IEEE TRANSACTIONS ON
EMERGING TOPICS IN COMPUTING

Received 30 November 2014; revised 9 February 2015; accepted 27 April 2015.
Date of Publication 5 May 2015; date of current version 8 June 2016.

Digital Object Identifier 10.1109/TETC.2015.2428654

QoS Routing Under Multiple Additive Constraints: A Generalization of the LARAC Algorithm

YING XIAO[1], KRISHNAIYAN THULASIRAMAN[2], (Fellow, IEEE), GUOLIANG XUE[3], (Fellow, IEEE), AND MAMTA YADAV[2], (Student Member, IEEE)

[1]VT iDirect Inc., Herndon, VA 20171 USA
[2]School of Computer Science, University of Oklahoma, Norman, OK 73019 USA
[3]School of Computing, Informatics and Decision Systems Engineering Arizona State University, Tempe, AZ 85287 USA
CORRESPONDING AUTHOR: K. THULASIRAMAN (thulasi@ou.edu)

The work of K. Thulasiraman was supported by the National Science Foundation (NSF) Information Technology Research under Grant ANI-0312435. The work of G. Xue was supported in part by the U.S. Army Research Office under Grant W911NF-09-1-0467 and in part by NSF under Grant 1217611 and Grant 1421685.

ABSTRACT Consider a directed graph $G(V, E)$, where V is the set of nodes and E is the set of links in G. Each link (u, v) is associated with a set of $k + 1$ additive nonnegative integer weights $C_{uv} = (c_{uv}, w_{uv}^1, w_{uv}^2, \ldots, w_{uv}^k)$. Here, c_{uv} is called the cost of link (u, v) and w_{uv}^i is called the ith delay of (u, v). Given any two distinguish nodes s and t, the QoS routing (QSR) problem QSR(k) is to determine a minimum cost $s - t$ path such that the i^{th} delay on the path is atmost a specified bound. This problem is NP-complete. The LARAC algorithm based on a relaxation of the problem is a very efficient approximation algorithm for QSR(1). In this paper, we present a generalization of the LARAC algorithm called GEN-LARAC. A detailed convergence analysis of GEN-LARAC with simulation results is given. Simulation results provide an evidence of the excellent performance of GEN-LARAC. We also give a strongly polynomial time approximation algorithm for the QSR(1) problem.

INDEX TERMS QoS routing, Lagrangian relaxation, combinatorial optimization, algorithms.

I. INTRODUCTION

Shortest path, minimum cost flow, and maximum flow computation are fundamental problems in operations research. Though interesting in their own right, algorithms for these problems also serve as building blocks in the design of algorithms for complex problems encountered in large scale industrial applications. Hence, over the years there has been an extensive literature on various aspects of these problems. These problems are solvable in polynomial time. But adding one or more additional additive constraints makes them intractable.

In this paper, we focus on the constrained shortest path (CSP) problem also known as the QoS routing (QSR) problem. This problem requires determination of a minimum cost path from a source node to a destination node of a network subject to bounds on certain additive metrics on the path. For instance, the additive metric commonly considered is the total delay of the path. The QoS problem has attracted considerable attention from different research communities: operations research, computer science, and telecommunication networks. The interest from the telecommunications community arises from the great deal of emphasis on the need to design communication protocols that deliver certain performance guarantees. This need, in turn, is the result of an explosive growth in high bandwidth real time applications that require stringent quality of service guarantees.

A. REVIEW OF LITERATURE

The literature on the QoS routing problem is vast. Hence in this subsection we survey only a subset of published works in this area. We first review the literature on the QoS routing problem under one additive metric followed by the literature on several additive metrics and other generalizations.

1) QoS ROUTING PROBLEM UNDER ONE ADDITIVE METRIC

It has been shown in [5], [29] that the QoS routing problem is NP-hard even for acyclic networks. So, in the literature, heuristic approaches and approximate algorithms have been proposed. Heuristics, in general, do not provide performance guarantees on the quality of the solution produced, though they are usually fast in practice. On the other hand, ϵ-approximation algorithms deliver solutions with cost within $(1+\epsilon)$ time the optimal cost, but are usually very slow in practice because they guarantee the quality of the solutions. Hassin [7] gave the first $\epsilon-$ approximation algorithm for the QoS routing problem under one additive metric.

As regards heuristics, a number of them have appeared in the literature providing different levels of performance with regard to the quality of the solution as well as the computation time required. For instance, the LHWHM algorithm [17] is a very simple heuristic which is very fast requiring only two invocations of Dijkstra's shortest path algorithm for a feasible problem. Reference [25] also discusses further enhancements of the LHWHM algorithm as well as a heuristic based on the Bellman-Ford-Moore (BFM) algorithm for the shortest path problem. It should be emphasized that in all these cases, only simulations are used to evaluate the performance of the algorithms. A comprehensive overview of a number of QoS routing algorithms may be found in [16].

There are heuristics that are based on sound theoretical foundations. These algorithms are based on solutions to the integer relaxation or the dual of the integer relaxation of the QoS routing problem. To the best of our knowledge, the first such algorithm was reported in [6] by Handler and Zhang. This is based on the geometric approach (also called the hull approach [19], [41]). More recently, in an independent work, Jüttner et al. [9] developed the LARAC algorithm which solves the Lagrangian relaxation of the QoS routing problem. In another independent work, Blokh and Gutin [3] defined a general class of combinatorial optimization problems (that are called the MCRT problems, namely, Minimum Cost Restricted Time Combinatorial Optimization problems) of which the QoS routing problem is a special case, and proposed an approximation algorithm to this problem. Xiao et al. [32] drew attention to the fact that the algorithms in [3] and [9] are equivalent. Mehlhorn and Ziegelmann [19] and Ziegelmann [41] have also observed this equivalence and have developed several insightful results. In view of this equivalence, we refer to these algorithms as the LARAC algorithm. The work in [32] also establishes certain results using the algebraic approach. These results also hold true in the case of the general optimization problem considered in [6]. In another independent work, Xue [39] also arrived at the LARAC algorithm using the primal-dual method of linear programming. Usually, approximation algorithms are developed using the dual of the relaxed version of the QoS routing problem. Xiao et al. [33] described an efficient approximation algorithm for the QoS routing problem using the primal simplex method of linear programming. In [10], Jüttner proved the strong polynomiality of the LARAC algorithm, both for the general case and for the QoS routing problem. He has used certain results from the general area of fractional combinatorial optimization. Jüttner [11] gave a general method to solve budgeted optimization problems in strongly polynomial time. Radzik [24] gives an excellent exposition of approaches to fractional combinatorial optimization problems.

B. MULTI-CONSTRAINED AND DISJOINT PATH ROUTING

An application of the parametric search method to the general class of combinatorial optimization problems involving two additive parameters may be found in [12]. Multi-constrained routing problem (routing under several metrics) has been considered in [8], [13], [15], [37]–[40]. Several interesting algorithms related to the QoS routing problem and motivated by applications have appeared in the literature. The constrained disjoint shortest path problem has been studied in [28] and [32]. [2], [4], [20] and [26] discussed issues involving reliability and hop constraints, [21] and [22] introduced the concept of exact QoS routing and tunability, and [14], [30], [34], [36] discussed constrained disjoint paths problem, routing under uncertainty and related approximation schemes.

Summarizing the LARAC algorithm and other variants of the mathematical programming based approach have been reported for the QoS routing problem under one additive metric. But no generalization of the LARAC algorithm applicable to multiple additive metric has been studied. In view of the highly efficient performance of the LARAC algorithm such a generalization is of great interest. In this paper we present such a generalization. This is an edited version of our earlier conference paper [35].

The paper is organized as follows. In Section II, we review the QoS routing problem and the LARAC algorithm of [9] for this problem. In Section III we develop GEN-LARAC, a generalization of the LARAC algorithm for the QoS routing problem with multiple additive constraints. In this section we also develop a strongly polynomial time algorithm for the integer relaxation of the special case of one additive metric. A detailed analysis of the convergence properties is also given with simulation results providing evidence of the excellent performance of GEN-LARAC.

II. QoS ROUTING (QSR) PROBLEM AND THE LARAC ALGORITHM

First we give a formal definition of the QoS routing problem. We use the terms links and nodes for edges and vertices, respectively, following the convention in the networking literature.

A. QoS Routing Problem (QSR)

Consider a network $G(V, E)$. Each link $(u, v) \in E$ is associated with two weights $c_{uv} > 0$ (say, cost) and $d_{uv} > 0$ (say, delay). Also are given two distinguished nodes s and t and a real number $\Delta > 0$. Let P_{st} denote the set of all

$s-t$ paths and for any $s-t$ path p, define

$$c(p) = \sum_{(u,v)\in p} c_{uv} \qquad \text{and} \qquad d(p) = \sum_{(u,v)\in p} d_{uv}.$$

Let $P_{st}(\Delta)$ be the set of all the $s-t$ paths p such that $d(p) \le \Delta$. A path in the set $P_{st}(\Delta)$ is called a feasible path. The QSR problem is to find a path $p^* = \arg\ min\{c(p)|p \in P_{st}(\Delta)\}$. In other words, the QSR problem is to find a minimum cost feasible $s-t$ path. It can be formulated as the following integer linear program.

QSR:

$$\text{Minimize} \quad \sum_{(u,v)\in E} c_{uv}x_{uv}$$

$$\text{subject to} \quad \forall u \in V,$$

$$\sum_{\{v|(u,v)\in E\}} x_{uv} - \sum_{\{v|(v,u)\in E\}} x_{vu} = \begin{cases} 1, & for\ \ u = s \\ -1, & for\ \ u = t \\ 0, & otherwise \end{cases}$$

$$\sum_{(u,v)\in E} -d_{uv} \cdot x_{uv} - w = -\Delta, \quad w \ge 0$$

$$x_{uv} = 0 \text{ or } 1, \quad \forall (u,v) \in E$$

The QSR problem is known to be NP-hard [5], [29]. The main difficulty lies with the integrality condition that requires that the variables x_{uv} be 0 or 1. Removing or relaxing this requirement from the above integer linear program and letting $x_{uv} \ge 0$ leads to RELAX-QSR, the relaxed QSR problem. It is often convenient to solve the dual of the relaxed form of the QSR problem which we present below.

The dual involves $s-t$ paths and a variable $\lambda \ge 0$. For each link (u,v), let the aggregated cost c_λ be defined as $c_{uv} + \lambda d_{uv}$. For a given λ, let $c_\lambda(p)$ denote the aggregated cost of the path p. Finally define $L(\lambda)$ as:

$$L(\lambda) = \min\{c_\lambda(p)|p \in P_{st}\} - \lambda\Delta. \qquad (1)$$

Note that in the above, $\min\{c_\lambda(p)|p \in P_{st}\}$ is the same as the minimum aggregated cost of an $s-t$ path with respect to a given value of λ. This can be easily obtained by applying Dijkstra's algorithm using aggregated link costs. Let the $s-t$ path which has minimum aggregated cost with respect to a given λ be denoted as p_λ. Then $L(\lambda) = c_\lambda(p_\lambda) - \lambda\Delta$ and the dual of the RELAX-QSR can be presented in the following form.

$$\textbf{DUAL} - \textbf{RELAX} - \textbf{QSR}: \text{Find } L^* = \max\{L(\lambda)|\lambda \ge 0\}.$$

We note that the problem of maximizing $L(\lambda)$ as above is also called the Lagrangian dual problem. The value of λ that achieves the maximum $L(\lambda)$ in DUAL-RELAX-QSR will be denoted by λ^*. Note that L^*, the optimum value of DUAL-RELAX-QSR is a lower bound on the optimum cost of the path solving the corresponding QSR problem. The key issue in solving DUAL-RELAX-QSR is how to search for the optimal λ and determining the termination condition for the search. The LARAC algorithm of [9] presented in Figure 1 is one such efficient search procedure.

Procedure LARAC(s,t,d,Δ)

Step 1. $p_c := Dijkstra(s,t,c)$
    if $d(p_c) \le \Delta$ then return p_c

Step 2. $p_d := Dijkstra(s,t,d)$
    if $d(p_d) > \Delta$ then
        return "there is no solution"

Step 3. **repeat**

$$\lambda := \frac{c(p_c) - c(p_d)}{d(p_d) - d(p_c)}$$

    $r := Dijkstra(s,t,c_\lambda)$
    if $c_\lambda(r) = c_\lambda(p_c)$ then return p_d
    else if $d(r) \le \Delta$ then $p_d := r$
    else $p_c := r$

end repeat
end procedure

FIGURE 1. **LARAC algorithm.**

B. Description of the Algorithm

In the LARAC algorithm $Dijkstra(s,t,c)$, $Dijkstra(s,t,d)$ and $Dijkstra(s,t,c_\lambda)$ denote, respectively, Dijkstra's shortest path algorithm using link costs, link delays, and aggregated link costs with respect to the multiplier λ.

1. In the first step, the algorithm calculates the shortest path on link costs. If the path found meets the delay constraint, this is surely the optimal path. Otherwise, the algorithm stores the path as the latest infeasible path, simply called the p_c path. Then it determines the shortest path on link delays denoted as p_d. If p_d is infeasible, there is no solution to this instance.
2. Set $\lambda = (c(p_c) - c(p_d))/(d(p_d) - d(p_c))$. With this value of λ, we can find a new c_λ-minimal path r. If $c_\lambda(r) = c_\lambda(p_c) = c_\lambda(p_d)$), we have obtained the optimal λ as proved in [9] and [32]. Otherwise, set r as the new p_c or p_d according to whether r is infeasible or feasible.

An algebraic study of the LARAC algorithm and an efficient implementation called the LARAC-BIN algorithm has been presented in [32].

III. GEN-LARAC: A GENERALIZED APPROACH TO THE CONSTRAINED SHORTEST PATH PROBLEM UNDER MULTIPLE ADDITIVE CONSTRAINTS

In this section we study the QSR(k) problem that requires determination of $s-t$ paths that satisfy $k > 1$ additive constraints. We develop a new approach using Lagrangian relaxation. We use the LARAC algorithm as a building block in the design of this approach.

A. FORMULATION OF THE QSR(K) PROBLEM AND ITS RELAXATION

Consider a directed graph $G(V, E)$ where V is the set of nodes and E is the set of links in G. Each link (u, v) is associated with a set of $k + 1$ additive non-negative integer weights $C_{uv} = (c_{uv}, w^1_{uv}, w^2_{uv}, \ldots, w^k_{uv})$. Here c_{uv} is called the cost of link (u, v) and w^i_{uv} is called the i^{th} delay of (u, v). Given two nodes s and t, an $s - t$ path in G is a directed simple path from s to t. Let P_{st} denote the set of all $s - t$ paths in G. For an $s - t$ path p define

$$c(p) \equiv \sum_{(u,v)\in p} c_{uv} \quad \text{and} \quad d_i(p) \equiv \sum_{(u,v)\in p} w^i_{uv}, \quad i = 1, \ldots, k.$$

The value $c(p)$ is called the cost of path p, and $d_i(p)$ is called the i^{th} delay of path p. Given k positive integers $r_1, r_2, \ldots, r_k$, an $s - t$ path is called feasible (*resp*. strictly feasible) if $d_i(p) \leq r_i$(*resp*. $d_i(p) < r_i$), for all $i = 1, 2, \ldots, k$ (r_i is called the bound on the i^{th} delay of a path).

The QSR(k) problem is to find a minimum cost $s - t$ path, such that for all i, the i^{th} delay of the path is at most a specified upper bound r_i. An instance of the QSR(k) problem is strictly feasible if all the feasible paths are strictly feasible. Without loss of generality, we assume that the problem under consideration is always feasible. In order to guarantee strict feasibility, we do the following transformation.

For $i = 1, 2, \ldots, k$, transform the i^{th} delay of each link (u, v) such that the new weight vector C'_{uv} is given by

$$C'_{uv} = (c_{uv}, 2w^1_{uv}, 2w^2_{uv}, \ldots, 2w^k_{uv}).$$

Also transform the bounds $r'_i s$ so that the new vector of bounds R' is given by

$$R' = (2r_1 + 1, 2r_2 + 1, \ldots, 2r_k + 1).$$

In the rest of the section, we only consider the transformed problem. Thus all link delays are even integers, and delay bounds are odd integers. We will use symbols with capital or bold letters to represent vectors. Also, for a matrix A, A^T denotes its transpose. For simplicity of presentation, we will use C_{uv} and R instead of C'_{uv} and R' to denote the transformed weight vector and the vector of bounds.

Two immediate consequences of this transformation are stated below.

Lemma 1: $\forall\, p \in P_{st}, \forall\, i \in \{1, 2, \ldots, k\}$, $d_i(p) \neq r_i$ *in the transformed problem.*

Lemma 2: An $s - t$ path in the original problem is feasible (resp. optimal) iff it is strictly feasible (resp. optimal) in the transformed problem.

Starting with an ILP (Integer Linear Programming) formulation of the QSR(k) problem and relaxing the integrality constraints we get the RELAX-QSR(k) problem below. In this formulation, for each $s - t$ path p, we introduce a variable x_p.

RELAX − QSR(*k*):

$$\text{Minimize} \quad \sum_p c(p)x_p \tag{2}$$

$$\text{subject to} \quad \sum_p x_p = 1 \tag{3}$$

$$\sum_p d_i(p)x_p \leq r_i, \quad i = 1, \ldots, k \tag{4}$$

$$x_p \geq 0, \quad \forall\, p \in P_{st} \tag{5}$$

The Lagrangian dual of RELAX-QSR(k) is given below.

DUAL − RELAX − QSR(*k*):

$$\text{Maximize} \quad w - \lambda_1 r_1 \cdots - \lambda_k r_k \tag{6}$$

$$\text{subject to} \quad w - d_1(p)\lambda_1 \cdots - d_k(p)\lambda_k \leq c(p), \quad \forall p \in P_{st} \tag{7}$$

$$\lambda_i \geq 0, \quad i = 1, \ldots, k \tag{8}$$

In the above dual problem $\lambda_1, \lambda_2, \ldots, \lambda_k$ and w are the dual variables, with w corresponding to (3) and each λ_i corresponding to the i^{th} constraint in (4).

It follows from (7) that $w \leq c(p) + d_1(p)\lambda_1 + \cdots + d_k(p)\lambda_k, \forall p \in P_{st}$. Since we want to maximize (6), the value of w should be as large as possible, i.e.

$$w = \min_{p\in P_{st}}\{c(p) + d_1(p)\lambda_1 + \cdots + d_k(p)\lambda_k\}.$$

With the vector Λ defined as $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$, define

$$L(\Lambda) = \min_{p\in P_{st}}\{c(p) + \lambda_1(d_1(p) - r_1) + \cdots + \lambda_k(d_k(p) - r_k)\}. \tag{9}$$

Notice that $L(\Lambda)$ is called the Lagrangian function in the literature and is a concave continuous function of Λ.

Then DUAL-RELAX-QSR(k) can be written as follows.

DUAL − RELAX − QSR(*k*):

$$\text{Maximize} \quad L(\Lambda)$$

$$\text{subject to} \quad \Lambda \geq 0 \tag{10}$$

The Λ^* that maximizes (10) is called the maximizing multiplier and is defined as

$$\Lambda^* = \arg\max_{\Lambda\geq 0} L(\Lambda) \tag{11}$$

Claim 1: If an instance of the QSR(k) problem is feasible and a path p_{opt} is an optimal path, then $\forall\Lambda \geq 0$, $L(\Lambda) \leq c(p_{opt})$.

We shall use $L(\Lambda)$ as a lower bound of $c(p_{opt})$ to evaluate the quality of the approximate solution obtained by our algorithm. Given $p \in P_{st}$ and Λ, define

$$C(p) \equiv (c(p), d_1(p), d_2(p), \ldots, d_k(p)),$$
$$D(p) \equiv (d_1(p), d_2(p), \ldots, d_k(p)),$$
$$R \equiv (r_1, r_2, \ldots, r_k),$$
$$c_\Lambda(p) \equiv c(p) + d_1(p)\lambda_1 + \cdots + d_k(p)\lambda_k, \quad \text{and}$$
$$d_\Lambda(p) \equiv d_1(p)\lambda_1 + \cdots + d_k(p)\lambda_k.$$

Here $c_\Lambda(p)$ and $d_\Lambda(p)$ are called the aggregated cost and the aggregated delay of path p, respectively. We shall use P_Λ to denote the set of $s-t$ paths attaining the minimum aggregated cost with respect to Λ. A path $p_\Lambda \in P_\Lambda$ is called a Λ-minimal path.

B. A STRONGLY POLYNOMIAL TIME APPROXIMATION ALGORITHM FOR QSR(1) PROBLEM

The key issue now is to search for the maximizing multiplier and determine termination conditions. If there is only one delay constraint, i.e., $k = 1$, we have the following claim from [9], [32] repeated below for ease of reference.

Theorem 1: Avalue $\lambda > 0$ maximizes the function $L(\lambda)$ if and only if there are paths p_c and p_d which are both c_λ-minimal and for which $d(p_c) \geq r$ and $d(p_d) \leq r$. (p_c and p_d can be the same. In this case $d(p_d) = d(p_c) = r$).

Theorem 2: DUAL-RELAX-QSR(1) is solvable in $O((m + n \log n)^2)$ time.

Proof: We prove this theorem by presenting an algorithm with $O((m + n \log n)^2)$ time complexity. The algorithm called PSQSR (parametric search based QoS routing) algorithm is based on a methodology first proposed by Megiddo [18] to solve fractional combinatorial optimization problems.

Step 1. $M_v = (x_v, y_v) = (+\infty, +\infty)$ for $v = 2, 3, \cdots, n$ and $M_1 = (0, 0)$
Step 2. $i \leftarrow 1$
Step 3. $u \leftarrow 1$
Step 4*. $\forall v, (u, v) \in E$, if $(x_v + \lambda^* y_v > x_u + \lambda^* y_u + c_{uv} + \lambda^* d_{uv})$ or $(x_v + \lambda^* y_v = x_u + \lambda^* y_u + c_{uv} + \lambda^* d_{uv})$ and $(x_v > x_u + c_{uv}))$
$M_v \leftarrow (x_u + c_{uv}, y_u + d_{uv})$
Step 5. $u \leftarrow u + 1$ and if $u \leq n$, go to Step 4.
Step 6. $i \leftarrow i + 1$ and if $i < n$, go to Step 3.

FIGURE 2. PSQSR algorithm for QSR(1) problem.

Assume node 1 is the source and node n is the target. In Figure 2, we present algorithm PSQSR for computing a QoS routing using lexicographic order on a pair of link weights $(l_{uv}, c_{uv}) \forall (u, v) \in E$ and based on parametric search, where $l_{uv} = c_{uv} + \lambda^* d_{uv}$ and λ^* is unknown. The algorithm is the same as the BFM (Bellman-Ford-Moore) algorithm except for Step 4 which needs special care. We use BFM algorithm here because it is easy to explain. Actually we use Dijkstra's algorithm for better time complexity results.

In Figure 2, we need extra steps (Oracle test) to evaluate the Boolean expression in the if statement in Step 4 since $\lambda^* \geq 0$ is unknown. If $x_v = \infty$, $y_v = \infty$, then the inequality holds. Assume x_v and y_v are finite (non-negative) values. Then it suffices to evaluate the following Boolean expression.

$$p + q\lambda^* \leq 0?, \quad \text{where} \quad p = x_u + c_{uv} - x_v \quad \text{and} \quad q = (y_u + d_{uv} - y_v).$$

Δ : Path delay constraint

Step 1. Let $\lambda = -p/q > 0$ for each link $(u, v) \in E$, define its length $l_{uv} = c_{uv} + \lambda d_{uv}$.
Step 2. Compute two shortest paths p_c and p_d using the lexicographic order on (l_{uv}, c_{uv}) and (l_{uv}, d_{uv}), respectively.
Step 3. Obviously, $d(p_c) \geq d(p_d)$. Only four cases are possible:
a) $d(p_c) > T$ and $d(p_d) > T$: By Theorem 1, $\lambda < \lambda^*$ and thus $p + q\lambda^* < 0$ if $q < 0$ and $p + q\lambda^* > 0$ otherwise.
b) $d(p_c) < T$ and $d(p_d) < T$: By Theorem 1, $\lambda > \lambda^*$ and thus $p + q\lambda^* > 0$ if $q < 0$ and $p + q\lambda^* < 0$ otherwise.
c) $d(p_c) > T$ and $d(p_d) < T$: By Theorem 1, $\lambda = \lambda^*$ and $p + q\lambda^* = 0$.
d) $d(p_c) = T$ or $d(p_d) = T$: By Lemma 1, this is impossible.

FIGURE 3. Oracle test algorithm.

If $p \cdot q \geq 0$, then it is trivial to evaluate the Boolean expression. Without loss of generality, assume $p \cdot q < 0$, i.e., $-p/q > 0$. The Oracle test algorithm is presented in Figure 3.

The time complexity of the Oracle test is $O(m + n \log n)$. On the other hand, we can revise the algorithm in Figure 2 using Dijkstra's algorithm and the resulting algorithm will have time complexity $O((m + n \log n)^2)$.

Next, we show how to compute the value of λ^* and $L(\lambda^*)$. The algorithm in Figure 2 computes a λ^*-minimal path p with minimal cost. Similarly, we can compute a λ^*-minimal path q with minimal delay. Then the value of λ^* is given by the following equation: $c(p) + \lambda^* d(p) = c(q) + \lambda^* d(q)$ and $L(\lambda^*) = c(p) + \lambda^*(d(p) - D)$, where D is the path delay constrain (here $k = 1$). Notice that $d(q) \neq d(p)$ is guaranteed by Theorem 1 and the transformation in Section III.A. ☐

Because PSQSR and LARAC algorithms are based on the same methodology and obtain the same solution, we shall also call our algorithm LARAC. In the rest of the paper, we shall discuss how to extend it for $k > 1$. In particular we develop an approach that combines the LARAC algorithm as a building block with certain techniques in mathematical programming. We shall call this new approach as GEN-LARAC.

C. GEN-LARAC FOR THE QSR(K) PROBLEM

1) OPTIMALITY CONDITIONS

Theorem 3: Given an instance of a feasible QSR(k) problem, a vector $\Lambda \geq 0$ maximizes $L(\Lambda)$ if and only iff the following problem in the variables u_j is feasible.

$$\sum_{p_j \in P_\Lambda} u_j \cdot d_i(p_j) = r_i, \quad \forall i, \lambda_i > 0 \qquad (12)$$

$$\sum_{p_j \in P_\Lambda} u_j \cdot d_i(p_j) \le r_i, \quad \forall i, \lambda_i = 0 \tag{13}$$

$$\sum_{p_j \in P_\Lambda} u_j = 1, \tag{14}$$

$$u_j \ge 0, \quad \forall p_j \in P_\Lambda \tag{15}$$

Proof:

Sufficiency: Let $x = (u_1, \dots, u_r, 0, 0, \dots)$ be a vector of size $|P_{st}|$, where $r = |P_\Lambda|$. Obviously, x is a feasible solution to RELAX-QSR(k). It suffices to show that x and Λ satisfy the complementary slackness conditions.

According to (7), $\forall p \in P_{st}, w \le c(p) + d_1(p)\lambda_1 + \dots + d_k(p)\lambda_k$. Since we need to maximize (6), the optimal $w = c(p_\Lambda) + d_1(p_\Lambda)\lambda_1 + \dots + d_k(p_\Lambda)\lambda_k, \forall p_\Lambda \in P_\Lambda$. For all other paths p, $w - c(p) + d_1(p)\lambda_1 + \dots + d_k(p)\lambda_k < 0$. Hence x satisfies the complementary slackness conditions. By (12) and (13), Λ also satisfies complementary slackness conditions.

Necessary: Let x^* and (w, Λ) be the optimal solution to RELAX-QSR(k) and DUAL-RELAX-QSR(k), respectively. It suffices to show that we can obtain a feasible solution to (12)-(15) from x^*.

We know that all the constraints in (7) corresponding to paths in $P_{st} - P_\Lambda$ are strict inequalities, and $w = c(p_\Lambda) + d_1(p_\Lambda)\lambda_1 + \dots + d_k(p_\Lambda)\lambda_k, \forall p_\Lambda \in P_\Lambda$. Therefore, from complementary slackness conditions we get $x_p = 0, \forall p \in P_{st} - P_\Lambda$.

Now let us set u_j corresponding to path p in P_Λ equal to x_p, and set all other u_j's corresponding to paths not in P_Λ equal to zero. The u_i's so elected will satisfy (12) and (13) since these are complementary conditions satisfied by (w, Λ). Since x_i's satisfy (3), u_j's satisfy (14). Thus we have identified a solution satisfying (12)-(15). □

2) GEN-LARAC: A COORDINATE ASCENT METHOD

Our approach for the QSR(k) problem is based on the coordinate ascent method in Figure 4 and proceeds as follows. Given a multiplier Λ, in each iteration we try to improve the value of $L(\Lambda)$ by updating one component of the multiplier vector. If the objective function is not differentiable, the coordinate ascent method may get stuck at a corner Λ_s not being able to make progress by only changing one component. We call Λ_s a pseudo optimal point which requires updates of at least two components to achieve improvement in the solution. We shall discuss later how to jump to a better solution from a pseudo optimal point. Our simulations show that the objective value attained at pseudo optimal points is usually very close to the maximum value of $L(\Lambda)$.

3) VERIFICATION OF OPTIMALITY OF Λ

In Step 3 we need to verify if a given Λ is optimal. We show that this can be accomplished by solving the following LP (Linear Programming) problem, where $P_\Lambda = \{p_1, p_2, \dots, p_r\}$ is the set of Λ-minimal paths.

$$\text{Maximize} \quad 0 \tag{16}$$

$$\text{subject to} \quad \sum_{p_j \in P_\Lambda} u_j \cdot d_i(p_j) = r_i, \quad \forall i, \lambda_i > 0 \tag{17}$$

```
Step 1: Λ^0 ← (0,0,...,0); t ← 0; flag ← true; B ← 0
Step 2: (Coordinate Ascent Steps)
        while (flag)
          flag ← false
          for i = 1 to k
            γ ← arg max_{ξ≥0} L(λ_1^t, ..., λ_{i-1}^t, ξ, λ_{i+1}^t, ..., λ_k^t)
            if (γ ≠ λ_i^t) then
              flag ← true
              λ_j^{t+1} = { γ      j = i, j = 1,2,...,k
                          { λ_j^t  j ≠ i
              t ← t + 1
            end if
          end for
        end while
Step 3. If Λ^t is optimal then return Λ^t.
Step 4. B ← B + 1 and go to Step 5 if B < B_max
        (B_max is the maximum number of iteration allowed);
        Otherwise, stop.
Step 5. Compute a new vector Λ^+ such that L(Λ^+) > L(Λ^t).
Step 6. t ← t + 1, Λ^t ← Λ^+, and go to Step 2.
```

FIGURE 4. GEN-LARAC: A coordinate ascent algorithm for the QSR(k) problem.

$$\sum_{p_j \in P_\Lambda} u_j \cdot d_i(p_j) \le r_i, \quad \forall i, \lambda_i = 0 \tag{18}$$

$$\sum_{p_j \in P_\Lambda} u_j = 1, \tag{19}$$

$$u_j \ge 0, \quad \forall p_j \in P_\Lambda \tag{20}$$

By Theorem 3, if the above linear program is feasible then the multiplier Λ is a maximizing multiplier.

Let $(y_1, \dots, y_k, \delta)$ be the dual variables corresponding to the above problem. Let $Y = (y_1, y_2, \dots, y_k)$. The dual of (17)-(20) is as follows

$$\text{Minimize} \quad RY^T + \delta \tag{21}$$

$$\text{subject to} \quad D(p_i)Y^T + \delta \ge 0, \quad i = 1, 2, \dots, r \tag{22}$$

$$y_i \ge 0, \quad \forall i, \quad \lambda_i > 0 \tag{23}$$

Evidently the LP problem (21)-(23) is feasible. From the relationship between primal and dual problems, it follows that if the linear program (16)-(20) is infeasible, then the objective of (21) is unbounded ($-\infty$). Thus, if the optimum objective of (21)-(23) is 0, then the linear program (16)-(20) is feasible and by Theorem 3 the corresponding multiplier Λ is optimal. In summary, we have the following lemma.

Lemma 3: If (16)-(20) is infeasible, then $\exists Y = (y_1, y_2, \dots, y_k)$ and δ satisfying (22)-(23) and $RY^T + \delta < 0$.

The Y in Lemma 3 can be identified by applying any LP solver on (21)-(23) and terminating it once the current objective value becomes negative.

Let Λ be a non-optimal Lagrangian multiplier and $\Lambda(s, Y) = \Lambda + Y/s$ for $s > 0$.

Theorem 4: If a multiplier $\Lambda \geq 0$ is not optimal, then

$$\exists M > 0, \quad \forall s > M, \quad L(\Lambda(s, Y)) > L(\Lambda).$$

Proof: If M is big enough, $P_\Lambda \cap P_{\Lambda(s,Y)} \neq \phi$. Let $p_j \in P_\Lambda \cap P_{\Lambda(s,V)}$.

$$\begin{aligned} L(\Lambda(s, Y)) &= c(p_j) + (D(p_j) - R)(\Lambda + Y/s)^T \\ &= c(p_j) + (D(p_j) - R)\Lambda^T + (D(p_j) - R)(Y/s)^T \\ &= L(\Lambda) + (D(p_j)Y^T - RY^T)/s. \end{aligned}$$

Since $D(p_j)Y^T + \delta \geq 0$ and $RY^T + \delta < 0, D(p_j)Y^T - RY^T > 0$.

Hence $L(\Lambda(s, Y)) > L(\Lambda)$. □

We can find the proper value of M by binary search after computing Y. The last issue is to compute P_Λ. It can be expected that the size of P_Λ is usually very small. In our experiments, $|P_\Lambda|$ never exceeded 4 even for large and dense networks. The k-shortest path algorithm can be adapted easily to computing P_Λ.

D. ANALYSIS OF THE ALGORITHM

In this section, we shall discuss the convergence properties of GEN-LARAC.

Lemma 4: If there is a strictly feasible path, then for any given τ, the set $S_\tau = \{\Lambda | L(\Lambda) \geq \tau\} \subset R^k$ is bounded.

Proof: Let p^* be a strictly feasible path. For any $\Lambda = (\lambda_1, \ldots, \lambda_k) \in S_\tau$, we have

$$c(p^*) + \lambda_1(d_1(p^*) - r_1) + \cdots + \lambda_k(d_k(p^*) - r_k) \geq L(\Lambda) \geq \tau.$$

Since $d_i(p^*) - r_i < 0$ and $\lambda_i \geq 0$ for $i = 1, 2..., k$, Λ must be bounded. □

By Theorem 1, we have the following lemma.

Lemma 5: A multiplier $\Lambda \geq 0$ is pseudo optimal if $\forall i\ \exists p_c^i, p_d^i \in P_\Lambda, d_i(p_c^i) \geq r_i$ and $d_i(p_d^i) \leq r_i$

For an n-vector $V = (v_1, v_2, \ldots, v_n)$, let $|V|_1 = |v_1| + |v_2| + \cdots + |v_n|$ denote the L^1-norm.

Lemma 6: Let Λ and H be two multipliers obtained in the same while-loop in Step 2 in Figure 4. Then $|L(H) - L(\Lambda)| \geq |H - \Lambda|_1$.

Proof: Let $\Lambda = \Lambda^1, \Lambda^2, \ldots, \Lambda^j = H$ be the consecutive sequence of multipliers obtained in Step 2. We first show that $|L(\Lambda^{i+1}) - L(\Lambda^i)| \geq |\Lambda^{i+1} - \Lambda^i|_1$.

Consider two cases.

Case 1: $\lambda_b^{i+1} > \lambda_b^i$.

By Theorem 1 and Lemma 1,

$$\exists p_{\Lambda^{i+1}} \text{ and } d_b(p_{\Lambda^{i+1}}) > r_b.$$

By definition, we have:

$$c(p_{\Lambda^{i+1}}) + \Lambda^{i+1} D^T(p_{\Lambda^{i+1}}) \leq c(p_{\Lambda^i}) + \Lambda^{i+1} D^T(p_{\Lambda^i}),$$

and

$$c(p_{\Lambda^{i+1}}) + \Lambda^i D^T(p_{\Lambda^{i+1}}) \geq c(p_{\Lambda^i}) + \Lambda^i D^T(p_{\Lambda^i}),$$

Then

$$\begin{aligned} &L(\Lambda^{i+1}) - L(\Lambda^i) \\ &= c(p_{\Lambda^{i+1}}) + \Lambda^{i+1}(D(p_{\Lambda^{i+1}}) - R)^T - [c(p_{\Lambda^i}) \\ &\quad + \Lambda^i(D(p_{\Lambda^i}) - R)^T] \\ &= c(p_{\Lambda^{i+1}}) + \Lambda^i(D(p_{\Lambda^{i+1}}) - R)^T + (\Lambda^{i+1} - \Lambda^i) \\ &\quad \times (D(p_{\Lambda^{i+1}}) - R)^T - [c(p_{\Lambda^i}) + \Lambda^i(D(p_{\Lambda^i}) - R)^T] \\ &\geq c(p_{\Lambda^i}) + \Lambda^i(D(p_{\Lambda^i}) - R)^T + (\Lambda^{i+1} - \Lambda^i) \\ &\quad \times (D(p_{\Lambda^{i+1}}) - R)^T - [c(p_{\Lambda^i}) + \Lambda^i(D(p_{\Lambda^i}) - R)^T] \\ &= (\Lambda^{i+1} - \Lambda^i)(D(p_{\Lambda^{i+1}}) - R)^T \\ &= (\lambda_b^{i+1} - \lambda_b^i)(d(p_{\Lambda^{i+1}}) - r_b) \\ &\geq |\lambda_b^{i+1} - \lambda_b^i|. \end{aligned}$$

Case 2: $\lambda_b^{i+1} < \lambda_b^i$.

By Theorem 1 and Lemma 1,

$$\exists p_{\Lambda^{i+1}} \quad \text{and} \quad d_b(p_{\Lambda^{i+1}}) < r_b.$$

The rest of the proof is similar to Case 1.

Hence

$$\begin{aligned} &|L(\Lambda^j) - L(\Lambda^1)| \\ &= |L(\Lambda^2) - L(\Lambda^1) + L(\Lambda^3) - L(\Lambda^2) \\ &\quad + \cdots + L(\Lambda^j) - L(\Lambda^{j-1})| \\ &= |L(\Lambda^2) - L(\Lambda^1)| + |L(\Lambda^3) - L(\Lambda^2)| \\ &\quad + \cdots + |L(\Lambda^j) - L(\Lambda^{j-1})| \\ &\geq |\Lambda^2 - \Lambda^1|_1 + |\Lambda^3 - \Lambda^2|_1 + \cdots + |\Lambda^j - \Lambda^{j-1}|_1 \\ &\geq |\Lambda^j - \Lambda^1|_1. \end{aligned}$$

The second equality holds because $L(\Lambda^1) < L(\Lambda^2) < \cdots < L(\Lambda^j)$. □

Obviously, if the while-loop in Step 2 in Figure 4 terminates in a finite number of steps, the multiplier is pseudo optimal by definition. If the while loop does not terminate in a finite number of steps (this occurs only when infinite machine precision is assumed, in practice, GEN-LARAC terminates in finite steps), we have the following theorem.

Theorem 5: Let $\{\Lambda^i\}$ be a consecutive sequence of multipliers generated in the same while-loop in Step 2 in Figure 4. Then the limit point of $\{L(\Lambda^i)\}$ is pseudo optimal.

Proof: Since $L(\Lambda^1) < L(\Lambda^2) < \cdots$ and $\{\Lambda^i\}$ is bounded from above, $\lim s_{s\to\infty} L(\Lambda^s)$ exists and is denoted as L^*. We next show the vector $\lim s_{s\to\infty} \Lambda^s$ also exists.

By Lemma 6, $\forall s, j > 0$,

$$|\Lambda^{s+j} - \Lambda^s|_1 \leq |L(\Lambda^{s+j}) - L(\Lambda^s)|$$

By Cauchy criterion, $\lim s_{s\to\infty} \Lambda^s$ exists. We denote it as Λ^*.

We label all the paths in P_{st} as $p_1, p_2, \ldots, p_N$ such that $c_{\Lambda^*}(p_1) \leq c_{\Lambda^*}(p_2), \ldots, c_{\Lambda^*}(p_N)$. Obviously p_1 is a Λ^*-minimal path.

Let

$$\theta = \min \{c_{\Lambda^*}(p_j) - c_{\Lambda^*}(p_i) | \forall p_i, \quad p_j \in P_{st}, \quad c_{\Lambda^*}(p_j) - c_{\Lambda^*}(p_i) > 0\},$$

and

$$\pi = \max \{d_w(p_j) - d_w(p_i) | \forall p_i, \ p_j \in P_{st}, \ w = 1, 2, \ldots, k\}.$$

IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING

Let M be a large number, such that $\forall\, t \geq M$, $|\Lambda^* - \Lambda^t|_1 < \theta/(2\pi)$.

Consider any component $j \in \{1, 2, \ldots, k\}$, of the multiplier after computing

$$\arg\max_{\xi \geq 0} L(\lambda_1, \ldots, \lambda_{j-1}, \xi, \lambda_{j+1}, \ldots, \lambda_k)$$

in iteration $t \geq M$.

By Theorem 1,

$$\exists p_c^t \quad \text{and} \quad p_d^t \in P_\Lambda^t \quad \text{and} \quad d_j(p_c^t) \geq l_j \geq d_j(p_d^t).$$

It suffices to show $P_{\Lambda^t} \subseteq P_{\Lambda^*}$.

Given $p_{\Lambda^t} \in P_{\Lambda^t}$, we shall show

$$c_{\Lambda^*}(p_{\Lambda^t}) = c_{\Lambda^*}(p_1).$$

We have

$$\begin{aligned}
0 &\leq c_{\Lambda^t}(p_1) - c_{\Lambda^t}(p_{\Lambda^t}) \\
&= [c(p_1) + d_{\Lambda^t}(p_1)] - [c(p_{\Lambda^t}) + d_{\Lambda^t}(p_{\Lambda^t})] \\
&= c(p_1) + d_{\Lambda^*}(p_1) - [c(p_{\Lambda^t}) + d_{\Lambda^*}(p_{\Lambda^t})] \\
&\quad + (\Lambda^t - \Lambda^*)(D(p_1) - D(p_{\Lambda^t}))^T \\
&= c_{\Lambda^*}(p_1) - c_{\Lambda^*}(p_{\Lambda^t}) + (\Lambda^t - \Lambda^*)(D(p_1) \\
&\quad - D(p_{\Lambda^t}))^T
\end{aligned}$$

Since

$$|(\Lambda^t - \Lambda^*)(D(p_i) - D(p_j))^T| \leq \pi|(\Lambda^t - \Lambda^*)|_1 \leq \theta/2,$$
$$0 \leq c_{\Lambda^t}(p_1) - c_{\Lambda^t}(p_{\Lambda^t}) \leq c_{\Lambda^*}(p_1) - c_{\Lambda^*}(p_{\Lambda^t}) + \theta/2.$$

Then

$$c_{\Lambda^*}(p_{\Lambda^t}) - c_{\Lambda^*}(p_1) \leq \theta/2,$$

which implies that $c_{\Lambda^*}(p_{\Lambda^t}) = c_{\Lambda^*}(p_1)$.

Hence,

$$\forall\, j \in \{1, 2, \ldots, k\}, \exists p_c^*, p_d^* \in P_{\Lambda^*}, d_j(p_c^*) \geq l_j \geq d_j(p_d^*).$$

□

E. SIMULATION

We use COPT, OPT, and POPT to denote the cost of the optimal path to the QSR(k) problem, the optimal value, and the pseudo optimal value of the Lagrangian function, respectively. In our simulation, we first verify that the objectives at pseudo optimal points are very close to the optimal objectives. We use 3 types of graphs: Power-law out-degree graph (PLO) [23], Waxman's random graph (RAN) [31], and regular graph (REG) [27]. The number of weights chosen are 4, 8 and 12, i.e., $k = 4$, 8, and 12. Link weights are random even integers uniformly distributed between 2 and 200. To decide proper delay bounds, we randomly choose an $s-t$ path p_{ran} and set $r_i = (1 + \epsilon)d_i(p_{ran})$ where ϵ is a random variable evenly distributed in interval $[-0.25, 0.25]$. For each type of topology, 10 different graphs with valid delay bounds and random source and target nodes are generated. The results reported are averaged over the 10 instances.

TABLE 1. Quality of pseudo-optimal paths.

Error: (OPT − POPT) / OPT   #SP: Number of computation of shortest path

Type	k	OPT	POPT	Error	$g(p)$	$f(p)$	#SP	Time(s)
REG	4	1116.8	1109.9	0.006	1.01	1.07	15.2	0.14
REG	8	1078.3	1069.0	0.032	1.00	1.09	20.7	0.21
REG	12	1066.2	1057.8	0.008	1.00	1.08	28.2	0.32
PLO	4	401.75	382.71	0.047	1.00	1.25	9.6	0.07
PLO	8	328.95	320.77	0.025	1.01	1.15	7.2	0.04
PLO	12	368.43	342.43	0.071	1.02	1.24	18.3	0.21
RAN	4	1543.6	1531.5	0.008	1.01	1.08	13.4	0.13
RAN	8	1473.3	1456.5	0.011	1.00	1.09	20.0	0.25
RAN	12	1438.6	1423.7	0.010	1.00	1.01	17.9	0.45

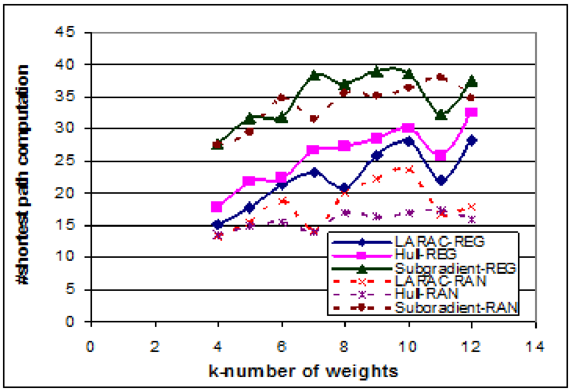


FIGURE 5. Comparision of GEN-LARAC, Hull approach, and subgradient method.

We use the following two metrics to measure the quality of path p in Table 1.

$$g(p) = c(p)/POPT \quad \text{and} \quad f(p) = \max_{i=1,2,\ldots,k} d_i(p)/r_i,$$

where $g(p)$ is the upper bound of the gap between the cost of p and $COPT$. The $f(p)$ indicates the degree of violation of p to the constraints on its delays.

In Figure 5, the label LARAC-REG means the results obtained by running GEN-LARAC algorithm on regular graphs. Other labels can be interpreted similarly. We only report the results on regular graphs and random graphs for better visibility.

We conducted extensive experiments to compare our algorithm with the Hull approach [19], the subgradient method [1], and the general-purpose LP solver CPLEX. Because the four approaches share the same objective, i.e., maximizing the Lagrangian function, they always obtain similar results. We only report the number of shortest path computations which dominate the running time of all the first three algorithms. All algorithms terminate when they have reached 99% of the OPT. Generally, GEN-LARAC algorithm and Hull approach are faster than the subgradient methods and CPLEX (See [41] for the comparison of Hull approach and CPLEX). But GEN-LARAC and Hull approach beat each other on different graphs. Figure 5 shows that on the regular graph, GEN-LARAC is the fastest. But for the random and Power-law out degree graphs, the Hull approach is the fastest. The probable reason is that the number of $s-t$ paths is relatively small in these two types of graphs because the length (number of hops) of $s-t$ paths is short even when the number of nodes is large. This will bias the results in favor of

Hull approach which adds one $s-t$ path into the linear system in each iteration [19]. We choose the regular graph because we have a better control of the length of $s-t$ paths.

IV. SUMMARY AND CONCLUSION

In this paper we developed a new approach to the QoS routing problem involving multiple additive constraints. Our approach uses the LARAC algorithm as a building block and combines it with certain ideas from mathematical programming to design a method that progressively improves the value of the Lagrangian function until optimum is reached. The algorithm is analyzed and its convergence property has been established. Simulation results comparing our approach with two other approaches show that the new approach is quite competitive.

Since the LARAC algorithm is applicable for the general class of optimization problems (involving one additive delay constraint) studied in [3] our approach can also be extended for this class of problems whenever an algorithm for the underlying optimization problem (such as Dijkstra's algorithm for the shortest path problem) is available. In other words, besides making an important contribution to the QoS routing area, this paper contains a significant advance to the general area of combinatorial optimization.

REFERENCES

[1] J. E. Beasley and N. Christofides, "An algorithm for the resource constrained shortest path problem," *Networks*, vol. 19, no. 4, pp. 379–394, 1989.

[2] Y. Bejerano, Y. Breitbart, A. Orda, R. Rastogi, and A. Sprintson, "Algorithms for computing QoS paths with restoration," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 648–661, Jun. 2005.

[3] D. Blokh and G. Gutin, "An approximate algorithm for combinatorial optimization problems with two parameters," *Austral. J. Combinat.*, vol. 14, pp. 157–164, 1996.

[4] A. Chakrabarti and G. Manimaran, "Reliability constrained routing in QoS networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 662–675, Jun. 2005.

[5] M. R. Garey and D. S. Johnson, *Command Intractability*. San Francisco, CA, USA: Freeman, 1979.

[6] G. Y. Handler and I. Zang, "A dual algorithm for the constrained shortest path problem," *Networks*, vol. 10, no. 4, pp. 293–310, 1980.

[7] R. Hassin, "Approximation schemes for the restricted shortest path problem," *Math. Oper. Res.*, vol. 17, no. 1, pp. 36–42, 1992.

[8] J. M. Jaffe, "Algorithms for finding paths with multiple constraints," *Networks*, vol. 14, no. 1, pp. 95–116, 1984.

[9] A. Jüttner, B. Szviatovski, I. Mécs, and Z. Rajkó, "Lagrange relaxation based method for the QoS routing problem," in *Proc. IEEE INFOCOM*, Anchorage, AK, USA, Apr. 2001, pp. 859–868.

[10] A. Jüttner, "On resource constrained optimization problems," in *Proc. 4th Jpn.-Hungarian Symp. Discrete Math. Appl.*, Budapest, Hungary, 2005.

[11] A. Jüttner, "On budgeted optimization problems," *SIAM J. Discrete Math.*, vol. 20, no. 4, pp. 880–892, 2006.

[12] A. Jüttner, "Optimization with additional variables and constraints," *Oper. Res. Lett.*, vol. 33, no. 3, pp. 305–311, May 2005.

[13] T. Korkmaz and M. Krunz, "Multi-constrained optimal path selection," in *Proc. IEEE INFOCOM*, vol. 2. Anchorage, AK, USA, Apr. 2001, pp. 834–843.

[14] T. Korkmaz and M. Krunz, "Bandwidth-delay constrained path selection under inaccurate state information," *IEEE/ACM Trans. Netw.*, vol. 11, no. 3, pp. 384–398, Jun. 2003.

[15] T. Korkmaz, M. Krunz, and S. Tragoudas, "An efficient algorithm for finding a path subject to two additive constraints," *Comput. Commun. J.* vol. 25, no. 3, pp. 225–238, 2002.

[16] F. Kuipers, T. Korkmaz, M. Krunz, and P. Van Mieghem, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Commun. Mag.*, vol. 40, no. 12, pp. 50–55, Dec. 2002.

[17] G. Luo, K. Huang, C. Hobbs, and E. Munter, "Multi-QoS constraints based routing for IP and ATM networks," in *Proc. IEEE Workshop QoS Support Real Time Internet Appl.*, Vancouver, BC, Canada, Jun. 1999.

[18] N. Megiddo, "Combinatorial optimization with rational objective functions," *Math. Oper. Res.*, vol. 4, no. 4, pp. 414–424, Nov. 1979.

[19] K. Mehlhorn and M. Ziegelmann, "Resource constrained shortest paths," in *Proc. 8th Eur. Symp. Algorithms (ESA)*, 2000, pp. 326–337.

[20] P. Van Mieghem, H. De Neve, and F. Kuipers, "Hop-by-hop quality of service routing," *Comput. Netw.*, vol. 37, nos. 3–4, pp. 407–423, 2001.

[21] P. Van Mieghem and F. A. Kuipers, "Concepts of exact QoS routing algorithms," *IEEE/ACM Trans. Netw.*, vol. 12, no. 5, pp. 851–864, Oct. 2004.

[22] H. De Neve and P. Van Mieghem, "TAMCRA: A tunable accuracy multiple constraints routing algorithm," *Comput. Commun.*, vol. 23, no. 7, pp. 667–679, 2000.

[23] C. R. Palmer and J. G. Steffan, "Generating network topologies that obey power laws," in *Proc. IEEE GLOBECOM*, San Francisco, CA, USA, Dec. 2000, pp. 434–438.

[24] T. Radzik, "Fractional combinatorial optimization," in *Handbook of Combinatorial Optimization*, vol. 1, D.-Z. Du and P. M. Pardalos, Eds. Norwell, MA, USA: Kluwer, Dec. 1998.

[25] R. Ravindran, K. Thulasiraman, A. Das, K. Huang, G. Luo, and G. Xue, "Quality of service routing: Heuristics and approximation schemes with a comparative evaluation," in *Proc. IEEE Int. Symp. Circuits Syst.*, Scottsdale, AZ, USA, May 2002, pp. 775–778.

[26] J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 541–550, Aug. 2002.

[27] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*. New York, NY, USA: Wiley, 1992.

[28] A. Orda and A. Sprintson, "Efficient algorithms for computing disjoint QoS paths," in *Proc. 23rd Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Hong Kong, Mar. 2004.

[29] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, Sep. 1996.

[30] A. Warburton, "Approximation of Pareto optima in multiple-objective, shortest-path problems," *Oper. Res.*, vol. 35, no. 1, pp. 70–79, 1987.

[31] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[32] Y. Xiao, K. Thulasiraman, G. Xue, and A. Jüttner, "The constrained shortest path problem: Algorithmic approaches and an algebraic study with generalization," *AKCE Int. J. Graphs Combinat.*, vol. 2, no. 2, pp. 63–86, Dec. 2005.

[33] Y. Xiao, K. Thulasiraman, and G. Xue, "QoS routing in communication networks: Approximation algorithms based on the primal simplex method of linear programming," *IEEE Trans. Comput.*, vol. 55, no. 7, pp. 815–829, Jul. 2006.

[34] Y. Xiao, K. Thulasiraman, X. Fang, D. Yang, and G. Xue, "Computing a most probable delay constrained path: NP-hardness and approximation schemes," *IEEE Trans. Comput.*, vol. 61, no. 5, pp. 738–744, May 2012.

[35] Y. Xiao, K. Thulasiraman, and G. Xue, "GEN-LARAC: A generalized approach to the constrained shortest path problem under multiple additive constraints," in *Proc. 16th Annu. Int. Symp. Algorithms Comput.*, Sanya, China, Dec. 2005, pp. 92–105.

[36] Y. Xiao, K. Thulasiraman, and G. Xue, "Constrained shortest link-disjoint paths selection: A network programming based approach," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 5, pp. 1174–1187.

[37] G. Xue, A. Sen, W. Zhang, J. Tang, and K. Thulasiraman, "Finding a path subject to many additive QoS constraints," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 201–211, Feb. 2007.

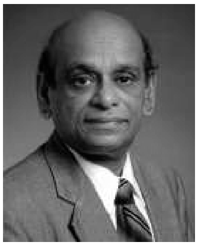
[38] G. Xue, W. Zhang, J. Tang, and K. Thulasiraman, "Polynomial time approximation algorithms for multi-constrained QoS routing," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 656–669.

[39] G. Xue, "Minimum-cost QoS multicast and unicast routing in communication networks," *IEEE Trans. Commun.*, vol. 51, no. 5, pp. 817–827, May 2003.

[40] X. Yuan, "Heuristic algorithms for multiconstrained quality-of-service routing," *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 244–256, Apr. 2003.

[41] M. Ziegelmann, "Constrained shortest paths and related problems," Ph.D. dissertation, Max-Planck-Inst. Inf., Saarbrücken, Germany, 2001.

YING XIAO received the Ph.D. degree in computer science from the University of Oklahoma, Norman, USA, in 2005. His research interests include graph theory, combinatorial optimization, distributed systems and networks, statistical inference, and machine learning.



KRISHNAIYAN THULASIRAMAN (M'72–SM'84–F'90) received the bachelor's and master's degrees in electrical engineering from the University of Madras, in 1963 and 1965, respectively, and the Ph.D. degree from IIT Madras, India, in 1968. He was on the Faculty of the Electrical Engineering and Computer Science Departments of IITM from 1965 to 1981. He was a Professor and the Chair of the Electrical and Computer Engineering Department with Concordia University, Montreal, Canada, from 1981 to 1994 and 1993 to 1994. Since 1994, he has been the Hitachi Chair and Professor with the School of Computer Science, University of Oklahoma, Norman. He has authored over 100 papers in archival journals. He has co-authored the books entitled *Graphs, Networks, and Algorithms* (New York: Wiley, 1981) and *Graphs: Theory and Algorithms* (New York: Wiley, 1992) with M. N. S. Swamy, and several book chapters. He holds a U.S. patent in distributed quality of service routing. His research interests have been in graph theory, combinatorial optimization, algorithms and applications in a variety of areas in computer science and electrical engineering, in particular, fault diagnosis, tolerance, and testing. His most recent interest is in the emerging area of network science and engineering. He has received several awards and honors, such as the Distinguished Alumnus Award of IIT Madras, the IEEE Circuits and Systems Society Technical Achievement Award, and the IEEE CAS Society Golden Jubilee Medal. He was a Senior Fellow of the Japan Society for Promotion of Science, the Gopalakrishnan Endowed Chair Professor of IIT Madras, and a fellow of the American Association for the Advancement of Science and the European Academy of Sciences.

GUOLIANG XUE (M'98–SM'99–F'11) received the B.S. degree in mathematics and the M.S. degree in operations research from Qufu Normal University, Qufu, China, in 1981 and 1984, respectively, and the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, MN, USA, in 1991. He is a Professor of Computer Science and Engineering with Arizona State University, Tempe, AZ, USA. His research interests include quality of service provisioning, resource allocation, survivability, security, and privacy issues in networks. He received best paper awards at the IEEE Globecom'2007, the IEEE ICC'2011, the IEEE MASS'2011, the IEEE Globecom'2011, and the IEEE ICC'2012, and the Best Paper Runner-Up Award at the IEEE ICNP'2010. He is an Associate Editor of the IEEE TRANSACTIONS ON NETWORKING, *ACM Transactions on Networking*, and the IEEE NETWORK, and a member of the Standing Committee of the IEEE INFOCOM. He was the General Co-Chair of the IEEE Conference on Communications and Network Security in 2014. He was a Keynote Speaker at the IEEE LCN'2011, the TCP Co-Chair of the IEEE INFOCOM'2010, and a Co-Organizer of the ARO Workshop on Human Centric Computing with Collective Intelligence-Challenges and Research Directions in 2012. He served on the Editorial Boards of *Computer Networks*, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He was a Distinguished Lecturer of the IEEE Communications Society.

MAMTA YADAV (S'13) received the bachelor's degree in computer science from the Institute of Engineering and Technology, Rajasthan University, Rajasthan, India, in 2005, and the master's degree in computer science from Oklahoma City University, OK, USA, in 2008. She is currently pursuing the Ph.D. degree in computer science with the University of Oklahoma, Norman, USA. Her research interests lie in the broad domain of topological study of complex networks, graph theory, and network science and engineering, in particular, community detection in complex networks using the topological and structural analysis approaches.