

Rollout Algorithms for Logical Topology Design and Traffic Grooming in Multihop WDM Networks

Kwangil Lee

Department of Electrical and Computer Engineering
University of Texas, El Paso, TX 79928, USA.
Email: kilee88@yahoo.com

Mark A. Shayman

Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742, USA
Email: shayman@glue.umd.edu

Abstract—In this paper we consider the logical topology design and traffic grooming problem in multihop WDM networks. Usually, this problem is defined as an integer linear program (ILP) which is NP-complete. This justifies the use of heuristic algorithms. Many heuristic algorithms differ in the order of traffic demands considered for lightpath provisioning. We apply the technique of ‘rollout’ to systematically improve the performance of various heuristic algorithms by approximately optimizing the order in which traffic demands are considered. Through simulation experiments, we show that the performance of the rollout algorithms we derive are clearly superior not only to that of the initial heuristic algorithms on which they are based, but also to that of other well-known heuristic algorithms.

I. INTRODUCTION

By dynamically reconfiguring the optical switches to set up lightpaths, wavelength division multiplexing (WDM) networks can provide single hop communication channels between end nodes. This eliminates the electronic processing at intermediate nodes along the path and significantly reduces delay. However, it is generally impossible to provide single hop connectivity between each pair of end nodes due to limited number of router interfaces, optical impairments and other scalability issues. Consequently, it is necessary to create multihop paths and have electronic switching over multiple lightpaths for traffic between some source and destination pairs [1], [9], [12], [13].

Much research has been done on the logical topology design and traffic grooming problem. That research has focused largely on the formulation of the problem using integer linear programming (ILP), but the problem is known to be NP-complete [1], [3], [5], [12]. Consequently, many heuristic algorithms have been proposed. However, most of these algorithms deal with only direct (single-hop) connection setup between source and destination pairs using heuristic functions, considering interface availability at the source and destination [1], [3], [4], [6], [7]. In this case, grooming for multihop traffic is typically left for routing policy at a higher layer such as IP or MPLS [6], [7]. Recently, some algorithms have been proposed for multihop optical networks by relaxing some constraints [2], [12], [13]. Also, some graph models have been proposed for multihop logical topology design and traffic grooming [8], [9].

In this paper, we investigate the multihop logical topology design and traffic grooming problem with *VNG (Virtual Neighbor Graph)* [9]. For this, we propose rollout algorithms

for multihop optical network design so as to get potentially near-optimal solutions. Also, we analyze the logical topology design algorithms with extensive simulations. This paper is organized as follows. Section 2 describes the problem and an integrated algorithm which determines the logical topology and traffic grooming simultaneously. We introduce an interface constraint shortest path algorithm for our integrated logical topology design algorithm. Also, we describe ‘index rollout’ algorithms in Section 4. In Section 5, we analyze the performance of the algorithms using various metrics and compare the performance to that of other well-known algorithms in multihop environments. Finally, we conclude the paper in Section 6.

II. PROBLEM DEFINITION

The problems of optical network design are formulated as integer linear programs. The problem definition and notations in this paper follow those defined in [9].

A. Network Model

We consider a network connected by bidirectional optical links forming an arbitrary physical topology. However, lightpaths are not feasible between all nodes due to the optical impairments as well as the number of interfaces. We assume that the effect of the optical impairments is to place a constraint on the maximum length of each lightpath. To connect two nodes whose distance exceeds this upper bound requires more than one lightpath. Each link has only one fiber consisting of W wavelengths. Each optical node is equipped with an optical cross connect (OXC). We assume each OXC has no capability of wavelength conversion. So, when wavelength conversion as well as optical signal regeneration is required, two lightpaths are necessary. But our algorithms can be easily extended to include networks that contain a mixture of nodes with and without wavelength conversion. Each OXC is connected to an edge device, e.g., IP router. For simplicity, we refer to the combination of OXC and IP router as a ‘node’ in this paper. Each node has a limited number of transmitters and receivers. Traffic between nodes is symmetric and each traffic demand is less than the capacity of a wavelength. We are considering only a single path for each traffic demand. If there is no single path that can accommodate a traffic demand, that traffic demand is considered blocked.

B. Objective Function

Let $\text{hop}(s,d)$ denote the (logical) hop distance of the provisioned path for t_{sd} , the traffic demand between source s and destination d . β_{sd} is 1 if t_{sd} gets a route (with sufficient bandwidth) in the logical topology; 0 otherwise. Thus, β_{sd} is the blocking binary variable. Then, we consider two metrics as our objective functions: the network throughput and the weighted hop count as shown in Equations (1) and (2), respectively. To get the optimal solution for this problem is known to be NP-complete [1], [3], [5], [12]. So, this gives justification to use heuristic algorithms.

$$\text{Maximize : } \sum_{s,d} t_{sd} \beta_{sd} \quad (1)$$

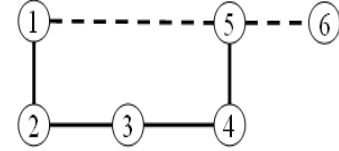
$$\text{Minimize : } \sum_{s,d} t_{sd} \text{hop}(s,d) / \sum_{s,d} t_{sd} \quad (2)$$

C. Integrated Logical Topology Design and Traffic Grooming

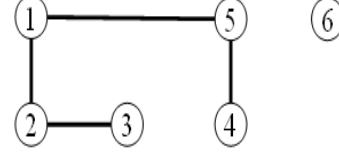
For the topology design and traffic grooming, we use an integrated approach, which performs the logical topology design and traffic grooming simultaneously. For this, we create an abstracted graph, which is called the *Virtual Neighbor Graph (VNG)*. The nodes in the virtual neighbor graph correspond to the network nodes. Given the current state of the logical topology design and grooming process, the VNG contains the information needed to determine what logical paths and (bidirectional) lightpaths can be formed. Two nodes x,y in the VNG are connected by an edge if either (1) there is already a lightpath between x and y , or (2) it is possible to construct a lightpath between them. This means that x and y both have available interfaces, are within the transmission range of each other, and the routing and wavelength assignment problem can be solved—i.e., there is a path from x to y and a wavelength that is available on all the optical links (the same wavelength). In this case, we say that there is a *potential link* between x and y . If x and y are connected by an edge in the VNG, we refer to them as (virtual) neighbors. Even though multiple lightpaths are available, we allow only one potential logical link between two nodes at each time.

Our integrated algorithm takes the following steps.

- 1) Initially, construct Virtual Neighbor Graph considering all constraints.
- 2) A traffic demand is chosen based on some criteria and (Section IV) and a shortest path is computed for the demand satisfying interface and bandwidth constraints (Section III). If none exists, then the demand is rejected.
- 3) If the path includes potential links, then those links are marked as actual links.
- 4) The capacity of each link on the path in the virtual graph is updated (decreased) to account for the bandwidth allocated to the demand routed.
- 5) The VNG is updated by eliminating all the potential links violate the interface constraints.



(a) Example Network



(b) Shortest Paths except node 6

Fig. 1. Interface Constraint Shortest Path Problem

- 6) Steps 2,3,4 and 5 are repeated until all traffic demands are considered. This way, a logical topology is created from the VNG and all the routes for the traffic demands are computed.

III. INTERFACE CONSTRAINT SHORTEST PATH PROBLEM

The interface constraint shortest path problem (ICSP) is to find a shortest path satisfying interface constraints. This problem occurs when finding a bi-directional shortest path between two nodes in the VNG. Consider the example shown in Figure 1. In this example, the interface constraint is two. And, solid links are actual links and dashed lines are potential links. All links are bi-directional and unweighted. We wish to find a shortest path in the graph consisting of actual links and potential links. If the computed path includes potential links, all of them are converted into actual links. This means that if the path contains a potential link from node i to node j , i and j must each have a free interface. Let us compute a shortest path from node 1 to each of the other nodes. As shown in Figure 1(b), a shortest path tree is formed for all nodes except for node 6. For node 6, the interface constraint rules out the path $1-5-6$ since node 5 has only one free interface but the path contains two potential links. Instead, the ICSP algorithm should find a path, $1-2-3-4-5-6$, for the request in the graph.

We proposed two algorithms to compute the optimal (shortest) path satisfying interface constraints: the perfect matching algorithm and the extension of Dijkstra's algorithm using 'gateway' nodes [18]. The complexity of these algorithms is $O(|E|^2 \log |E|)$ and $O(|V||E|)$ respectively. The ICSPF algorithms require more complexity than that of Dijkstra's algorithm, $O(|V|^2)$. We refer [18] for the details of the algorithms.

IV. ROLLOUT ALGORITHMS FOR LOGICAL TOPOLOGY DESIGN AND TRAFFIC GROOMING

The performance of the logical topology design and traffic grooming depends on the order in which the traffic demands are considered. We start with reasonable heuristics for demand ordering and use the rollout technique to improve the heuristics to obtain potentially near-optimal solutions.

A. Basic Rollout Algorithm

Rollout is a general method for obtaining an improved policy for a Markov decision process starting with a base heuristic policy. The rollout policy is a one step lookahead policy, with the optimal cost-to-go approximated by the cost-to-go of the base policy [17]. We use the specialization of rollout to discrete multistage deterministic optimization problems. Consider the problem of maximizing $G(u)$ over a finite set of feasible solutions U . Each solution u consists of N components $u = (u_1, \dots, u_N)$. We can think of the process of solving this problem as a multistage decision problem in which we choose one component of the solution at a time. Suppose that we have a heuristic algorithm, the so-called “base heuristic”, that given a partial solution $u = (u_1, \dots, u_n)$ ($n < N$) extends it to a complete solution $u = (u_1, \dots, u_N)$. Let $H(u_1, \dots, u_n) = G(u_1, \dots, u_N)$. In other words, the value of H on the partial solution is the value of G on the full solution resulting from application of the base heuristic. The rollout algorithm R takes a partial solution (u_1, \dots, u_{n-1}) and extends it by one component to $R(u_1, \dots, u_{n-1}) = (u_1, \dots, u_n)$ where u_n is chosen to maximize $H(u_1, \dots, u_n)$. Thus, the rollout algorithm considers all admissible choices for the next component of the solution and chooses the one that leads to the largest value of the objective function if the remaining components are selected according to the base heuristic. The rollout algorithm typically achieves a substantial performance improvement over the base heuristic at the expense of extra computation that is equal to the computation time of the base heuristic times a factor that increases polynomially with the problem size.

B. Rollout Algorithms for the Logical Topology Design and Traffic Grooming

We propose rollout algorithms for the topology control and traffic grooming in this paper. The order in which traffic demands are considered for lightpath provisioning plays an important role in determining the throughput of the resulting topology. We call a rollout that seeks to optimize this order, *Index Rollout*.

Suppose we are given a heuristic algorithm that specifies the order in which demands are to be routed using the VNG. If a partial topology has been obtained by choosing routes using the VNG for n demands (t_1, \dots, t_n) from the traffic matrix, the base heuristic specifies which demand t_{n+1} should be routed next. For this demand, it chooses a route using interface constrained shortest path first (ICSPF) in the VNG as explained in the previous section. The route chosen for this demand is a shortest bi-directional path in the partial topology satisfying the constraints. This means that every link in the path must have sufficient residual bandwidth for the demand; every potential link in the path must have available interfaces at both its head node and at its tail node. If there is no feasible path, then the ‘null’ route is assigned—i.e., the demand is blocked. Once t_{n+1} has been routed, the base heuristic selects and routes the next demand t_{n+2} in the same way using the partial topology existing after t_{n+1} has been routed. The base

heuristic algorithm continues in this way until all demands have been routed (or assigned null routes).

The index rollout algorithm works as follows: In the first step, the rollout algorithm uses ICSPF to route the demand t_1 determined by the requirement that it maximize the total network throughput when the base heuristic is used to complete the topology starting with t_1 . Now, suppose that the demands (t_1, \dots, t_{n-1}) have been routed in this order by the rollout algorithm. In the next step, the rollout algorithm uses ICSPF to route the remaining demand t_n determined by the requirement that it maximize the total network throughput when the base heuristic is used to complete the topology starting with (t_1, \dots, t_n) .

C. Heuristic Algorithms

In this section, we explain three base heuristic algorithms to which we apply the index rollout. These algorithms were proposed for the integrated logical topology design and traffic grooming in [9].

1) *Maximum Traffic Demands (MTD)*: The traffic demands are considered in order of decreasing magnitude; the s-d pair with maximum traffic demand is considered first for logical topology design and traffic grooming.

2) *Maximum Network Resource Usage (MNR)*: In this algorithm, at each step we find the traffic demand that requires maximum network resources and provision for that demand next. The amount of network resources required for each demand is computed by multiplying the traffic demand by the number of logical hops that would be required if the demand were provisioned now. We refer to this number of hops as the expected number of logical hops (ELH).

3) *Resource Efficiency Factor (REF)*: By allocating network resources efficiently, we can achieve higher network performance. The resource efficiency in topology design and traffic grooming is related to the number of (logical) links that are used by a flow. We define the resource efficiency factor (REF) to be the quotient obtained by dividing each traffic demand by the ELH, the expected number of logical hops.

V. ANALYSIS

A. Complexity Analysis

We compare the complexity of our algorithms with other well-known algorithms, HLDA (Heuristic Logical Topology Design Algorithm) [1], [3] and MRU (Maximum Resource Utilization) [1]. In their original form, these algorithms do not include a distance constraint. Thus, in order to make them comparable to our algorithms, we modify them to include such a constraint. Our algorithms determine logical topology and traffic grooming simultaneously. But, these tasks are independent in HLDA and MRU. For comparison, we assume that shortest path routing policy is used for the purpose of traffic grooming for residual traffic demands after the logical topology design. Also, we assume that ICSPF algorithm with ‘gateway’ is used for our integrated algorithm. For RWA problem, shortest path and FF(First-Fit) algorithm is used. Table 1 summarizes the complexity of the algorithms for the

TABLE I
SUMMARY OF THE COMPLEXITY OF THE ALGORITHMS

Algorithm	Complexity
HLDA	$O(M V ^2)$
MRU	$O(M^2 V ^2)$
MTD	$O(M V E)$
MNR	$O(M^2 V E)$
REF	$O(M^2 V E)$
rMTD	$O(M^2 V E)$
rMNR	$O(M^3 V E)$
rREF	$O(M^3 V E)$

logical topology design and traffic grooming, where M is the number of source-destination pair. Algorithms rMTD, rMNR and rREF refer to index rollout algorithms with MTD, MNR and REF respectively.

B. Experimental Analysis

1) *Simulation Environment*: We analyze the proposed integrated logical topology design and traffic grooming algorithms through simulations using the GLASS/SSF simulator [14], [15]. We consider a 16-node NSFNet physical topology. We assume that each link has 20 wavelengths and there is only one optical link (fiber) between nodes. And, each node has no wavelength conversion capability. So, if wavelength conversion is required, then setup of two lightpaths is necessary. In our simulations, each node has five transmitters and receivers. The capacity of each wavelength is normalized to one bandwidth unit (BU). Each entry in the traffic matrix represents the aggregated traffic demand of a source-destination pair. It is generated independently using the uniform distribution between 0 and 0.5 BU. For the analysis, we used 10 randomly generated traffic matrices. Also, we have run simulations with different values for the maximum number of (physical) hops permissible for lightpaths. But, we present only the simulation results when the length of each lightpath is constrained to be at most three since the results are very similar. For the case where the lightpath length is unconstrained, we refer to our previous work [9]. We measured the performance metrics of weighted hop count and network throughput. The objective function used for rollout is network throughput. And, the weighted hop count is used for tie-breaking.

2) *Heuristic Algorithms*: We compared the performance of the three heuristic algorithms we proposed and the other well known algorithms. The network throughput and weighted hop count are shown in Figures 2 and 3. Except for REF, the integrated algorithms show poorer performance than the other heuristics. HLDA and MTD provide a good example for comparison because both algorithms use the same ordering of the entries in the traffic matrix for the lightpath provisioning. However, HLDA yields 6% better network throughput and 4% better weighted hop count compared to MTD. The only difference between HLDA and MTD is how to handle multihop traffic when either the source or destination of the demand does not have free interfaces. HLDA leaves the path selection for multihop traffic to the routing, which is done after the logical topology is complete. On the other hand, MTD selects the paths for multihop traffic earlier, as part of the logical

topology design. A large demand will be routed early even if it requires a large amount of resources—i.e., many logical hops. This can result in later demands being blocked.

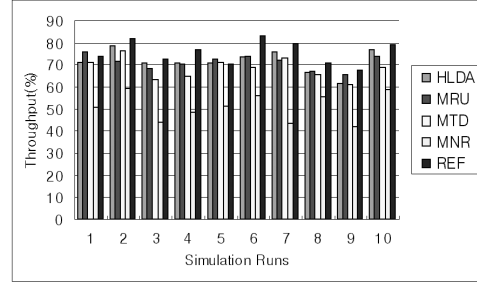


Fig. 2. Throughput: Heuristic Algorithms

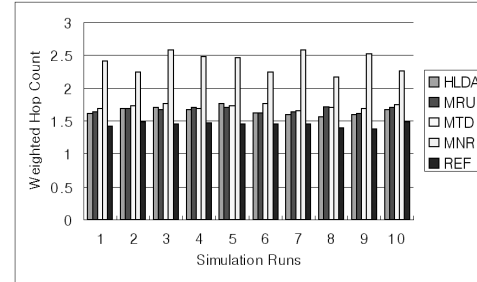


Fig. 3. Weighted Hop Count : Heuristic Algorithms

We also compare REF algorithm and other heuristic algorithms. As shown in the figures, REF algorithm works better than any other algorithm as measured by both weighted hop count and network throughput. REF increases the network throughput 4 and 6 % compared to HLDA and MRU respectively. It also reduces weighted hop count by between 14 and 16%. Compared to MTD, REF increases the network throughput 8 % and reduces the weighted hop count 18%. It also increases the network throughput 47% and reduces the weighted hop count by 67% compared to MNR. REF considers not only the magnitude of the demands but also the number of (logical) hops to be routed. In contrast to MTD and MNR, REF will delay the routing of a large traffic demand if to do so would require a large number of logical hops.

3) *Base Heuristics vs. Rollout Algorithms*: We are interested in how much the performance of the heuristics is improved by applying rollout. The results are shown in Figures 4 and 5.

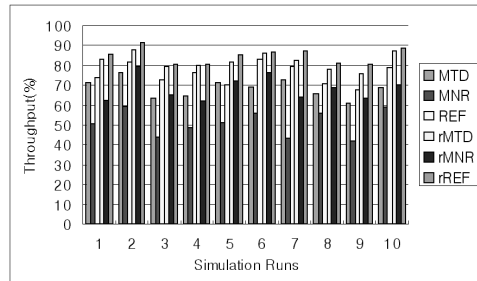


Fig. 4. Throughput: Basic Heuristic vs. Rollout Algorithms

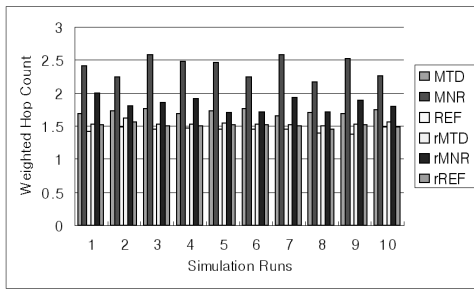


Fig. 5. Weighted Hop Count : Base Heuristic vs. Rollout Algorithms

As shown in the figures, significant performance improvements are achieved by the rollout algorithms. For the network throughput, rMTD and rREF outperform MTD and REF by 15 and 20% respectively. Also, the throughput of rMNR exceeds that of MNR by up to 50% and average 35 %. The weighted hop count is also reduced. rMTD and rMNR reduce the weighted hop count 11 and 31 % compared to their base heuristics. But, rREF increases the weighted hop count 4 % compared to REF. Although the weighted hop count of rREF is increased, it is still smaller than that of the other algorithms. Since we are rolling out with respect to the performance measure of throughput, rather than weighted hop count, it is not especially surprising to see a negative effect on weighted hop count.

4) *Heuristic Algorithms vs. Rollout Algorithms:* Finally, we compared the performance between well known heuristic algorithms and rollout algorithms. The results are shown in Figures 6 and 7.

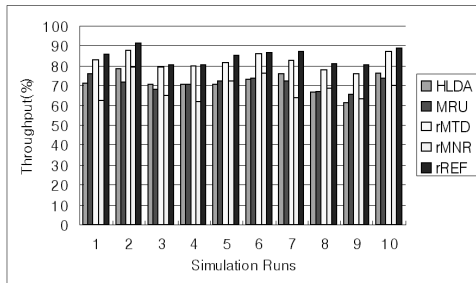


Fig. 6. Throughput: Heuristic vs. Rollout Algorithms

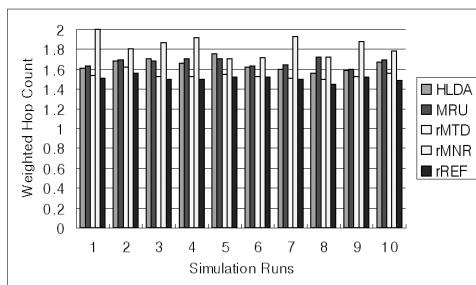


Fig. 7. Weighted Hop Count : Heuristic vs. Rollout Algorithms

In the figures, we see that the rollout algorithms except rMNR perform better than the HLDA algorithm. Note that MTD performs worse than HLDA. However, rMTD performs better than HLDA. The rollout algorithms (rMTD and rREF)

increase the network throughput 14 and 12 % and reduce the weighted hop count 6 and 9% compared to HLDA. Also, rMTD and rREF increases the network throughput 13 and 15%, and reduce the weighted hop count 9 and 11% compare to MRU.

VI. CONCLUSION

In this paper, we have described our work on the topology design and traffic grooming in multihop WDM networks. To get a potentially near optimal solution, we start with reasonable heuristics use the technique of rollout to improve the order in which traffic demands are considered. Three different rollout algorithms are obtained corresponding to different choices for the base heuristic. The simulation experiments show that the rollout algorithms improve the network throughput and the weighted hop count significantly at the expense of increased computational complexity. Of these, rREF, index rollout of REF, shows the best performance in the simulations. This confirms that REF is a good heuristic for logical topology design and traffic grooming.

REFERENCES

- [1] K. Zhu, B. Mukherjee, "Traffic Grooming in an Optical WDM Mesh Networks," IEEE JSAC, pp. 122-133, Jan. 2002.
- [2] M. Kodialam, T. V. Lakshman, "Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks," IEEE Infocom, pp. 358-366, 2001.
- [3] R. Ramaswami, K. N. Sivarajan, "Design of Logical Topologies for Wavelength-Routed optical Networks," IEEE JSAC, pp. 840-851, Jun. 1996.
- [4] B. Mukherjee, Optical Communication Networks, New York, McGraw-Hill, 1997.
- [5] D. Banerjee, B. Mukherjee, "Wavelength-routed Optical Networks: Linear Formulation, Resource Budgeting Tradeoffs and a Reconfiguration Study," IEEE/ACM Trans. Networking, vol. 8, pp. 598-607, Oct. 2000.
- [6] K. H. Liu, C. Liu, J. L. Pastor, A. Roy, J. Y. Wei, "Performance and Testbed Study of Topology Reconfiguration in IP over WDM," IEEE Transactions on Communications, 2002.
- [7] E. Leonardi, M. Mellia, M. A. Marsan, "Algorithms for the Logical Topology Design in WDM All-Optical Networks," Optical Networks Magazine, pp. 35-46, Jan. 2000.
- [8] H. Zhu, H. Zang, K. Zhu, B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," IEEE/ACM Transaction on Networking, vol. 11, no. 2, Apr. 2003.
- [9] K. I. Lee, M. Shayman, "Optical Network Design with Optical Constraints in IP over WDM Networks," ICCCN, Oct. 2004.
- [10] J-F. P. Labourdette, A.S. Acampora, "Logically Rearrangeable Lightwave Networks," IEEE Transactions on Communications, pp. 1223-1230, Aug. 1991.
- [11] A. S. Acampora, G. W. Hart, J-F. P. Labourdette, "Branch-Exchange Sequences for Reconfiguration of Lightwave Networks," IEEE Transactions on Communications, vol. 42, no. 10, pp. 2822-2832, Oct. 1994.
- [12] J. Q. Hu, B. Leida, "Traffic Grooming, Routing, and Wavelength Assignment in Optical Mesh Networks," Apr. 2002.
- [13] L. Gouveia, P. Patricio, A. F. Sousa, R. Valadas, "MPLS over WDM Network Design with Packet Level QoS Constraints based on ILP Models", IEEE Infocom, Apr. 2003
- [14] GLASS simulator, <http://dns.antd.nist.gov/glass/>
- [15] SSFnet, <http://www.ssfnet.org>
- [16] Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin, Network Flows: Theory, Algorithms and Applications, Prentice Hall, 1993.
- [17] D. P. Bertsekas, Dynamic Programming and Optimal Control, Athena Scientific, 2000.
- [18] S. Khuller, K. I. Lee, M. Shayman, "On the Degree Constraint Shortest Paths," 13th Annual European Symposium on Algorithms (ESA '05), Oct. 2005.