On Allocating Capacity in Networks with Path Length Constrained Routing

Murali Kodialam   T. V. Lakshman

Bell Laboratories
Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733, USA
{muralik, lakshman}@bell-labs.com

Abstract

In this paper, we consider capacity allocation in networks where traffic has to be routed along paths with bounded lengths. Routing with path length constraints is a basic problem in networking. These problems arise when connections have to satisfy delay, jitter, reliability or signal quality constraints. Despite its practical importance, these problems have not been adequately addressed in literature. In this paper, we focus on the simplest path length constrained flow problem, which is the Length Constrained Maximum Flow Problem (LCMF). We develop a Fully Polynomial Time Approximation Scheme (FPTAS) for this problem based on a primal-dual approach. Solutions to LCMF can be used to derive bounds to the Bounded Length Edge Disjoint Path (BLEDP) problem that has been shown to be hard to approximate. This approach can also be extended to solve multicommodity flow problems [9].

1 Introduction

In this paper, we consider capacity allocation issues that arise in networks where some of the traffic has to be routed on paths with specified bounds on path lengths. The need for such routing arises in both packet-based and optical networks. For capacity allocation and provisioning purposes, it is necessary to determine the amount of traffic that can be routed between sources and destinations in a given network. For the case without path-length constraints, there is extensive literature on capacity allocation and routing [1]. Generic maximum flow problems and multicommodity flow problems can be used to compute possible traffic flows when there are no path length restrictions. However, there are no good methods to solve these problems once path-length restrictions are introduced, and this is the focus of this paper.

For the problem considered, we assign a length metric to each link in the network. We assume that the length metric is additive, i.e., the length of a path is the sum of length of the links in the path. In optical networks, setting the length of the link to the physical length of the link captures the distance traveled by the signal. Setting all the link lengths to unity models the number of hops in the path. It is also easy to model multiplicative metrics, like signal degradation or packet loss probability, on a link by taking logarithms and converting them to additive metrics. We believe that the model and the algorithms developed in this paper are applicable to capacity allocation for a range of networking problems that arise in practice, and to our knowledge no satisfactory prior solutions exist. We now review the areas where routing with path length bounds arises naturally.

Path Length Restrictions in Optical Networks

Path length bounds are an important constraint for wavelength routing in optical networks [13]. This is because the optical signal attenuates as it traverses fiber links and this needs to be considered in the routing. Signal attenuation is a function of the type of fiber as well as the wavelength of the optical signal. As the signal traverses a link, it is amplified periodically using an In-Line Optical Amplifier. This is also referred to as the 1R (re-amplification) function. This amplifier boosts the signal as well as the noise. After the signal traverses longer distances, one has to regenerate the signal using an Optical-Electrical-Optical regeneration device. This is referred to as the 3R (retime, reshape, re-amplify) function. The regenerators are among the most expensive components in an optical network and in order to minimize the amount of regeneration, it is desirable to keep the length of the path traversed by the optical signal reasonably small. This can be modeled as a path length constraint for the paths traversed by the signal. In this paper, we abstract the problem in optical networks as a path length constraint. In a companion paper, we use the techniques developed here to model capacity allocation taking into consideration the location of the amplifiers and regenerators in an optical network.

Path Length Restrictions in Packet Networks

Path length restrictions in packet networks usually arise as restrictions on the number of hops traversed by the packet. Every time a router processes a packet, it introduces delays and jitter. As the length of the path increases, the delay as well as the delay jitter increases. For instance, in rate proportional processor sharing networks the worst case session i end-to-end delay ${D_i}^*$ is bounded as follows [11]:

$$ {D_i}^* \leq \frac{\sigma_i + 2(H-1)L_i}{\rho_i} + \sum_{m=1}^{H} \frac{L_{max}}{r^m}. $$

Here, session i traffic is characterized by its leaky bucket parameters (σ_i, ρ_i). H is the number of hops in the path, L_i is the maximum packet size for session i packets, L_{max} is the maximum allowed packet size for all sessions, and r^m is the capacity of the m^{th} link on the path. The key point to note is that both terms in the delay bound increase with the number of hops H in the path. Delay and jitter issues have also recently received attention in the context of the EF traffic class in the DiffServ framework. A bound of

$$ \frac{H}{1-(H-1)\alpha}(\max_m \frac{L_{max}}{r^m} + \tau) $$

for end-to-end delay jitter for EF traffic is derived in [2]. Here, τ is a delay factor that is dependent on the traffic parameters, and α is the maximum utilization of any link in the path. Therefore jitter bounds impose a hop count bound that is dependent on the traffic parameters.
Another application of restricted path length routing is for the case where link error rates are known and one wants paths to satisfy certain error rate constraints. An example case is that of TCP flows for which the achieved throughput is inversely proportional to the square root of the path loss probability. Hence, it would be desirable to route such flows so as to stay within a loss budget.

A related problem to routing with path length constraints is the problem of routing with link preferences taken into account. This can be modeled by associating an auxiliary preference "cost" for each link. The problem then is to route on paths that not only meet specified length constraints but also meet constraints on the preference cost.

Algorithmic Aspects

Apart from the networking applications, the problems considered are of independent theoretical interest. The problems we consider generalize the maximum flow and multicommodity flow problems which are fundamental problems in network optimization. Unlike the maximum flow and multicommodity flow problems for which there are polynomially bounded linear programs, flow problems along restricted paths do not seem to have polynomially bounded linear programming representation. We develop primal-dual type algorithms to generate provably close approximate solutions to the problem.

The algorithms that we develop are fully polynomial time approximation schemes (FPTAS). The idea in FPTAS is to obtain an ϵ optimal solution to the problem. An ϵ optimal solution to the maximizing problems that we consider in this paper is a solution to the problem that has a value at least $(1-\epsilon)$ times the optimal solution. An FPTAS is a family of algorithms that finds an ϵ-optimal solution in time that is a polynomial function of the problem parameters and $\frac{1}{\epsilon}$. The problem parameters in our case are the number of nodes in the graph n, the number of arcs (links) in the network m. This work extends the multicommodity flow work of Shahrokhi and Matula [14], Garg and Könemann [4] to the case where flows can be restricted to be along paths of bounded length. (See [4] for a review of the multicommodity flow literature related to this problem). Unlike the existing literature where the dual problem is polynomially solvable, the dual problem in our case is NP-complete. We show that using an approximation algorithm to solve the dual can still be used to preserve the FPTAS for the bounded length flow problem.

2 Problem Definition

In this paper we consider the simplest length constrained flow problem which we term the Length Constrained Maximum Flow (LCMF) problem. This problem can be stated as follows:

Length Constrained Maximum flow (LCMF)

Input

A directed capacitated network $G = (V, E)$ where $u(e)$ represents the capacity of edge e. Associated with link e is a non-negative additive metric $l(e)$. Two distinguished nodes $s, t \in V$ and a non-negative real number L.

Output

The maximum amount of flow that can be routed from s to t without violating capacity constraints, such that no path on which flow is routed is longer than L.

Throughout this paper we use n to represent the number of nodes in the network and m to represent the number of links. In the case where there are no path length constraints, the maximum flow between s and t represents the total amount of flow that can be routed between nodes s and t. Further the max-flow min-cut theorem of Ford-Fulkerson and Elias-Feinstein-Shannon gives the set of links that determines the s-t capacity. We show that adding path length con-

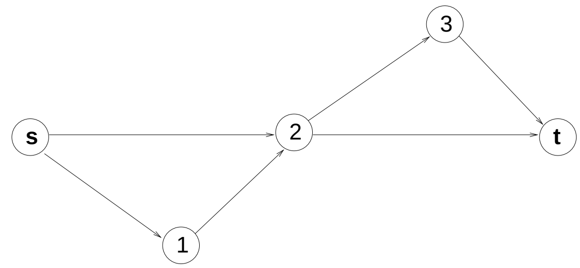

Figure 1: Flow Decomposition in LCMF: All link lengths = 1. Flow on each link = 1

straints makes the problem significantly more complex. One of the most important reasons bounded length flow problems have node been adequately studied is the absence of a natural flow formulation. We outline this and related problems below:

2.1 No Natural Flow Formulation

The main impediment to solving LCMF is the fact that there is no natural flow formulation for this problem. This is due to the fact that the usual flow balance constraint does not carry the identity of the flows. Therefore at some intermediate node in the network, it is not possible to know the length of the path traversed by the flow from the source node. Therefore, we use a path-arc formulation for this problem. (The path-arc formulation has an exponential number of columns.)

2.2 Flow Decomposition is not Easy

In the case of the the standard maximum flow problem, given the maximum flow, it is easy to decompose this max-flow into flow along paths from s to t. In general, this decomposition is not unique. In the case of LCMF, even if the maximum flow is given, it is not easy to decompose it into flow along paths. For example, consider the network shown in Figure 1. The flow on each edge is one unit and the weight of each edge is one. Therefore there is a flow of two units from s to t. If the flow is decomposed as $s-2-t$ with one unit of flow and $s-1-2-3-t$ with one unit of flow then the length of the first path is two and the second path is four. However, if the decomposition is $s-2-3-t$ with a flow of one unit and $s-1-2-t$ with a flow of one unit then the length of both the paths is three units. Therefore, for example, if the path length bound is three, the first decomposition will be infeasible but the second decomposition will be feasible. Therefore, it is impossible to just consider the flows to decide whether the flows satisfy the path length bound.

2.3 Non-Integrality of the LCMF Polyhedron

In the case of regular maxflow, if all the capacities are integral, then there exists a maximum flow that sends integral amount of flow on each link in the network. This is due to the fact that all the extreme points of the maxflow polyhedron are integral. This is not the case for LCMF. This is shown in Figure 2. All link capacities are assumed to be one unit and all links have a length of one unit. The maximum flow (without path length constraints) that can be sent from source to destination is 3 units. If we send the length bound to 4 units, then the solution to LMCF is 2.5 as shown in Table 1. Therefore even if all the capacities are integral, the optimal solution to LCMF can be non-integral.

Path	Flow
$s-2-4-6-t$	0.5
$s-1-5-t$	0.5
$s-3-2-6-t$	0.5
$s-2-6-7-t$	0.5
$s-1-4-5-t$	0.5

Table 1: Flow with Hop Constraint of 4: Total Flow = 2.5

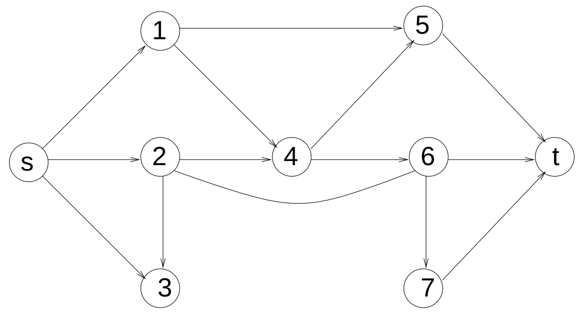

Figure 2: All Link Lengths =1 , All Link Capacities = 1 unit

In the next section, we outline two related problems to LCMF.

3 Some Related Problems

In this section, we outline two problems that are closely related to LCMF. Both these problems are NP-hard. The first is the Length Constrained Lightest Path (LCLP) problem. This can be viewed as the shortest path version of LCMF problem. The second is the Bounded Length Edge Disjoint Path (BLEDP) problem. This problem can be viewed as the integral version of LMCF.

3.1 Length Constrainted Lightest Path (LCLP) Problem

The Length Constrained Lightest Path Problem has been studied extensively in the literature under the names Constrainted Shortest Path Problem or the Restricted Shortest Path Problem. The LCLP problem is the following:

Length Constrained Lightest Path (LCLP)

Input
A directed network $G = (V, E)$ with two non-negative additive metrics associated with each link in the network, two nodes $s, t \in V$ and a non-negative real number L. Let $l(e)$ and $w(e)$ represent the length of link e and the weight of link e respectively.

Output
The minimum weight path in the network whose length does not exceed L.

LCLP arises naturally in dynamic routing problems where the link-length represents the delay or loss on the link and the weight of the link is the cost of routing a connection on the link. (We use the term weight instead of cost because the solution of this problem is used as a function in our algorithm and the weights represent the dual variables). The bound L represents some QoS guarantee that has to be met. It is known that LCLP is NP-complete [6]. There are some pseudo-polynomial time algorithms [8] and several fully polynomial approximation schemes [6] [10] to solve this problem. All these approximation schemes use dynamic programming with scaling. The current fastest algorithm is due to Lorenz and Raz [10] and it runs in time $O(mn \log n + \frac{mn}{\epsilon})$. The Lorenz-Raz algorithm obtains a path that satisfies the length constraint, that is at most $(1 + \epsilon)$ times the lightest path. If the length of all the links are the same, then without loss of generality we can assume that all the links have unit length. Then LCLP becomes a hop constrainted lightest path problem. It is well known [1], that the hop constrained lightest path problem can be solved in $O(nm)$ time using the Bellman-Ford algorithm.

3.2 Bounded Length Edge Disjoint Path (BLEDP) Problems

In this section, we consiser the maximum bounded length edge disjoint path problem.

Bounded Length Edge Disjoint Path Problem

Input
A directed network $G = (V, E)$, specified nodes s and t, and positive integers $J, L \leq n$.

Question
Does G contain J or more mutually edge disjoint paths from s to t, none involving more than L edges.

This problem is NP-hard for $J \geq 5$, polynomially solvable for $J \leq 3$ and open for $J = 4$. See [7] for the NP-hardness proof. The NP-hardness of this problem is a consequence of the non-integrality for LCMF. Also note that LCMF can be used to generate an upper bound for the maximum length-bounded disjoint path problem. It has been shown in [5] that it is NP-hard to approximate BLEDP to within a factor to $m^{\frac{1}{2}-\epsilon}$ for arbitrarily small ϵ. Note the LCMF gives an upper bound on BLEDP. For example in Figure 1, the maximum number of edge disjoint paths between s and t with less than 4 hops is two ($s-2-4-6-t$ and $s-1-5-t$ is one possible solution). LCMF between s and t with a hop bound of 4 gives a value of 2.5 as illustrated earlier.

4 Approximation Algorithm for LCMF Problem

We use the following notation in the linear programming formulation. The linear program is a path arc formulation where the variables are flows associated with paths in the network. Let $\mathcal{P}_L$ represent the set of paths between s and t whose length is not greater than L. Let P represent a generic path in $\mathcal{P}_L$. We associate flow variables $x(P)$ with path P. In other words, $x(P)$ represents the flow that is sent on path P. Since P is a collection of links, there will be a flow of $x(P)$ for each edge $e \in P$. The linear programming representation of CBPL is the following:

$$\max \sum_{P \in \mathcal{P}_L} x(P)$$

$$\begin{aligned} \sum_{P \in \mathcal{P}_L : P \ni e} x(P) &\leq u(e) \qquad \forall e \in E \\ x(P) &\geq 0 \qquad \forall P \in \mathcal{P}_L \end{aligned}$$

The first set of constraints restricts the flow on each link to be less than the capacity of the link. Note that the path length constraints are handled implicitly in the formulation. The dual to this problem assigns a weight to each edge of the graph. Let $w(e)$ represent the weight of edge $e \in E$. The weight $w(P)$ of a path $P \in \mathcal{P}_L$ is defined as the sum of the weights of the links in P.

$$\min \sum_{e \in E} u(e) w(e)$$

$$\begin{aligned} \sum_{e \in P} w(e) &\geq 1 \qquad \forall P \in \mathcal{P}_L \\ w(e) &\geq 0 \qquad \forall e \in E \end{aligned}$$

Given a vector of weights w, let $\alpha(w)$ represent the minimum weight path between s and t whose length is not greater than L. Therefore the dual can be equivalently written as

$$\min_{w \geq 0} \frac{\sum_e w(e) u(e)}{\alpha(w)}.$$

Let β represent the optimal dual solution value.

4.1 The Primal-Dual Algorithm

As in Garg and Könemann [4], the FPTAS for this problem starts off by setting $w(e) = \delta$ for all edges $e \in E$ to some precomputed value δ. While there is a path set $P \in \mathcal{P}_L$ such that $w(P)$ is less than 1, the algorithm augments flow along P. The amount of flow f_i sent in iteration i, is equal to the smallest capacity (not residual capacity) of any edge in P. Let this bottleneck link have a capacity u. The flow might violate the capacity constraint but this solution can be scaled finally to obtain a feasible solution to the primal. The weight of each edge in $e \in P$ is increased to $w(e) \leftarrow w(e) \left(1 + \frac{\epsilon u}{u(e)}\right)$. The weight of all the edges not in P is left unchanged.

Let $w_i(e)$ be the weight of arc e at the beginning of iteration i. This process of adding flows and adjusting arc weights is repeated, until for all $P \in \mathcal{P}_L$ the value of $w(P) > 1$. Let this happen in iteration t. In order to develop a FPTAS, at iteration i instead of picking some $P \in \mathcal{P}_L$ such that $w(P)$ is less than 1, we have to pick the min-weight path in $\mathcal{P}_L$. Note that determining the min-weight path in $\mathcal{P}_L$ is NP-hard. This is the key difference between the multicommodity flow problems solved in the literature and LCMF. Typically, the dual problem is a shortest path problem which is polynomially solvable. However, for LCMF the dual problem is the length constrained lightest path (LCLP) problem which is NP-hard. Since we cannot get the exact optimal solution, it affects the choice of the initial parameter δ as well as the running time analysis. However, we show that a FPTAS for the problem can be obtained even though the dual is solved approximately. In each step of the algorithm we determine the (approximate) length constrained lightest path using Lorenz-Raz algorithm. This algorithm gets to within $(1+\epsilon)$ of the optimal solution. If the bound is on the number of hops (and not length) then the Bellman-Ford algorithm can be used to get the optimal solution. In the rest of the analysis, we assume the general case of bounded length flows where the solution is within $(1+\epsilon)$ of the optimal solution. Let $D(i) = \sum_{e \in E} w_i(e)u(e)$. Let the vector of weights in iteration i be represented by w_i and the corresponding LCLP be represented by P^i. Then

$$\begin{aligned} D(i) &\leq \sum_e w_{i-1}(e)u(e) + \epsilon(1+\epsilon) \sum_{e \in P^i} w_{i-1}(e)u \\ &= D(i-1) + \epsilon(1+\epsilon)\,(f_i - f_{i-1})\,\alpha(i-1) \\ &\leq D(i-1) + \epsilon(1+\epsilon)\,(f_i - f_{i-1})\,\alpha(i-1) \end{aligned}$$

Therefore

$$D(i) \leq D(0) + \epsilon(1+\epsilon) \sum_{j=1}^{i} (f_j - f_{j-1})\,\alpha(j-1).$$

Let β be the dual optimal solution. We now want to bound $\frac{\beta}{f_t}$.

$$\beta = \min_w \frac{D(w)}{\alpha(w)} \leq \frac{D(w_i - w_0)}{\alpha(l_i - l_0)} \leq \frac{D(l_i) - D(l_0)}{\alpha(w_i) - \delta L}.$$

Substituting the bound on $D(i) - D(0)$ we get,

$$\alpha(i) \leq \delta L + \frac{\epsilon(1+\epsilon)}{\beta} \sum_{j=1}^{i} (f_j - f_{j-1})\,\alpha(j-1).$$

For any j note that $\alpha(j)$ is maximum when all these inequalities hold as equalities. This implies that

$$\alpha(i) \leq \delta L e^{\frac{\epsilon(1+\epsilon) f_i}{\beta}} \qquad i = 1, 2, \ldots, t.$$

Since

$$1 \leq \alpha(t) \leq \delta L e^{\frac{\epsilon(1+\epsilon) f_i}{\beta}}.$$

Therefore,

$$\frac{\beta}{f_t} \leq \frac{\epsilon(1+\epsilon)}{\log(\delta L)^{-1}}.$$

The scaling factor and hence the choice of the initial length δ for LCMF is different from to the regular flow problem in [4].

Lemma 1 *There is a feasible flow of value*

$$\frac{f_t}{\log_{1+\epsilon}\frac{(1+\epsilon)}{\delta}}.$$

Proof:
For every $u(e)$ units of flow the length increases by at least $1+\epsilon$. Since $l_t(e) < (1+\epsilon)$ the total flow on the link is $u(e)\log_{1+\epsilon}\frac{(1+\epsilon)}{\delta}$. Therefore, scaling the final flow f_t by $\log_{1+\epsilon}\frac{(1+\epsilon)}{\delta}$ creates a feasible flow. □

The ratio of the dual to the primal optimal solution γ is $\frac{\beta}{f_t}\log_{(1+\epsilon)}\frac{(1+\epsilon)}{\delta}$. Substituting the bound on $\frac{\beta}{f_t}$,

$$\gamma \leq \frac{\epsilon(1+\epsilon)}{\ln(1+\epsilon)}\frac{\ln\frac{(1+\epsilon)}{\delta}}{\ln(\delta L)^{-1}}$$

Setting the value of $\delta = L^{-\frac{1}{\epsilon}}(1+\epsilon)^{1-\frac{1}{\epsilon}}$,

$$\gamma \leq \frac{\epsilon(1+\epsilon)}{(1-\epsilon)\log(1+\epsilon)} \leq \frac{1+\epsilon}{(1-\epsilon)^2} \leq (1+4\epsilon)$$

Theorem 2 *The running time of the algorithm is $O(m\lceil\frac{1}{\epsilon}\log_{1+\epsilon}L\rceil T_{LCLP})$. where T_{LCLP} is the running time of the length constrained lightest path problem.*

Proof:
At iteration i, the length of the minimum capacity edge is increased by a factor of $1+\epsilon$. Since $l_t(e) < (1+\epsilon)$ the number of iterations in which e is the minimum capacity edge on the path set chosen is at most $\lceil\frac{1}{\epsilon}\log_{1+\epsilon}L\rceil$. Since there are m arcs, the total number of iterations is bounded by $m\lceil\frac{1}{\epsilon}\log_{1+\epsilon}L\rceil$. Each iteration involves solving one LCLP problem and the result follows. □

The analysis above is similar to the analysis of multicommodity max-flow (without path length restrictions) in [3] where the subproblems are solved within a factor of a lower bound in order to obtain improved running times.

5 Conclusions

With the widespread use of optical networks and with increased interest in providing low delay, jitter guarantees in packet networks, an important problem is the determination of network paths with bounded lengths. Optical networks have path length limitations due to signal degradation, amplification, and regeneration considerations. In packet networks, bounded hop-count paths are useful to ensure the QoS needs for the EF class. A basic question in network capacity allocation is the determination of the amount of traffic that can be routed between a network's end-points. Whereas this capacity can be determined efficiently using multi-commodity flow algorithms when routing is unconstrained, there is no satisfactory method to determine possible traffic capacity when network routing is constrained by path-length bounds. This paper presents simple, easy to implement, fully-polynomial time approximation schemes for solving these capacity allocation problems. We have extended the approach in this paper to solve the multicommodity capacity allocation problems [9] where the objective is to route a given demand matrix on a capacitated network with additional path length constraints.

References

[1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin "*Network Flows: Theory, Algorithms, and Applications*", Prentice Hall, 1993.

[2] Bennett., J., Benson K., Charny., A., Courtney., Leboudec., J-Y., "Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding", *To appear, IEEE/ACM Transactions on Networking*, 2002.

[3] Fleisher, L.K, "Approximating Fractional Multicommodity Flow Independent of the Number of Commodities", *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999.

[4] Garg, N., and Könemann, J., "Faster and Simpler Algorithms for Multicommodity Flow and other Fractional Packing Problems", *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pp.300-309, 1998.

[5] Guruswami, V., Khanna, S., Rajaraman, R., Shepard, B., and Yannakakis, M., "Near Optimal Hardness Results and Approximation Algorithms for Edge-Disjoint Paths and Related Problems ", *Proceedings of the ACM Symposium on the Theory of Computation*, 1999.

[6] Hassin, R., "Approximation Schemes for the Restricted Shortest Path Problem", *Mathematics of Operations Research*, Vol. 17, No. 1, pp. 36-42, 1992.

[7] Itai, A., Perl, Y., and Shiloach, Y., "The Complexity of Finding Maximum Disjoint Paths with Length Constraints", *Report 94*, Department of Computer Science, Technion, Haifa, Israel.

[8] Jaffe, J, "Algorithms for Finding Paths with Multiple Constraints", *Networks* , Vol. 14, pp. 95-116, 1984.

[9] Kodialam, M., and Lakshman, T.V., "Capacity Allocation in Networks with Path Length Routing Constraints ", *Submitted for Publication* .

[10] Lorenz, D., and Raz, D., "A Simple Efficient Approximation Scheme for the Restricted Shortest Path Problem", *Operations Research Letters*, Vol. 28, No. 5, pp. 213-219, 2001.

[11] Parekh., A., Gallager R., "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case", *IEEE/ACM Transactions on Networking* April, 1994.

[12] Plotkin, S., Shmoys, D., and Tardos, E., "Fast Approximation Algorithms for Fractional Packing and Covering Problems", *Mathematics of Operations Research*, Vol. 20, pp. 257-301, 1995.

[13] Ramaswamy, R., Sivarajan, K., "*Optical Networks: A Practical Perspective*", Morgan Kaufmann, 1998.

[14] Shahrokhi, F., and Matula, D., "The Maximum Concurrent Flow Problem", *Journal of the ACM*, 37, pp. 318-334, 1990.