost	delay	$1+\epsilon$	$O((m+n\log n)\mathcal{H}/\epsilon)$
Korkmaz and Krunz [14]	≥ 2	K constraints	none	heuristic	$O(Km + Kn\log n)$
Yuan [28]	≥ 2	K constraints	none	heuristic	$O(mn(n/\epsilon)^{K-1})$
This paper	≥ 2	none	K constraints	K	$O(Km + n\log n)$
This paper	≥ 2	none	K constraints	$1+\epsilon$	$O(m(n/\epsilon)^{K-1})$
This paper	≥ 2	none	K constraints	$1+\epsilon$	$O(n\log n + m(\mathcal{H}/\epsilon)^{K-1})$

to FMCP(G,s,t,K,W), but not vice versa. When DMCP(G,s,t,K,W) is infeasible, every optimal solution to FMCP(G,s,t,K,W) is also an optimal solution to SMCP(G,s,t,K,W).

Let p be an s–t path in G and $\beta \geq 1$ be a constant. If $\omega_k(p) \leq \beta \cdot \zeta^{opt} \cdot W,\ \forall\ 1 \leq k \leq K$ ($\omega_k(p) \leq \beta \cdot \xi^{opt} \cdot W,\ \forall\ 1 \leq k \leq K$, respectively), then p is called a *β-approximation* to SMCP(G,s,t,K,W) (to FMCP(G,s,t,K,W), respectively). If $\mathcal{A}$ is an algorithm that guarantees a β-approximation, then $\mathcal{A}$ is called a *β-approximation algorithm.* $\mathcal{A}_\epsilon$ is called a *fully polynomial time approximation scheme* (FPTAS), if for any fixed $\epsilon > 0$, $\mathcal{A}_\epsilon$ is a $(1+\epsilon)$-approximation algorithm with running time bounded by a polynomial in the *input size* of the instance, and in $\frac{1}{\epsilon}$.

Our FPTASs for SMCP(G,s,t,K,W) and FMCP(G,s,t,K,W) need to solve instances of the following *restricted version* of DMCP (where W is denoted by τ in the restricted version, and the edge weights take *positive integer values*) repeatedly with τ is bounded by a polynomial in n.

*Definition 2.4 (*RMCP(G,s,t,K,τ)*):* INSTANCE: an undirected graph $G=(V,E)$, with K ***positive integer-valued*** edge weights $\omega_k(e), 1 \leq k \leq K$, associated with each edge $e \in E$; a positive integer constant τ; and a source-destination node pair (s,t). QUESTION: Is there an s–t path p such that $\omega_k(p) \leq \tau,\ \forall\ 1 \leq k \leq K$?

III. A SIMPLE K-APPROXIMATION ALGORITHM FOR SMCP

A very simple K-approximation algorithm, named K-Approx, is presented in Algorithm 1. The algorithm computes an auxiliary edge weight $\omega_M(e)$ as the maximum of all K edge weights $\omega_1(e),\ldots,\omega_K(e)$ divided by W. It then computes a shortest s–t path p_M using this auxiliary edge weight (instead of using K edge weight functions). The path p_M is guaranteed to be a K-approximation of SMCP(G,s,t,K,W). Note that *the auxiliary edge weights can be computed locally at each node, and the shortest path can be computed using either Dijkstra's algorithm or Bellman-Ford's algorithm.* Therefore our K-approximation algorithm can be implemented as either a centralized or a distributed algorithm, and can be used by existing routing protocols such as RIP and OSPF [10].

```
Algorithm 1 K-Approx$(G,s,t,K,W)$
Step_1 For each edge $e$ of $G$, compute an auxiliary edge
       weight $\omega_M(e) = \max_{1 \leq k \leq K} \frac{\omega_k(e)}{W}$.
Step_2 Compute a shortest $s$–$t$ path $p_M$ in $G$, where the dis-
       tance is measured using the auxiliary edge weighting
       function $\omega_M$. Output $p_M$.
```

Theorem 3.1: The path p_M found by K-Approx is a K-approximation to SMCP(G,s,t,K,W), i.e., $\omega_k(p_M) \leq K \cdot \zeta^{opt} W,\ \forall\ 1 \leq k \leq K$, where ζ^{opt} is the optimal value of SMCP(G,s,t,K,W). Moreover,

1) if Dijkstra's algorithm is used, the time complexity of K-Approx is $O(Km + n\log n)$;
2) if centralized Bellman-Ford algorithm is used, the time complexity of K-Approx is $O(Km + mn)$;
3) if distributed Bellman-Ford algorithm is used, the time complexity (measured in terms of the number of rounds of executions needed) of K-Approx is $O(n)$.

PROOF. With a centralized algorithm the auxiliary weights can be computed in $O(Km)$ time as there are m edges. The rest of the time analysis comes from well known results [5]. For distributed computation, each node can compute the auxiliary weights of the adjacent edges in one round of execution. Bellman-Ford distributed shortest path algorithm will terminate in at most n rounds. So the overall time complexity of the distributed algorithm is $O(n)$.

Recall that ζ^{opt} is the optimal value of SMCP(G,s,t,K,W). Therefore there exists an s–t path p^{opt} such that $\omega_k(p^{opt}) \leq \zeta^{opt} W,\ \forall\ 1 \leq k \leq K$. This implies

$$\sum_{e \in p^{opt}} \omega_k(e) \leq \zeta^{opt} W,\ \forall\ 1 \leq k \leq K. \tag{1}$$

We can rewrite (1) as

$$\sum_{e \in p^{opt}} \frac{\omega_k(e)}{W} \leq \zeta^{opt},\ \forall\ 1 \leq k \leq K. \tag{2}$$

Summing (2) over all K possible values of k, we have

$$\sum_{e \in p^{opt}} \sum_{k=1}^{K} \frac{\omega_k(e)}{W} \leq K \cdot \zeta^{opt}. \tag{3}$$

Since for every edge $e \in E$ we have $\omega_M(e) = \max_{1 \leq k \leq K} \frac{\omega_k(e)}{W} \leq \sum_{k=1}^{K} \frac{\omega_k(e)}{W}$, (3) implies

$$\sum_{e \in p^{opt}} \omega_M(e) \leq K \cdot \zeta^{opt}. \tag{4}$$

Note that the left hand side of (4) is $\omega_M(p^{opt})$. Since p_M is a shortest s–t path in G with respect to ω_M, we must have $\omega_M(p_M) \le \omega_M(p^{opt})$. Therefore

$$\omega_M(p_M) \le \omega_M(p^{opt}) \le K \cdot \zeta^{opt}. \tag{5}$$

Since for every edge $e \in E$ we have $\omega_M(e) = \max_{1\le k\le K} \frac{\omega_k(e)}{W} \ge \frac{\omega_k(e)}{W}$, $\forall\ 1 \le k \le K$, (5) implies

$$\sum_{e\in p_M} \frac{\omega_k(e)}{W} \le K \cdot \zeta^{opt},\ \forall\ 1 \le k \le K. \tag{6}$$

We can rewrite (6) as $\omega_k(p_M) \le K \cdot \zeta^{opt} W,\ \forall\ 1 \le k \le K$. This proves that p_M is a K-approximation to SMCP(G, s, t, K, W). □

IV. An FPTAS for SMCP

In this section, we will present an FPTAS for SMCP(G, s, t, K, W) for any constant $K \ge 2$.

A. A Pseudo-Polynomial Time Algorithm for RMCP

We first present an $O(m\tau^{K-1})$ time algorithm for RMCP(G, s, t, K, τ). The algorithm is named PseudoRMCP and is listed as Algorithm 2. When the answer is YES, our algorithm also computes a feasible s–t path p^G which minimizes $\max_{1\le k\le K} \omega_k(p)$ among all s–t paths in G.

Recall that all edge weights in RMCP are positive integers. Therefore along any path, each hop increases each of the K path lengths by at least 1. Our algorithm uses a transformation from an undirected graph G to a *directed acyclic graph* $G^{K,\tau}$. Each vertex $v \in G$ is associated with $(1+\tau)^{K-1}$ vertices in $G^{K,\tau}$, of the form $(v, \mathcal{C}_2, \dots, \mathcal{C}_K)$, where the integers $\mathcal{C}_2, \dots, \mathcal{C}_K \in [0, \tau]$ are used to record the kth path length for $k = 2, \dots, K$. Therefore for each undirected edge (u, v) in G, we have directed edges in $G^{K,\tau}$ from $(u, \mathcal{C}_2, \dots, \mathcal{C}_K)$ to $(v, \mathcal{D}_2, \dots, \mathcal{D}_K)$ such that $\mathcal{D}_k = \mathcal{C}_k + \omega_k(u, v), k = 2, \dots, K$; as well as directed edges from $(v, \mathcal{C}_2, \dots, \mathcal{C}_K)$ to $(u, \mathcal{D}_2, \dots, \mathcal{D}_K)$ such that $\mathcal{D}_k = \mathcal{C}_k + \omega_k(u, v), k = 2, \dots, K$. All such edges have *length* $\omega_1(u, v)$. Therefore a feasible solution to RMCP(G, s, t, K, τ) corresponds to a path from $(s, 0, \dots, 0)$ to $(t, \tau, \dots, \tau)$ with length no more than τ.

Algorithm 2 PseudoRMCP(G, s, t, K, τ)

```
Step_1  Construct a directed graph $G^{K,\tau}$ with node set $V^{K,\tau} = V \times \{0, 1, \dots, \tau\}^{K-1}$ and edge set $E^{K,\tau}$.
        Let $(u, v)$ be an undirected edge in $E$. $E^{K,\tau}$ contains directed edges from vertex $(u, \mathcal{C}_2, \dots, \mathcal{C}_K)$ to vertex
        $(v, \mathcal{D}_2, \dots, \mathcal{D}_K)$ such that $\mathcal{D}_k = \mathcal{C}_k + \omega_k(u, v), k = 2, \dots, K$; as well as directed edges from vertex
        $(v, \mathcal{C}_2, \dots, \mathcal{C}_K)$ to vertex $(u, \mathcal{D}_2, \dots, \mathcal{D}_K)$ such that $\mathcal{D}_k = \mathcal{C}_k + \omega_k(u, v), k = 2, \dots, K$; The length of all
        such edges is $\omega_1(u, v)$. In addition, $E^{K,\tau}$ also contains zero-length edges from vertex $(t, \mathcal{C}_2, \dots, \mathcal{C}_K)$ to
        vertex $(t, \mathcal{D}, \dots, \mathcal{D})$ where $\mathcal{D} = \max\{\mathcal{C}_j | 2 \le j \le K\}$ if not all of the $K-1$ values $\mathcal{C}_j$ are equal,
        $\mathcal{D} = \mathcal{C} + 1$ if $\mathcal{C}_2 = \cdots = \mathcal{C}_K = \mathcal{C} < \tau$.
Step_2  Compute shortest paths $p^{K,\mathcal{D}}$ from vertex $(s, 0, \dots, 0)$ to vertices $(t, \mathcal{D}, \dots, \mathcal{D})$ in $G^{K,\tau}$,
        $\mathcal{D} = 1, 2, \dots, \tau$. If the length of path $p^{K,\tau}$ is greater than $\tau$, then RMCP$(G, s, t, K, \tau)$ does not have a
        feasible solution. Output NO.
Step_3  Find the smallest integer $X \le \tau$ such that the shortest path $p^{K,X}$ has length no more than $X$. Output YES,
        together with the path $p^G$ corresponding to $p^{K,X}$, obtained by ignoring the last $K-1$ components
        within each node along the path $p^{K,X}$.
```

Theorem 4.1: Algorithm 2 computes a feasible path p^G for RMCP(G, s, t, K, τ) if it has a feasible solution. The worst case time complexity of the algorithm is $O(m\tau^{K-1})$. In addition, when the answer is YES, the path p^G is a feasible solution to RMCP$(G, s, t, K, \mathcal{D})$, where $\mathcal{D}$ is the smallest integer less than or equal to τ such that RMCP$(G, s, t, K, \mathcal{D})$ is feasible.

Proof. Note that $G^{K,\tau}$ has $n(\tau+1)^{K-1}$ vertices and $O(2m\tau^{K-1} + n\tau^{K-1}) = O(m\tau^{K-1})$ edges (recall that G is connected and K is a constant). It follows from the construction of $G^{K,\tau}$ that $G^{K,\tau}$ has a directed path $p^{K,\tau}$ from $(u, \mathcal{C}_2, \dots, \mathcal{C}_K)$ to $(v, \mathcal{D}_2, \dots, \mathcal{D}_K)$ with length ℓ if and only if the corresponding path p^G in G (obtained by keeping only the first component in each vertex on $p^{K,\tau}$) satisfies $\omega_1(p) = \ell$ and $\omega_k(p) \le \mathcal{D}_k - \mathcal{C}_k$ for $k = 2, \dots, K$. Therefore RMCP(G, s, t, K, τ) has a feasible solution if and only if there is a directed path from vertex $(s, 0, \dots, 0)$ to vertex $(t, \tau, \tau, \dots, \tau)$ in $G^{K,\tau}$ with length at most τ. This proves the correctness of Algorithm 2.

We note that if a directed path $p^{K,\mathcal{D}}$ (from vertex $(s, 0, \dots, 0)$ to vertex $(t, \mathcal{D}, \dots, \mathcal{D})$ in $G^{K,\tau}$) computed in Step_2 *has length at most* $\mathcal{D}$ for some integer $\mathcal{D} \in \{1, 2, \dots, \tau\}$, then the path $p^{K,\mathcal{D}}$ is also a feasible solution to RMCP$(G, s, t, K, \mathcal{D})$. The reverse is also true. This proves that the path p^G returned by Algorithm 2 is a feasible solution to RMCP$(G, s, t, K, \mathcal{D})$, where $\mathcal{D}$ is the smallest integer less than or equal to τ such that RMCP$(G, s, t, K, \mathcal{D})$ is feasible, provided that RMCP(G, s, t, K, τ) is feasible.

Since each $\omega_k(e)$ is a positive integer, for $k = 2, \dots, K$ and $e \in E$, the existence of a directed edge from $(u, \mathcal{C}_2, \dots, \mathcal{C}_K)$ to $(v, \mathcal{D}_2, \dots, \mathcal{D}_K)$ in $G^{K,\tau}$ implies that $\mathcal{C}_k < \mathcal{D}_k,\ \forall\ 2 \le k \le K$. Therefore the graph $G^{K,\tau}$ is acyclic. As a result, it takes $O(m\tau^{K-1} + n\tau^{K-1}) = O(m\tau^{K-1})$ time to compute the shortest paths from $(s, 0, \dots, 0)$ to all other vertices in $G^{K,\tau}$, since the single source shortest paths in an acyclic graph can be computed in linear time (see [5], page 592). □

B. Polynomial Time Approximate Testing

Our FPTAS uses the following polynomial time approximate testing procedure [15]. For a given positive real number θ, we construct an auxiliary graph $G^\theta = (V, E, \omega_1^\theta, \dots, \omega_K^\theta)$ which is the same as G except that the edge weighting function ω_k is changed to ω_k^θ such that $\omega_k^\theta(e) = \lfloor \omega_k(e) \cdot \theta \rfloor + 1$ for every $e \in E$, which is called the k^{th} *θ-scaled weighting function*. For given real numbers $\mathbf{C} > 0$ and $\epsilon > 0$ (we assume $\epsilon < K$), we define TEST$(\mathbf{C}, \epsilon)$ = YES if RMCP$(G^\theta, s, t, K, \lfloor \frac{n-1}{\epsilon} \rfloor + n - 1)$ is feasible (where

$\theta = \frac{n-1}{\mathbf{C}\cdot W \cdot \epsilon}$) and define $\mathsf{TEST}(\mathbf{C},\epsilon) = \mathsf{NO}$ otherwise. Using standard techniques of scaling and rounding [9], [11], [15], [21], one can prove that $\mathsf{TEST}(\mathbf{C},\epsilon) = \mathsf{NO}$ implies $\zeta^{opt} > \mathbf{C}$ and that $\mathsf{TEST}(\mathbf{C},\epsilon) = \mathsf{YES}$ implies $\zeta^{opt} < \mathbf{C}\cdot(1+\epsilon)$. Recall that ζ^{opt} is the optimal value of $\mathsf{SMCP}(G,s,t,K,W)$. This is formally stated in the following theorem.

Theorem 4.2: Let ζ^{opt} be the optimal value of $\mathsf{SMCP}(G,s,t,K,W)$. Let $\mathbf{C}$ and ϵ be two fixed positive numbers. Then

- $\mathsf{TEST}(\mathbf{C},\epsilon) = \mathsf{YES}$ implies $\zeta^{opt} < \mathbf{C}\cdot(1+\epsilon)$;
- $\mathsf{TEST}(\mathbf{C},\epsilon) = \mathsf{NO}$ implies $\zeta^{opt} > \mathbf{C}$.

Furthermore, the time complexity of $\mathsf{TEST}(\mathbf{C},\epsilon)$ is $O(m(\frac{n}{\epsilon})^{K-1})$. □

C. *The* FPTAS *for* SMCP

We may apply Algorithm 1 to compute an s–t path p_M. According to Theorem 3.1, $\omega_M(p_M)/K$ is a lower bound for ζ^{opt} and $\omega_M(p_M)$ is an upper bound for ζ^{opt}. If $\omega_M(p_M) = 0$, we know that p_M is also an optimal solution to $\mathsf{SMCP}(G,s,t,K,W)$. If $\omega_M(p_M) > 0$, we can use the approximate testing procedure to generate a sequence of lower bound-upper bound pairs so that the ratio of the upper bound over the corresponding lower bound goes sufficiently close to 1, and then solve an instance of RMCP to obtain an $(1+\epsilon)$-approximation to SMCP. Following the ideas of Lorenz and Raz [15], we say that $\mathsf{UB}^{[0]}$ ($\mathsf{UB}^{[i]}$, respectively) is an *approximate upper bound* for ζ^{opt} if $2\cdot\mathsf{UB}^{[0]} \ge \zeta^{opt}$ ($2\cdot\mathsf{UB}^{[i]} \ge \zeta^{opt}$, respectively). We set the initial lower bound of ζ^{opt} to $\mathsf{LB}^{[0]} \equiv \omega_M(p_M)/K$ and initial *approximate upper bound* of ζ^{opt} to $\mathsf{UB}^{[0]} \equiv \omega_M(p_M)/2$. Our FPTAS is presented in Algorithm 3.

Algorithm 3 $\mathsf{FPTAS\text{-}SMCP}(G,s,t,K,W)$

```
Step_1 Apply Algorithm 1 to compute a $K$-approximation
       $p_M$ to $\mathsf{SMCP}(G,s,t,K,W)$.
       if $\omega_M(p_M) = 0$, output $p_M$ and stop. $p_M$ is an
       optimal solution to $\mathsf{SMCP}(G,s,t,K,W)$.
Step_2 Set $\mathsf{LB} := \mathsf{LB}^{[0]} := \omega_M(p_M)/K$ and set $\mathsf{UB} :=$
       $\mathsf{UB}^{[0]} := \omega_M(p_M)/2$;
Step_3 if $\mathsf{UB} \le 2\cdot\mathsf{LB}$ then
         goto Step_4;
       else
         let $\mathbf{C} := \sqrt{\mathsf{UB}\cdot\mathsf{LB}}$;
         if $\mathsf{TEST}(\mathbf{C},1) = \mathsf{NO}$, set $\mathsf{LB} := \mathbf{C}$;
         if $\mathsf{TEST}(\mathbf{C},1) = \mathsf{YES}$, set $\mathsf{UB} := \mathbf{C}$;
         goto Step_3;
       endif
Step_4 Set $\theta := \frac{n-1}{\mathsf{LB}\cdot W\cdot\epsilon}$. Apply Algorithm 2 to
       $\mathsf{RMCP}(G^\theta,s,t,K,\lfloor\frac{2\mathsf{UB}(n-1)}{\mathsf{LB}\epsilon}\rfloor + n - 1)$ and output
       the corresponding feasible path $p^G$.
```

Theorem 4.3: Algorithm 3 finds a $(1+\epsilon)$-approximation to $\mathsf{SMCP}(G,s,t,K,W)$ in $O(m(\frac{n}{\epsilon})^{K-1})$ time.

PROOF. Let $\{\mathsf{LB}^{[i]}\}$ and $\{\mathsf{UB}^{[i]}\}$ denote the sequences of lower bounds and approximate upper bounds generated by Step_2 and Step_3 of the algorithm. We know that

$$\mathsf{LB}^{[i]} \le \zeta^{opt} \le 2\cdot\mathsf{UB}^{[i]} \tag{7}$$

is true for $i = 0$. Assume that (7) is true for $i = l \ge 0$. If $\mathsf{TEST}(\sqrt{\mathsf{LB}^{[l]}\cdot\mathsf{UB}^{[l]}},1) = \mathsf{NO}$, we set $\mathsf{LB}^{[l+1]} := \sqrt{\mathsf{LB}^{[l]}\cdot\mathsf{UB}^{[l]}}$ and $\mathsf{UB}^{[l+1]} := \mathsf{UB}^{[l]}$. If $\mathsf{TEST}(\sqrt{\mathsf{LB}^{[l]}\cdot\mathsf{UB}^{[l]}},1) = \mathsf{YES}$, we set $\mathsf{LB}^{[l+1]} := \mathsf{LB}^{[l]}$ and $\mathsf{UB}^{[l+1]} := \sqrt{\mathsf{LB}^{[l]}\cdot\mathsf{UB}^{[l]}}$. It follows from Theorem 4.2 that (7) is also true for $i = l+1$. Also from the definition of the sequences $\{\mathsf{LB}^{[i]}\}$ and $\{\mathsf{UB}^{[i]}\}$, we have

$$\log\mathsf{UB}^{[i+1]} - \log\mathsf{LB}^{[i+1]} = \frac{\log\mathsf{UB}^{[i]} - \log\mathsf{LB}^{[i]}}{2},\ i \ge 0. \tag{8}$$

Therefore Step_3 of the algorithm is executed no more than $\lceil\log(\log\mathsf{UB}^{[0]} - \log\mathsf{LB}^{[0]})\rceil$ times. However, $\log\mathsf{UB}^{[0]} - \log\mathsf{LB}^{[0]} \le \log K$ according to Theorem 3.1. As a result, the worst case running time required by Step_2 and Step_3 of the algorithm is bounded by $O(mn^{K-1} \times \log\log K)$.

In the rest of this proof, we will use θ to denote $\frac{n-1}{\mathsf{LB}\cdot W\cdot\epsilon}$ (as in Step_4) to simplify notations within the proof. Let p^{opt} be an optimal solution to $\mathsf{SMCP}(G,s,t,K,W)$, i.e., p^{opt} is an s–t path such that $\omega_k(p^{opt}) \le \zeta^{opt}\cdot W$ for $k = 1,2,\ldots,K$. Since $\omega_k^\theta(e) = \lfloor\omega_k(e)\cdot\frac{n-1}{\mathsf{LB}\cdot W\cdot\epsilon}\rfloor + 1 \le \omega_k(e)\cdot\frac{n-1}{\mathsf{LB}\cdot W\cdot\epsilon} + 1$ for every edge $e \in E$, we have (noting that p^{opt} has at most $n-1$ edges)

$$\omega_k^\theta(p^{opt}) \le \omega_k(p^{opt})\cdot\frac{n-1}{\mathsf{LB}\cdot W\cdot\epsilon} + n - 1 \le \zeta^{opt}\cdot\frac{n-1}{\mathsf{LB}\cdot\epsilon} + n - 1$$
$$\le \frac{2\mathsf{UB}(n-1)}{\mathsf{LB}\epsilon} + n - 1,\ \forall\ 1 \le k \le K. \tag{9}$$

Since ω_k^θ always have integer values, (9) implies

$$\omega_k^\theta(p^{opt}) \le \lfloor\frac{2\mathsf{UB}(n-1)}{\mathsf{LB}\epsilon}\rfloor + n - 1,\ \forall\ 1 \le k \le K. \tag{10}$$

This implies that p^{opt} is a feasible solution to $\mathsf{RMCP}(G^\theta,s,t,K,\lfloor\frac{2\mathsf{UB}(n-1)}{\mathsf{LB}\epsilon}\rfloor + n - 1)$. Therefore Step_4 of the algorithm is guaranteed to find a feasible path. Note that (9) also implies

$$\max_{1\le k\le K}\omega_k^\theta(p^{opt}) \le \zeta^{opt}\cdot\frac{n-1}{\mathsf{LB}\cdot\epsilon} + n - 1. \tag{11}$$

Let p^G be the s–t path found in Step_4 of the algorithm. It follows from Theorem 4.1 that p^G is a feasible solution to $\mathsf{RMCP}(G^\theta,s,t,K,\mathcal{D})$, where $\mathcal{D}$ is the smallest integer less than or equal to $\lfloor\frac{2\mathsf{UB}(n-1)}{\mathsf{LB}\epsilon}\rfloor + n - 1$ such that $\mathsf{RMCP}(G^\theta,s,t,K,\mathcal{D})$ is feasible. Since p^G is *optimal* while p^{opt} is only *feasible*, the maximum path weight of p^G cannot exceed the maximum path weight of p^{opt}:

$$\max_{1\le k\le K}\omega_k^\theta(p^G) \le \max_{1\le k\le K}\omega_k^\theta(p^{opt}). \tag{12}$$

Combining (12) with (11), we obtain

$$\max_{1\le k\le K}\omega_k^\theta(p^G) \le \zeta^{opt}\cdot\frac{n-1}{\mathsf{LB}\cdot\epsilon} + n - 1. \tag{13}$$

On the other hand, we also have

$$\omega_k^\theta(p^G) = \sum_{e\in p^G}\omega_k^\theta(e) \ge \sum_{e\in p^G}\frac{\omega_k(e)\cdot(n-1)}{\mathsf{LB}\cdot W\cdot\epsilon}$$

$$= \omega_k(p^G) \cdot \frac{n-1}{\mathsf{LB} \cdot W \cdot \epsilon}, \ \forall\, 1 \le k \le K. \tag{14}$$

Combining (13) and (14), we have

$$\omega_k(p^G) \cdot \frac{n-1}{\mathsf{LB} \cdot W \cdot \epsilon} \le \zeta^{opt} \cdot \frac{n-1}{\mathsf{LB} \cdot \epsilon} + n - 1, \ \forall\, 1 \le k \le K. \tag{15}$$

Some algebraic manipulations on (15) yield the following.

$$\omega_k(p^G) \le \zeta^{opt} \cdot W + \mathsf{LB} \cdot W \cdot \epsilon \le (1+\epsilon) \cdot \zeta^{opt} \cdot W, \ \forall\, 1 \le k \le K. \tag{16}$$

Therefore p^G is an $(1+\epsilon)$ approximation to SMCP(G, s, t, K, W). It follows from Theorem 4.1 that the worst case time complexity of Step_4 is $O(m(\frac{n}{\epsilon})^{K-1})$. Since $K \ge 2$ is a constant, the time complexity of Step_4 dominates the time complexity of all other steps. Therefore the overall time complexity of Algorithm 3 is $O(m(\frac{n}{\epsilon})^{K-1})$. □

The limited granularity heuristic algorithm of Yuan [28] is closest to our approximation scheme. Given a precision $\epsilon > 0$, Yuan's heuristic maintains a table of size $O((\frac{n}{\epsilon})^{K-1})$ at each node and has a time complexity of $O(mn(\frac{n}{\epsilon})^{K-1})$. We would like to point out the difference between Yuan's *heuristic* and our FPTAS-SMCP. Yuan's heuristic is designed for DMCP(G, s, t, K, W), the *decision version* of the problem. *If there is an s–t path p such that every path weight of p is no more than $(1-\epsilon)W$, Yuan's heuristic is guaranteed to find a feasible path. When the above condition is not true, the heuristic does not guarantee finding a feasible path, even in the case where a feasible path exists.* Note that for a given $\epsilon > 0$, checking the existence of an s–t path p with every path weight no more than $(1-\epsilon)W$ is itself an NP-hard problem. Therefore when Yuan's heuristic fails to find an s–t path, we do not know whether DMCP(G, s, t, K, W) is feasible or infeasible. This is a character common to all heuristics. In contrast, FPTAS-SMCP is designed for SMCP(G, s, t, K, W), an *optimization version* of the problem. In the case where DMCP(G, s, t, K, W) is feasible, FPTAS-SMCP always finds an s–t path p such that every path weight of p is no more than $(1+\epsilon)W$. Regardless of the feasibility of DMCP(G, s, t, K, W), FPTAS-SMCP always finds a $(1+\epsilon)$-approximation to the optimal solution of SMCP(G, s, t, K, W). In addition, if for the found path p^G we have $\max\limits_{1 \le k \le K} \dfrac{\omega_k(p^G)}{W} > 1 + \epsilon$, we can conclude that DMCP(G, s, t, K, W) is infeasible.

Note that Yuan's heuristic can also be implemented with the constraint W enlarged to $W' = W/(1-\epsilon)$. To distinguish from its original form, we call Yuan's heuristic implemented in this way *Yuan's reverse heuristic*. Note that the feasibility of DMCP(G, s, t, K, W) guarantees Yuan's reverse heuristic to find an s–t path p such that $\max\limits_{1 \le k \le K} \omega_k(p) \le \dfrac{W}{1-\epsilon}$. Note that for $\epsilon \in (0, 0.5]$, $\frac{1}{1-\epsilon} = 1 + \epsilon + \epsilon^2 + \cdots$, which is a number in the interval $(1 + \epsilon + \epsilon^2, 1 + 2\epsilon]$. So for $\epsilon \in (0, 0.5]$, the feasibility of DMCP(G, s, t, K, W) guarantees Yuan's reverse heuristic to find an s–t path p such that $\max\limits_{1 \le k \le K} \omega_k(p) \le (1 + 2\epsilon)W$. When it fails to find an s–t path, we can conclude that DMCP(G, s, t, K, W) is infeasible. So Yuan's reverse heuristic exhibits several advantages over the original Yuan's heuristic. We point out that compared with Yuan's reverse heuristic, FPTAS-SMCP has a lower time complexity and a better approximation performance.

Note that for $K = 2$, FPTAS-SMCP runs in time $O(\frac{mn}{\epsilon})$, which is faster than the FPTAS of Lorenz and Raz [15] with a complexity of $O(mn(\log\log n + 1/\epsilon))$. We point out that for SMCP$(G, s, t, 2, W)$ FPTAS-SMCP approximates both the cost and the delay while the FPTAS of [15] is designed to compute a path that minimizes cost under delay constraint.

V. An FPTAS for FMCP

Following the technique of [7], we present an FPTAS for FMCP(G, s, t, K, W) which guarantees finding a $(1+\epsilon)$-approximation to FMCP(G, s, t, K, W). When DMCP(G, s, t, K, W) is feasible, our FPTAS finds a $(1+\epsilon)$-approximation to FMCP(G, s, t, K, W) in $O(n \log n + m(\frac{\mathcal{H}}{\epsilon})^{K-1})$ time, where $\mathcal{H}$ is the minimum length (in hops) of any feasible path to DMCP(G, s, t, K, W). The worst-case time complexity of the algorithm (in all cases) is $O(m(\frac{n}{\epsilon})^{K-1})$, which is asymptotically the same as the time complexity of FPTAS-SMCP. This FPTAS is named FPTAS-FMCP and presented in Algorithm 4.

Algorithm 4 FPTAS-FMCP(G, s, t, K, W)

```
Step_1 Apply Algorithm 1 to compute a $K$-approximation
       $p_M$ to SMCP$(G, s, t, K, W)$.
       if ($\omega_M(p_M) \le 1 + \epsilon$) then
          output $p_M$ and stop. $p_M$ is a $(1+\epsilon)$-approximation
          to FMCP$(G, s, t, K, W)$.
       else
          set LB $:= \omega_M(p_M)/K$; set UB $:= \omega_M(p_M)/2$;
          set $H := 1$;
       endif
Step_2 Set $\theta := \frac{H}{\mathsf{LB} \cdot W \cdot \epsilon}$. Apply Algorithm 2 to
       RMCP$(G^\theta, s, t, K, \lfloor \frac{2\mathsf{UB} \cdot H}{\mathsf{LB} \cdot \epsilon} \rfloor + H)$.
       if (RMCP$(G^\theta, s, t, K, \lfloor \frac{2\mathsf{UB} \cdot H}{\mathsf{LB} \cdot \epsilon} \rfloor + H)$ is infeasible)
       then
          set $H := \min\{2H, n-1\}$; goto Step_2;
       else
          Let $p^G$ be the $s$–$t$ path returned by Algorithm 2.
          if $\omega_k(p^G) \le (1+\epsilon) \cdot W, \ \forall\, 1 \le k \le K$ then
             output path $p^G$ and stop.
             $p^G$ is a $(1+\epsilon)$-approximation to
             FMCP$(G, s, t, K, W)$.
          endif
       endif
Step_3 if $H < n - 1$ then
          set $H := \min\{2H, n-1\}$; goto Step_2;
       else
          output path $p^G$ and stop.
          DMCP$(G, s, t, K, W)$ is infeasible ($\xi^{opt} > 1$).
          $p^G$ is a $(1+\epsilon)$-approximation to
          SMCP$(G, s, t, K, W)$.
       endif
```

Theorem 5.1: FPTAS-FMCP finds a $(1+\epsilon)$-approximation to the $\mathsf{FMCP}(G,s,t,K,W)$ problem in $O(m(\frac{n}{\epsilon})^{K-1})$ time. In particular, when $\mathsf{DMCP}(G,s,t,K,W)$ is feasible, FPTAS-FMCP finds a $(1+\epsilon)$-approximation to the $\mathsf{FMCP}(G,s,t,K,W)$ problem in $O(n\log n + m(\frac{\mathcal{H}}{\epsilon})^{K-1})$ time, where $\mathcal{H}$ is the minimum length (in hops) among all feasible solutions to $\mathsf{DMCP}(G,s,t,K,W)$.

PROOF. Due to the condition checking in Step_1, we know that the path p_M is a $(1+\epsilon)$-approximation to $\mathsf{FMCP}(G,s,t,K,W)$ if the algorithm stops within Step_1. Similarly, we know that the path p^G is a $(1+\epsilon)$-approximation to $\mathsf{FMCP}(G,s,t,K,W)$ if the algorithm stops within Step_2. If Algorithm 4 stops within Step_3, we know (from the correctness proof of Algorithm 3) that the path p^G is a $(1+\epsilon)$-approximation to $\mathsf{SMCP}(G,s,t,K,W)$, since the constraint in the instance of RMCP is set to $\lfloor\frac{2\mathsf{UB}\cdot(n-1)}{\mathsf{LB}\cdot\epsilon}\rfloor + n - 1$ (which is equivalent to Step_4 of FPTAS-SMCP).

Next, we analyze the time complexity of Algorithm 4. Step_1 calls our K-approximation algorithm, which has a time complexity of $O(Km + n\log n)$. Suppose that Step_2 solves t instances of RMCP using Algorithm 2, with H taking the values $1 = H_1 < H_2 < \cdots < H_t$. Then $H_j = 2^{j-1}$ for $j = 1, 2, \ldots, t-1$ and $H_t = \min\{2^t, n-1\}$. Therefore the total time required for these t calls to Algorithm 2 is bounded by (recall that $K \geq 2$ is a constant)

$$O(m(\frac{H_1}{\epsilon})^{K-1}) + O(m(\frac{H_2}{\epsilon})^{K-1}) + \cdots O(m(\frac{H_t}{\epsilon})^{K-1})$$
$$= O(m(\frac{H_t}{\epsilon})^{K-1}). \quad (17)$$

In the worst-case, we have $H_t = n-1$. Therefore the time complexity of Algorithm 4 is bounded by $O(Km + n\log n + m(\frac{n}{\epsilon})^{K-1}) = O(m(\frac{n}{\epsilon})^{K-1})$ in the worst case.

Finally, we prove the claim corresponding to the case where $\mathsf{DMCP}(G,s,t,K,W)$ is feasible. If the algorithm stops within Step_1, the running time is $O(Km + n\log n)$, which is bounded by $O(n\log n + m(\frac{\mathcal{H}}{\epsilon})^{K-1})$. Therefore we will assume that the algorithm does not stop within Step_1. Let p^{opt} be a feasible solution to $\mathsf{DMCP}(G,s,t,K,W)$ which, among all feasible solutions to $\mathsf{DMCP}(G,s,t,K,W)$, has the minimum number of hops, denoted by $\mathcal{H} = |p^{opt}|$. We will prove that Algorithm 4 will stop within Step_2 with $H = H_t$ such that $H_{t-1} < \mathcal{H} \leq H_t$ (H_0 is assumed to be 0 for notational purpose, in case $H_1 = \mathcal{H} = 1$).

Since p^{opt} is a feasible solution to $\mathsf{DMCP}(G,s,t,K,W)$, we have

$$\omega_k(p^{opt}) \leq W, \ \forall\ 1 \leq k \leq K. \quad (18)$$

The feasibility of $\mathsf{DMCP}(G,s,t,K,W)$ also implies that $\zeta^{opt} \leq 1$. It follows from Theorem 3.1 that $\mathsf{LB} \leq \zeta^{opt} \leq 2\cdot\mathsf{UB}$. It follows from the description of Algorithm 4 that we must have $\omega_M(p_M) > (1+\epsilon)$ when the algorithm enters Step_2. In other words, we have $2\cdot\mathsf{UB} = \omega_M(p_M) > (1+\epsilon)$. Therefore the following inequality is true.

$$\mathsf{LB} \leq \zeta^{opt} \leq 1 < 1+\epsilon < 2\cdot\mathsf{UB}. \quad (19)$$

Assume that we enter Step_2 with $H \geq \mathcal{H}$. We will have $\theta = \frac{H}{\mathsf{LB}\cdot W\cdot\epsilon}$. Since p^{opt} is a feasible solution to $\mathsf{DMCP}(G,s,t,K,W)$ with hop-length equal to $\mathcal{H}$, we have

$$\omega_k^\theta(p^{opt}) = \sum_{e\in p^{opt}}(\lfloor\theta\cdot\omega_k(e)\rfloor + 1) < \sum_{e\in p^{opt}}(\theta\cdot\omega_k(e) + 1)$$
$$= \theta\cdot\omega_k(p^{opt}) + |p^{opt}| \leq \theta\cdot W + \mathcal{H}, \ \forall\ 1 \leq k \leq K. \quad (20)$$

It follows from (19) that $2\cdot\mathsf{UB} > 1$. Therefore (20) implies

$$\omega_k^\theta(p^{opt}) \leq \theta\cdot W + H = \frac{H}{\mathsf{LB}\cdot W\cdot\epsilon}\cdot W + H$$
$$\leq \frac{2\mathsf{UB}\cdot H}{\mathsf{LB}\cdot\epsilon} + H, \ \forall\ 1 \leq k \leq K. \quad (21)$$

Since $\omega_k^\theta(p^{opt})$ is an integer, (21) implies

$$\omega_k^\theta(p^{opt}) \leq \lfloor\frac{2\mathsf{UB}\cdot H}{\mathsf{LB}\cdot\epsilon}\rfloor + H, \ \forall\ 1 \leq k \leq K. \quad (22)$$

Therefore p^{opt} is a feasible solution to $\mathsf{RMCP}(G^\theta, s, t, K, \lfloor\frac{2\mathsf{UB}\cdot H}{\mathsf{LB}\cdot\epsilon}\rfloor + H)$.

It follows from (20) that

$$\max_{1\leq k\leq K}\omega_k^\theta(p^{opt}) \leq \theta\cdot W + H. \quad (23)$$

Let p^G be the s–t path returned by Algorithm 2 for $\mathsf{RMCP}(G^\theta, s, t, K, \lfloor\frac{2\mathsf{UB}\cdot H}{\mathsf{LB}\cdot\epsilon}\rfloor + H)$. Since p^G is *optimal* and p^{opt} has been proved to be *feasible*, we must have the following.

$$\max_{1\leq k\leq K}\omega_k^\theta(p^G) \leq \max_{1\leq k\leq K}\omega_k^\theta(p^{opt}) \leq \theta\cdot W + H. \quad (24)$$

It follows from the definition of $\omega_k^\theta(e)$ that $\omega_k^\theta(p^G) \geq \theta\cdot\omega_k(p^G)$. Therefore (24) implies

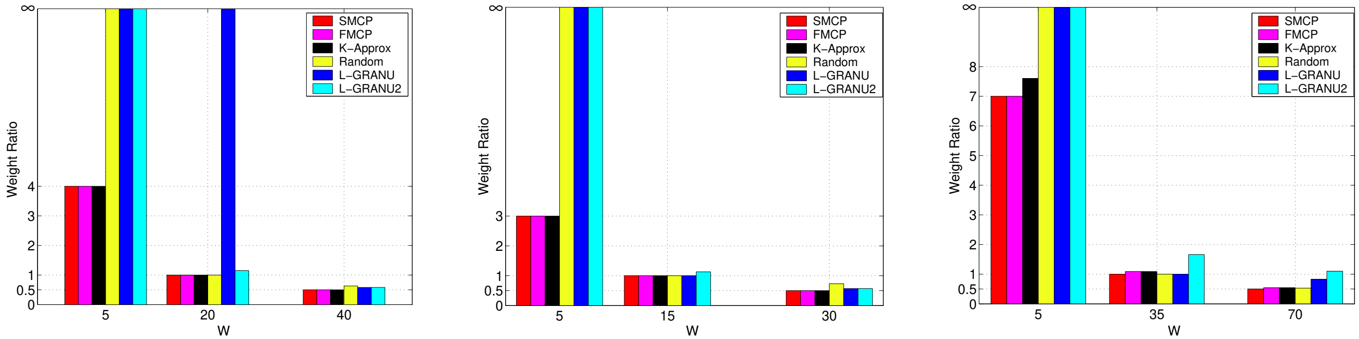
$$\omega_k(p^G) \leq \omega_k^\theta(p^G)/\theta \leq \frac{\theta\cdot W + H}{\theta} = W + \mathsf{LB}\cdot W\cdot\epsilon$$
$$\leq (1+\epsilon)\cdot W, \ \forall\ 1 \leq k \leq K. \quad (25)$$

This proves that p^G is guaranteed to be a $(1+\epsilon)$-approximation to $\mathsf{FMCP}(G,s,t,K,W)$ and that Algorithm 4 must stop after p^G is computed. Since this is the first time we have entered Step_2 with $H \geq \mathcal{H}$, we conclude that $H \leq 2\cdot\mathcal{H}$. Therefore the running time in this case is $O(n\log n + m(\frac{\mathcal{H}}{\epsilon})^{K-1})$. □

We point out some features of FPTAS-FMCP. When $\mathsf{DMCP}(G,s,t,K,W)$ is feasible, it runs very fast, often at the speed of K-Approx. When $\mathsf{DMCP}(G,s,t,K,W)$ is infeasible, it still computes a *provably good path*, although the running time could be larger than that of FPTAS-SMCP. When K is reduced to 2, its corresponding running time is faster than that of [7] (again note that the goal here is slightly different from that in [7]).

VI. NUMERICAL RESULTS

To verify the theoretical analysis of the algorithms presented in this paper, we implemented K-Approx, FPTAS-SMCP and FPTAS-FMCP and compared them with Yuan's heuristic (L-GRANU), Yuan's reverse heuristic (L-GRANU2), as well as Korkmaz and Krunz's randomized heuristic (RANDOM). The tests were performed on a 2.4GHz Linux PC with 1G bytes of memory.



(a) NSFNET (b) ARPANET (c) Italian National Network

Fig. 1. Ratio of path weight vs constraint.

We used well known Internet topologies to verify the suitability of the algorithms, and randomly generated topologies to verify the computational scalability of the algorithms. The known Internet topologies include NSFNET (14 nodes and 21 edges) [4], ARPANET (20 nodes and 32 edges) [1], and Italian National Network (33 nodes and 67 edges) [1]. As in [3], [7], [13], [14], [28], the edge weights were uniformly generated in a range (we used the range $[1, 10]$). From our analysis, one should expect our algorithms to perform similarly on various edge weights. For each topology, we used the same (s, t) pair for all algorithms. For a fair comparison with the limited granularity heuristic, we used $1/\epsilon$ as the parameter n in [28]. For each topology, we tested three different values of W: a *small value* of W such that $\mathsf{DMCP}(G, s, t, K, W)$ is infeasible; a *large value* of W such that $\mathsf{DMCP}(G, s, t, K, W)$ is feasible; and a *larger value* of W such that $\mathsf{DMCP}(G, s, t, K, W/2)$ is feasible. We performed the tests with $K = 3$.

For NSFNET, we first set W to 5 (the small value of W). FPTAS-SMCP, FPTAS-FMCP and K-Approx were all able to find a path whose three path weights are $20, 20, 14$, for each of $\epsilon \in \{1.0, 0.5, 0.2, 0.1\}$. However, RANDOM, L-GRANU and L-GRANU2 all failed to find any path. Next, we set $W = 20$. This time, RANDOM was able to find a path with path weights $20, 20, 14$. L-GRANU2 was able to find a path with path weights $19, 21, 23$ for $\epsilon = 0.5, 0.2$. For $\epsilon = 0.1$, L-GRANU2 was able to find a path with path weights $20, 20, 14$. However, L-GRANU still failed to find a path. Finally, we set $W = 40$. This time, all algorithms were able to find a path. Figure 1(a) illustrates the quality of the paths found (with $\epsilon = 0.5$), where the y-axis measures the maximum ratio of the path weight over the path constraint for the path found by the different algorithms. In case no path was found, we treat this ratio as infinity. Figures 1(b) and 1(c) illustrate the results for ARPANET and Italian National Network respectively. We can see that in all cases, FPTAS-SMCP, FPTAS-FMCP and Approx find a good path.

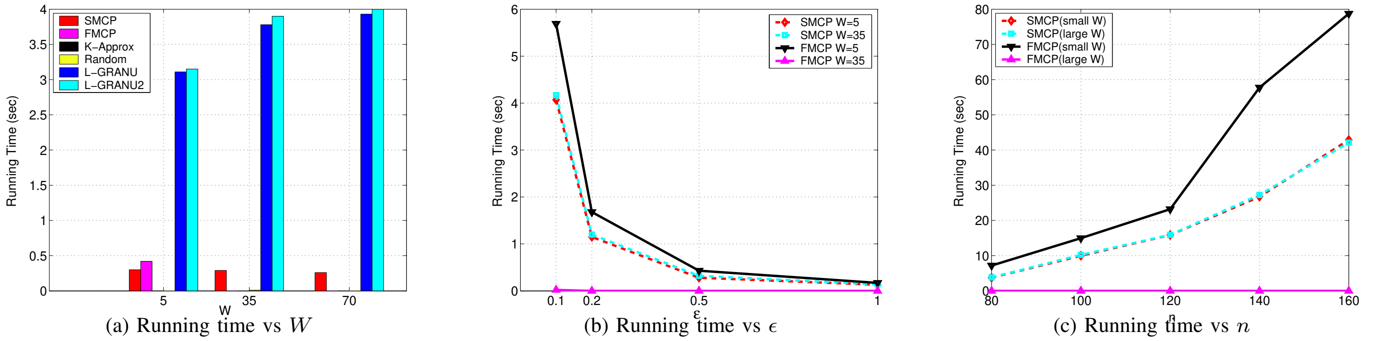
Figure 2(a) illustrates the running times (in seconds) of the different algorithms, as well as their dependency on the value of W, using the case of $\epsilon = 0.5$ for Italian National Network. As expected, K-Approx and RANDOM are always the fastest. Also, we observe that both FPTAS-SMCP and FPTAS-FMCP are much faster than L-GRANU and L-GRANU2. As expected, the running time of FPTAS-SMCP is independent of W, while the running time of FPTAS-FMCP is either slightly larger than that of FPTAS-SMCP (when W is small), or very small (when W is large). We also observe that the running times of L-GRANU and L-GRANU2 may increase with W slightly, but not significantly. This is due to the fact that more edge relaxations may be performed by L-GRANU and L-GRANU2 for larger values of W.

Figure 2(b) illustrates the running times (in seconds) of our two FPTASs as functions of ϵ, using the case of $W = 5, 35$ for Italian National Network. As expected, the running times increase with $\frac{1}{\epsilon}$. Again, we note that the running time of FPTAS-SMCP is independent of W, while the running time of FPTAS-FMCP is either slightly larger than that of FPTAS-SMCP (for small W) or very small (for large W).

To study the scalability of our FPTASs with the network size, we also tested FPTAS-SMCP and FPTAS-FMCP on large network topologies generated by BRITE, a well known Internet topology generator [2]. The values of W were chosen similarly as in the case of well known topologies. We report results with $\epsilon = 0.5$ for all topology sizes for illustration.

BRITE provides several well-known models (including the Waxman model [24]) for generating reasonable network topologies. We adopted the Waxman model (with default parameters provided by BRITE) to generate random networks. We used five different numbers of nodes: $80, 100, 120, 140, 160$. Correspondingly, BRITE generated five network topologies with the following sizes: (1) 80 nodes with 314 edges, (2) 100 nodes with 390 edges, (3) 120 nodes with 474 edges, (4) 140 nodes with 560 edges, (5) 160 nodes with 634 edges. For each topology, we ran 10 test cases. For each test case, we randomly generated a source-destination node pair (s, t) and used this pair for all tested algorithms. For this node pair, we used a small value of W and a large value of W to test the algorithms (note that these values of W may change when the node pair changes). The running times of our algorithms are shown in Figure 2(c), where the running time shown for each algorithm is the average over 10 cases, where the largest standard deviation is 1.29.

We observe that the running times of all algorithms (expect



(a) Running time vs W

(b) Running time vs ϵ

(c) Running time vs n

Fig. 2. Running time vs various factors.

that of FPTAS-FMCP with large W) increase with the increase of the network size. For the random networks generated, m is approximately $4n$. Since we used $K = 3$ and $\epsilon = 0.5$, the running times of both FPTAS-SMCP and FPTAS-FMCP should be $O(m(\frac{n}{0.5})^2) = O(n^3)$ for the cases tested. For small values of W, FPTAS-FMCP may have to solve instances of RMCP with τ set to $7H$ (note that $\lfloor \frac{2\mathsf{UB}}{\mathsf{LB}\cdot\epsilon} \rfloor = 6$ in this case) for $H = 1, 2, 4, \ldots, 2^{\lfloor \log_2(n-1) \rfloor}, n-1$. Note that for the same setting, FPTAS-SMCP only needed to solve one instance of RMCP with τ set to $7(n-1)$. Since the running time of PseudoRMCP is proportional to τ^{K-1}, we can use $1 + [1 + \frac{1}{4} + (\frac{1}{4})^2 + \ldots] = \frac{7}{3}$ as an estimate of the maximum of the ratio of the running time of FPTAS-FMCP over the running time of FPTAS-SMCP. Figure 2(c) conforms to this analysis.

In terms of running time, most heuristic algorithms are fast, but do not guarantee quality of solutions. There could be situations where they will produce solutions that deviate from the optimal solutions to a considerable extent. So, RANDOM is quite fast since it requires $K+1$ shortest path computations. K-Approx is the fastest since it requires only one shortest path computation. We note that K-Approx produces solutions better than or as good as the solutions produced by RANDOM whenever the latter is successful. FPTASs take more time as the value of ϵ chosen becomes smaller. In other words, the more accuracy required for the solution, the more is the time taken. This is not surprising since FPTASs provide guarantees on the accuracy of the solutions produced. This guarantee requirement is the reason for large running times taken by the FPTASs. K-Approx is a constant factor approximation algorithm with the constant equal to the number of edge weights. However, simulation results in comparison with those of our FPTASs which guarantee accuracy of the solutions (specified arbitrarily by the value of ϵ), provide evidence that K-Approx performs quite well in practice.

VII. Conclusions

In this paper, we have studied the MCP problem with K additive QoS constraints, where $K \geq 2$ is a fixed constant. We presented a novel $O(Km + n \log n)$ time K-approximation algorithm which uses a *single* auxiliary edge weight to compute a shortest path. Because of this property, the algorithm is easily implementable in a hop-by-hop environment. We also presented two FPTASs for two slightly different versions of the problem whose time complexities, when reduced to the case of $K = 2$, compare favorably with existing algorithms. To implement the FPTASs proposed in this paper in the current networking environment, some careful modifications will be necessary. The routing tables will have to store the next hop addresses for every source destination pair for a few discrete values of ϵ. These values of ϵ may be used to determine the traffic classes in the network. It may be noted that the results presented in this paper, although derived under the model of an undirected graph, are equally valid for the case of directed graphs.

Acknowledgment

The authors wish to thank the associate editor and the anonymous reviewers whose valuable comments on earlier versions of this paper helped to improve the quality of the paper significantly.

References

[1] R. Andersen, F. Chung, A. Sen and G. Xue; On disjoint path pairs with wavelength continuity constraint in WDM networks; *IEEE Infocom'04*, pp. 524–535.

[2] BRITE; http://www.cs.bu.edu/brite/.

[3] S. Chen and K. Nahrstedt; On finding multi-constrained paths; *IEEE ICC'1998*; pp. 874–879.

[4] X. Chu and B. Li, Dynamic routing and wavelength assignment in the presence of wavelength conversion for all-optical networks; *IEEE/ACM Trans.on Networking*,Vol.13(2005),pp.704-715.

[5] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein; *Introduction to Algorithms*; second edition; McGraw Hill, 2001.

[6] F. Ergun, R. Sinha and L. Zhang; An improved FPTAS for restricted shortest path; *Information Processing Letters*; Vol. 83(2002), pp. 287–291.

[7] A. Goel, K.G. Ramakrishnan, D. Kataria and D. Logothetis; Efficient computation of delay-sensitive routes from one source to all destinations; *IEEE Infocom'2001*; pp. 854–858.

[8] R. Guerin and A. Orda; QoS routing in networks with inaccurate information: theory and algorithms; *IEEE/ACM Transactions on Networking*; Vol. 7(1999), pp. 350–364.

[9] R. Hassin; Approximation schemes for the restricted shortest path problem; *Mathematics of Operations Research*; vol. 17(1992), pp. 36–42.

[10] C. Huitema, *Routing in the Internet*, 2nd ed., Prentice Hall PTR, 2000.

[11] O.H. Ibarra and C.E. Kim; Fast approximation algorithms for the knapsack and sum of subset problems; *Journal of the ACM*; Vol. 22(1975), pp. 463–468.

[12] J.M. Jaffe; Algorithms for finding paths with multiple constraints; *Networks*; Vol. 14(1984), pp. 95–116.

[13] A. Juttner, B. Szviatovszki, I. Mecs and Z. Rajko; Lagrange relaxation based method for the QoS routing problem; *IEEE Infocom'2001*, pp. 859–868.

[14] T. Korkmaz and M. Krunz; Mult-constrained optimal path selection; *IEEE Infocom'2001*; pp. 834–843.

[15] D.H. Lorenz and D. Raz; A simple efficient approximation scheme for the restricted shortest path problem; *Operations Research Letters*; Vol. 28(2001), pp. 213–219.

[16] Q. Ma and P. Steenkiste; Quality-of-service routing for traffic with performance guarantees; *IWQoS'97*, May 1997.

[17] M.A. Marsan, G. Corazza, M. Listanti and A. Roveri (eds); Quality of service in multiservice IP networks; *Lecture Notes in Computer Science*, Vol. 2601, 2003.

[18] A. Orda; Routing with end-to-end QoS guarantees in broadband networks; *IEEE/ACM Transactions on Networking*; Vol. 7(1999), pp. 365–374.

[19] A. Orda and A. Sprintson; Precomputation schemes for QoS routing; *IEEE/ACM Transactions on Networking*; Vol. 11(2003), pp. 578–591.

[20] A. Orda and A. Sprintson; Efficient algorithms for computing disjoint QoS paths; *IEEE Infocom'2004*; pp. 727–738.

[21] S. Sahni; General techniques for combinatorial approximation; *Operations Research*, Vol. 25(1977), pp. 920–936.

[22] Z. Wang and J. Crowcroft; Quality-of-service routing for supporting multimedia applications; *IEEE Journal on Selected Areas in Communications*; Vol. 14(1996), pp. 1228–1234.

[23] A. Warburton; Approximation of Pareto optima in multiple-objective, shortest path problem; *Operations Research*; Vol. 35(1987), pp. 70–79.

[24] B.M. Waxman; Routing of multipoint connections; *IEEE Journal on Selected Areas in Communications*; Vol. 6(1988), pp. 1617-1622.

[25] G. Xue; Primal-dual algorithms for computing weight-constrained shortest paths and weight-constrained minimum spanning trees; *IEEE IPCCC'2000*; pp. 271–277.

[26] G. Xue; Minimum cost QoS multicast and unicast routing in communication networks; *IEEE Transactions on Communications*; Vol. 51(2003), pp. 817–824.

[27] G. Xue, A. Sen and R. Banka; Routing with many additive QoS constraints; *ICC'2003: IEEE International Conference on Communications*; pp. 223–227.

[28] X. Yuan; Heuristic algorithms for multiconstrained quality-of-service routing; *IEEE/ACM Transactions on Networking*; Vol. 10(2002), pp. 244–256.

Guoliang Xue is a Full Professor in the Department of Computer Science and Engineering at Arizona State University. He received the Ph.D degree in Computer Science from the University of Minnesota in 1991 and has held previous positions at the Army High Performance Computing Research Center and the University of Vermont. His research interests include efficient algorithms for optimization problems in networking, with applications to fault tolerance, robustness, and privacy issues in networks ranging from WDM optical networks to wireless ad hoc and sensor networks. He has published over 130 papers in these areas. His research has been continuously supported by federal agencies including NSF and ARO. He is the recipient of an NSF Research Initiation Award in 1994, and an NSF-ITR Award in 2003. He is an Associate Editor of Computer Networks (COMNET), the IEEE Network Magazine, and the IEEE Transactions on Circuits and Systems-I: Regular Papers. He has served on the executive/program committees of many IEEE conferences, including Infocom, Secon, Icc, Globecom and QShine. He served as the General Chair of *IEEE International Performance, Computing, and Communications Conference* in 2005, and will serve as a TPC co-chair of *IEEE Globecom'2006 Symposium on Wireless Ad Hoc and Sensor Networks*, as well as a TPC co-chair of *IEEE ICC'2007 Symposium on Wireless Ad Hoc and Sensor Networks*. He is a senior member of IEEE.

Arunabha Sen received the XYZ degrees in ABCD from... His research interests include ...

Weiyi Zhang received the B.E. and M.E. degrees from Southeast University, China, in 1999 and 2002 respectively. Currently he is a Ph.D student in the Department of Computer Science and Engineering at Arizona State University. His research interests include reliable communication in networking, protection and restoration in WDM networks, and QoS provisioning in communication networks.

Jian Tang received the B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, China, in 1998 and 2001 respectively. Currently, he is a Ph.D candidate in the Computer Science and Engineering Department of Arizona State University. His research interest is in the area of wireless networking and mobile computing with emphases on routing, scheduling and cross-layer design in wireless networks. He is a student member of IEEE.

Krishnaiyan Thulasiraman received the Bachelor's degree (1963) and Master's degree (1965) in electrical engineering from the university of Madras, India, and the Ph.D degree (1968) in electrical engineering from IIT, Madras, India. He holds the Hitachi Chair and is Professor in the School of Computer Science at the University of Oklahoma, Norman, where he has been since 1994. Prior to joining the University of Oklahoma, Thulasiraman was professor (1981-1994) and chair (1993-1994) of the ECE Department in Concordia University, Montreal. He was on the faculty in the EE and CS departments of the IITM during 1965-1981.

Dr. Thulasiraman's research interests have been in graph theory, combinatorial optimization, algorithms and applications in a variety of areas in CS and EE: electrical networks, VLSI physical design, systems level testing, communication protocol testing, parallel/distributed computing, telecommunication network planning, fault tolerance in optical networks, interconnection networks etc. He has published more than 100 papers in archival journals, coauthored with M. N. S. Swamy two text books "Graphs, Networks, and Algorithms" (1981) and "Graphs: Theory and Algorithms" (1992), both published by Wiley Inter-Science, and authored two chapters in the Handbook of Circuits and Filters (CRC and IEEE, 1995) and a chapter on "Graphs and Vector Spaces " for the handbook of Graph Theory and Applications (CRC Press,2003).

Dr. Thulasiraman has received several awards and honors: Endowed Gopalakrishnan Chair Professorship in CS at IIT, Madras (Summer 2005), Elected member of the European Academy of Sciences (2002), IEEE CAS Society Golden Jubilee Medal (1999), Fellow of the IEEE (1990) and Senior Research Fellowship of the Japan Society for Promotion of Science (1988). He has held visiting positions at the Tokyo Institute of Technology, University of Karlsruhe, University of Illinois at Urbana-Champaign and Chuo University, Tokyo.

Dr. Thulasiraman has been Vice President (Administration) of the IEEE CAS Society (1998, 1999), Technical Program Chair of ISCAS (1993, 1999), Deputy Editor-in-Chief of the IEEE Transactions on Circuits and Systems I (2004-2005), Co-Guest Editor of a special issue on "Computational Graph Theory: Algorithms and Applications" (IEEE Transactions on CAS , March 1988), , Associate Editor of the IEEE Transactions on CAS (1989-91, 1999-2001), and Founding Regional Editor of the Journal of Circuits, Systems, and Computers and an editor of the AKCE International Journal of Graphs and Combinatorics. Recently, he founded the Technical Committee on "Graph theory and Computing" of the IEEE CAS Society.