# Network Planning for Stochastic Traffic Demands

Phuong Nga Tran, Bharata Dwi Cahyanto, and Andreas Timm-Giel

Institute of Communication Networks
Hamburg-Harburg University of Technology, Hamburg, Germany
{phuong.tran,bharata.cahyanto,timm-giel}@tuhh.de

**Abstract.** Traffic in communication networks is not constant but fluctuates heavily, which makes the network planning task very challenging. Overestimating the traffic volume results in an expensive solution, while underestimating it leads to a poor Quality of Service (QoS) in the network.

In this paper, we propose a new approach to address the network planning problem under stochastic traffic demands. We first formulate the problem as a chance-constrained programming problem, in which the capacity constraints need to be satisfied in probabilistic sense. Since we do not assume a normal distribution for the traffic demands, the problem does not have *deterministic equivalent* and hence cannot be solved by the well-known techniques. A heuristic approach based on genetic algorithm is therefore proposed. The experiment results show that the proposed approach can significantly reduce the network costs compared to the peak-load-based approach, while still maintaining the robustness of the solution. This approach can be applied to different network types with different QoS requirements.

**Keywords:** Network planning, stochastic traffic demands, chance constrained programming, genetic algorithm.

## 1 Introduction

Network planning is an old but never outdated research topic in telecommunication networks. It is a very complex task because it has to resolve the conflict of interest between the network service provider, who wants to minimize the expenditure and the services users, who expect a good QoS. Accurate network planning is one of the crutial factors that ensure a business success for the network operators.

The classical network planning problem, assuming static traffic demands given by single trafic matrix, has been studied extensively for decades [1]. Different studies focused on different network technologies with different QoS requirements. But the general objective is to find a network with minimum cost, which can accomodate the given traffic demand. The common approach to solve this problem is to model it as Linear Programming (LP) problem and use some well-known optimization tools, e.g. CPLEX [2] to solve it. Besides, many researchers

D. Pesch et al. (Eds.): MONAMI 2013, LNICST 125, pp. 216–227, 2013.

have also proposed heuristic algorithms such as genetic algorithm, simulated annealing, local search, and etc, to solve the problem for large networks whose solution cannot be obtained from CPLEX within a reasonable time. Even though the above problem is already very complex, its solution can be inefficient. This is because the traffic is not deterministic but fluctuates heavily over time, as shown in Fig.1. If the traffic demand matrix covers the peak rate of the traffic demands, the network will be very costly due to the overestimated traffic volume. If the traffic demand matrix represents the mean rate, the resulted network may not be able to guarantee a certain QoS. Therefore, the traffic demands must be represented by so-called *effective rates*, which are between the peak rates and the mean rates. However, there is so far no effective way to determine these rates that can result in the most efficient network.

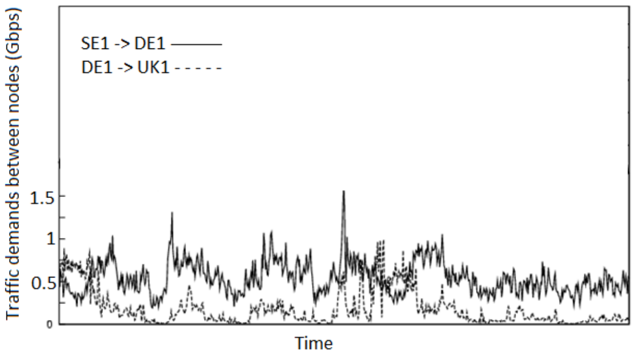


**Fig. 1.** Traffic fluctuation - Data was taken from GEANT project [3]

Assuming static traffic demands may be acceptable when the traffic characteristic is unknown to the operator. However, for a better network planning, one should carefully study the potential traffic behaviour and has some assumption on the stochastic traffic model. Moreover, once the network is already under operation and the traffic volume can be easily captured by measurement, re-optimizing (re-planning) the network should definitely take into account the stochastic behavior of the traffic. Nowadays, when the traditional networking infrastructure tends to migrate to the cloud networking, in which the network resourses can be easily accommodated on demand, the re-planning should be done frequently to optimize the expenditure as well as the network QoS. This motivates us to study the network planning problem for stochastic traffic demands.

## 2 Related Work

The network design problem under traffic uncertainty has attracted many researchers. Many methods have been proposed to handle the data uncertainty.

The first approach is to use Stochastic Programming [4]. In this approach, traffic uncertainty is captured by a finite number of matrice, each of which is assumed to occur with a certain probability. This is then modelled by a deterministic equivalent linear programming problem. Even though it is one of the earliest techniques to deal with uncertainty data, it is not very popular in telecommunications.

The second approach, a well-known method in communication network design, is Robust Optimization introduced by Soyster in 1973 [5]. Using this approach, no information of the probabilistic distribution of the traffic uncertainty is required. Instead, a solution is robust if it is feasible for all traffic volumes in the given uncertainty set. Robust Optimization can be applied differently in network design. In multi-hour network planning, the traffic fluctuation is modelled by multiple traffic matrice [6, 7]. The network is designed so that each traffic matrix can be accommodated non-simultaneously in installed capacities. This problem can be formulated as an LP problem in a similar way to the classical network planning, but it is much more complex due to a larger ammount of constraints. Another realization of Robust Optimization is to use the *hose model* [8]. The model defines upper bounds on the sum of the incoming and outgoing traffic flows for all network nodes while allowing each traffic flow to vary. This model has attracted a lot of attention recently [9–11]. In 2004, Bertsimas and Sim introduced the $\Gamma$-model as an extention of Robust Optimization [12]. In realistic scenarios, it is unlikely that all traffic demands are at their peak rate at the same time. Hence, in the $\Gamma$-model, a (small) non-negative value $\Gamma$ is introduced to restrict the number of simultaneous peaks. Changing $\Gamma$ relates to adjusting the robustness and the level of conservatism of the solutions and therefore provides additional flexibility. The $\Gamma$-model was used for robust network design in [13–16]. The weakness of this model is that its complexity increases exponentially with $\Gamma$ due to a large combination of simultaneous peaks. Additionally, the choice of $\Gamma$ is rather vague because it does not say much about the robustness of a solution although they are related.

The third approach is to use Chance-Constrained Programming (CCP) introduced by Charnes and Cooper in 1959 [17]. In CCP, the constraints must be maintained at a prescribed level of probability. In communication network planning, these are usually capacity constraints guaranteed at a certain probability, which is actually the overload probability of the links. Using the CCP, we must assume that the probability distribution of traffic demands is known. This is usually not the case of greenfield network planning. However, when a network already exists and re-optimizing is required, this information can be obtained from traffic measurement data. If the traffic follows a normal distribution, the CCP problem has a deterministic equivalent, and hence becomes an LP problem, which can be solved by optimization tools. If the traffic follows a log-normal distribution, the CCP problem can be approximated by a deterministic equivalent, which turns into an LP problem as well. In other cases, the problem is very hard. Our contribution in this work is a genetic algorithm to solve the network planning problem for stochastic traffic demands with arbitrary probability distribution modelled by Chance-Constrained Programming.

The rest of the paper is organized as follows. Section 3 presents the mathematical model of the problem using CCP. Section 4 introduces our proposed genetic algorithm. Section 5 discusses the performance of the algorithm and Section 6 concludes our work.

## 3 Problem Formulation

We consider the following network design problem. An undirected connected graph $G = (V, E)$ representing a potential network topology is given. On each link, capacity can be installed with a certain cost. The installed capacity is bounded by the available physical capacity of the link. A traffic demand between any two nodes is stochastic and is given by a (discrete) probability distribution function. Traffic demands are assumed to be statistically independent from each other. A coefficient $\epsilon$ is introduced as the QoS parameter. The task is to find a network with the minimal cost, in which the overload probability of each link is bounded by $\epsilon$.

The problem can be mathematically presented using the following notations:

- $k \in K$ denotes a commodity representing a traffic demand.
- $l$ denotes a link in the potential topology.
- $r \in R$ denotes a route conneting the source and destination nodes. A set of possible routes $R$ connecting any two nodes is pre-computed.

### Given Parameters

- $T_k$: traffic demand of the commodity $k$. Since the traffic demands are stochastic, $T_k$ is a probability density function (PDF).
- $C_l$: available physical capacity of link $l$.
- $c_l$: cost of a bandwidth unit on link $l$.

### Decision Variables

- Bandwidth assignment $b_l$: a positive real variable denoting the bandwidth allocated on link $l$
- Routing variable: $f_r^k = 1$ if the traffic of the commodity $k$ is routed through route $r$. Otherwise, $f_r^k = 0$. We assume a single-path routing, hence $f_r^k$ is a binary variable.

### Constraints

- Routing constraint:

$$\sum_{r \in R} f_r^k = 1 \qquad \forall k \tag{1}$$

Equation (1) ensures that every flow is routed through one of the pre-computed paths.

Assuming $N (N \leq K)$ traffic flows going through link $l$ and $T_k$ being the PDF of each traffic demand, the PDF of the aggregated traffic on link $l$, $T^l$, is the convolution of all flows and given as:

$$T^l = T_1 \otimes T_2 \otimes ... \otimes T_N \tag{2}$$

– Link overload probability constraint:

$$\int_0^{b_l} T^l(x)\, dx \geq 1 - \epsilon \qquad \forall l \tag{3}$$

Equation (3) guarantees that the traffic load on link $l$ is smaller than or equal to its installed capacity $b_l$ with the probability of $1 - \epsilon$. This equals to link overload probability smaller than $\epsilon$. Equation (3) is equivalent to:

$$\prod_{k, f_l^k = 1} \int_0^{b_l} T_k(x)\, dx \geq 1 - \epsilon \qquad \forall l \tag{4}$$

– Physical capacity constraint:

$$b_l \leq C_l \qquad \forall l \tag{5}$$

**Objective**

$$\min \sum_l c_l \cdot b_l \tag{6}$$

The objective is simply to minimize the total network cost.

## 4 Genetic Algorithm

The mathematical model presented in the previous section is a non-linear optimization problem. In this section, we introduce a heuristic approach based on the genetic algorithm to solve it. Genetic algorithm uses the concepts of population genetics and evolution theory to optimize the *fitness* of a *population* of *individuals* through *mutation* and *crossover* of their *genes*. The advantage of the genetic algorithm is that it is able to explore a large solution space and hence to avoid local optima.

### 4.1 Encoding

A chromosome is encoded by a set of numbers $\{r_1, r_2, ..., r_k, ...r_K\}$ where $r_p$ is a positive integer representing a route of commodity $p$. Each position $k$ is related to a certain commodity and the corresponding traffic demand. A chromosome thus represents the routing solution for all demands on the network.

### 4.2 Forming a New Population

A new population is evolved by a mechanism to select and to form new individuals using genetic operators called *crossover* and *mutation*. Crossover produces new individuals that inherit genes from their parents while mutation enables offsprings to have different genes from their parents. Both of these genetic operators aim to produce some (hopefully) better individuals for the next generation (iteration). The bad performing individuals (according to the fitness parameter) from the previous iteration will be naturally removed and substituted by the new ones. In this work, each population is composed of 20 individuals. At each iteration, 20 new individuals will be generated.

### 4.3 Calculating the Solution Cost and Fitness Evaluation

The cost of a solution is the the total cost of capacities installed on all links to accommodate traffic using the routing solution represented by an encoded chromosome. From the chromosome, one knows which flows are going through each link. The PDFs of all traffic flows are convoluted to get the PDF of the aggregated traffic flow. After that, we can calculate the bandwidth needed on the link so that the overload probability is bounded by a given parameter $\epsilon$. In this work, we assume a discrete PDF, which is represented by a finite vector, for the ease of the computation. If the PDFs of the traffic demands are given as continuous functions, they should be discretized for the convolution computation.

Fitness evaluation is to decide which chromosomes meet the expectation and can be carried to the next iteration. In this work, the fitness is reflected by the total cost of a solution. The current five best solutions (with lowest cost) are always chosen to the next step. The other solutions including the invalid solutions, are chosen with a certain probability. The reason to choose some bad solutions is to avoid the local optima.

### 4.4 Algorithm Framework

The algorithm is represented in details by the flowchart in Fig. 2 The algorithm starts with 20 initial individuals encoded into chromosomes as described in section IV.A. From this initial population, 20 child solutions are generated by crossover or mutation of random genes. The cost of these solutions (including the parent solutions) are calculated and evaluated. 20 solutions which qualify the fitness evaluation will be carried to the next iteration. The process is repeated until a termination condition is reached. The condition can be for example a pre-determined number of iterations or a certain number of iterations during which no better solution is found.

## 5 Performance Evaluation

To evaluate the performance of the proposed network planning algorithm, we carried out two experiments on a small 6-node network and the German 16-node network, shown in Fig.3. The genetic algorithm was implemented in C++.

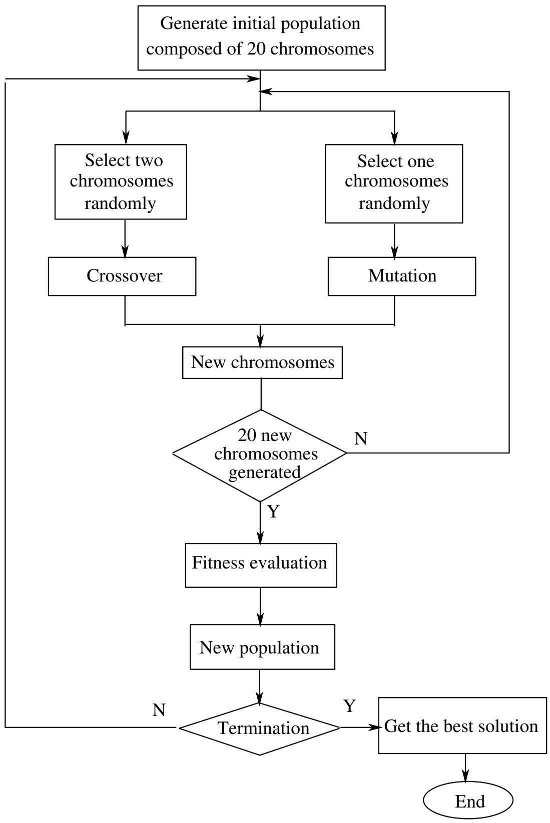


**Fig. 2.** Genetic algorithm framework

The traffic demands for the 6-node network are taken from GEANT project [3], while the traffic demands for the German network are taken from the DFN network provided at [18]. The traffic data was firstly extracted and its discrete PDF was constructed. We assumed that the cost of a bandwidth unit on each link is 1, so that the total cost is simply the sumation of the bandwidths. For the German network, 5 shortest paths were pre-calculated while for the 6-node network, all possible routes were considered.

### 5.1 Performance of Genetic Algorithm

In this work, the number of iterations in the genetic algorithm is set to 500 for the small network and 1000 for the big network. In practice, one can terminate the algorithm based on the quality of the solution, e.g. the objective function reaches a certain value or the best solution does not change over a certain number of iterations. Fig.4 and 5 show the evolution of the genetic algorithm over time (iterations) in two experiments. The red curve represents the best solution at each iteration while the blue one represents the average of the whole population.

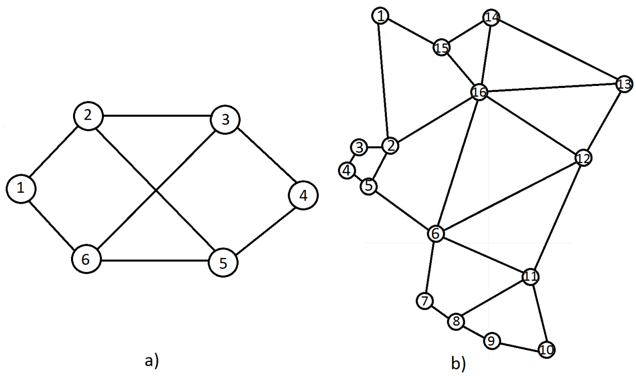


**Fig. 3.** Network topologies: a) 6-node network, b) German 16-node

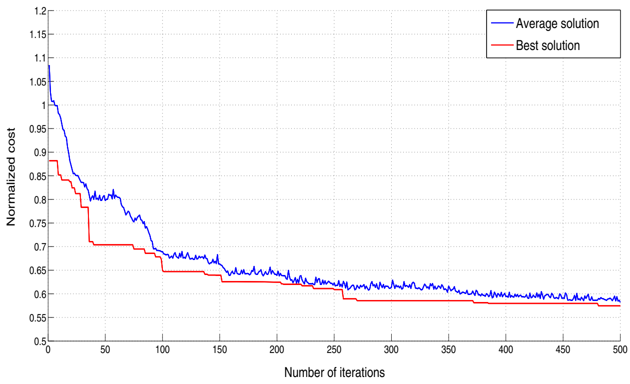


**Fig. 4.** Performance of genetic algorithm - 6-node network

The best solution always gets better or at least stays the same after each iteration, while the average solution can get worse. This is due to the fact that a bad solution can still be selected to the next iteration with a certain probability. This helps to avoid the local optima. As can be seen from the figures, the larger the network is, the more iterations are required to obtain a good solution. For the small network, from the iteration 250, the solution is improved very slowly and not much. For the German network, the solution still has potential to be improved at the iteration 1000. Therefore, it is difficult in practice to determine the number of iterations before terminating the algorithm. It is recommended to terminate the algorithm when the best solution does not change for a certain time.

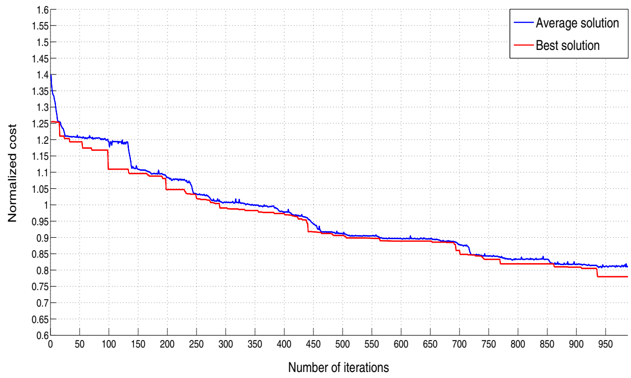


**Fig. 5.** Performance of genetic algorithm - German network

### 5.2 Network Cost

The resulting network cost under different link overload probability is shown in the Fig.6 and 7. The higher the acceptable link overload probability is, the lower is the network cost. Especially, the network cost in case of a small link violation probability can decrease significantly compared to the one when no link overload is tolerated at all. The cost decreases slowlier when the link overload probability is high. Therefore, it makes sense to accept a small link violation probability to reduce the cost while not sacrifying much the QoS.

In the small network, if 5% link overload proability is accepted, we can save about 45% of the cost. However, for the large network, at 5% link violation probability, we save only about 25% network cost. This is because in a large network, there are many more flows going through a link. The traffic loads on links tend to be averaged out, rather than heavily fluctuate. This avoids some extreme peak and hence the cost saving from a small violation probability is also reduced. This can be seen in the Fig.8 and 9.

**Table 1.** Comparison of statistical network planning with other approaches

| | Peak-based | $\epsilon = 0\%$ | Mean-based | $\epsilon = 5\%$ |
|---|---|---|---|---|
| Normalized cost | 1.24 | 1 | 0.45 | 0.53 |
| Link overload prob. | 0% | 0% | 18% - 25% | 5% |

Table 1 shows the comparison of the proposed statistical network planing method with mean-load-based and peak-load-based approach for the 6-node network. The solution for the mean-load-based and peak-load-based approaches are found by linear programming (CPLEX). Using the peak-load-based approach, the normalized network cost is 1.24 while using the proposed statistical planning method, the normalized cost is 1. In both cases, the link violation probability is

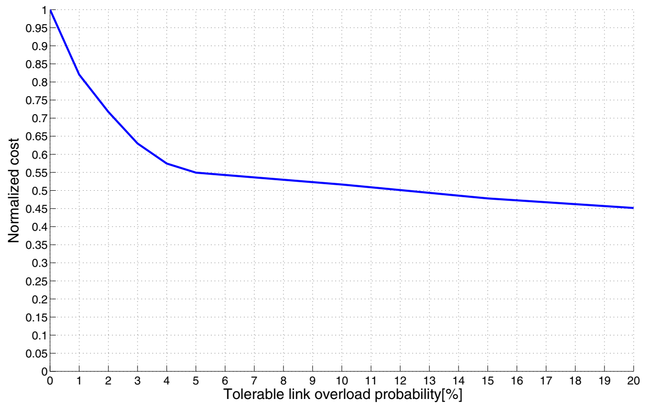


**Fig. 6.** Network cost vs. link overload probability - 6-node network

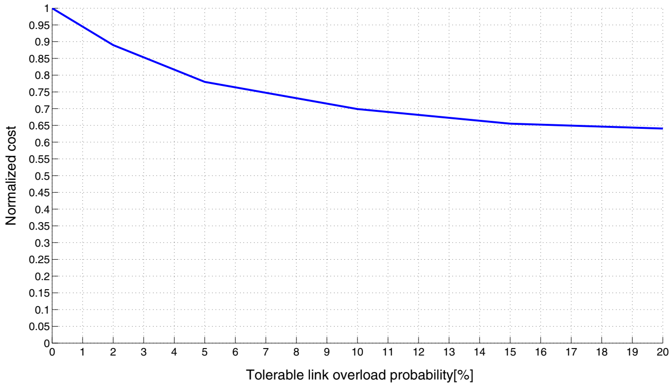


**Fig. 7.** Network cost vs. link overload probability - German network

0%. Using the mean-load-based approach, the cost is low (only 0.42) but the QoS is also very poor. The link overload probability ranges from 18%-25%. Accepting 5% of link overload probability results in a good compromise.

Another advantage of this proposed approach is that for different networks with different QoS requirements, one can simply adjust the link violation probability accordingly.

## 5.3 Aggregated Traffic on Links

Fig.8 and 9 shows the traffic load on a link in the 6-node network and the German network, respectively. In each figure, the red line indicates the bandwidth allocated on the link. As can be seen from the figure, the link overload probability is bounded by the predefined parameter $\epsilon = 5\%$. We can see that by accepting a small violation probability, the capacity needed on the link is reduced, especially for the small network.

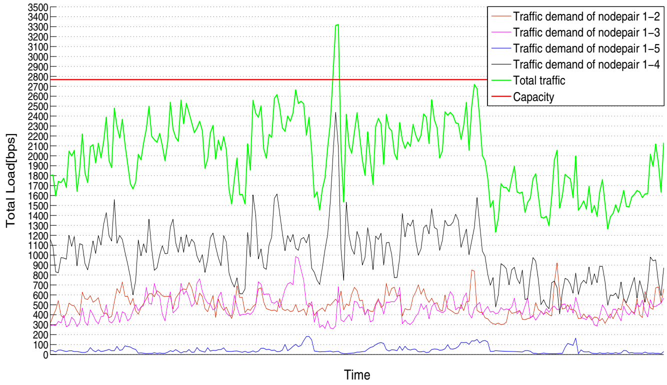


**Fig. 8.** Traffic load on the link (1-2) - $\leq 5\%$ violation probability

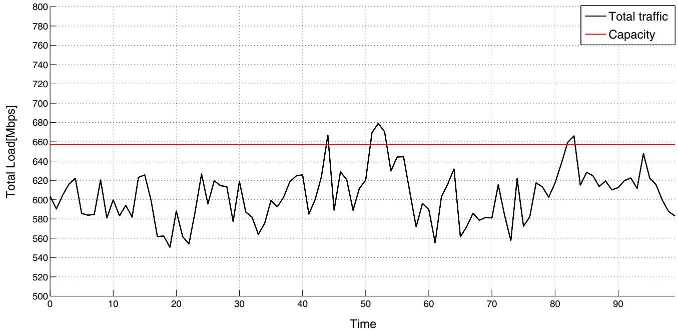


**Fig. 9.** Traffic load on the link (1-2) - $\leq 5\%$ violation probability

## 6 Conclusion

In this paper, we have proposed a new approach to solve the network planning problem under stochastic traffic demands using the genetic algorithm. The proposed method guarantees the network to carry stochastic traffic under a predefined link overload probability. The experiments showed that by accepting a small link overload probability, the network cost can be reduced significantly. Compared to the peak-load-based approach, the proposed method applied for $\epsilon = 0\%$ results in a clearly lower cost. Even though the algorithm cannot guarantee the optimal solution (due to the nature of the heuristic), it provides a relatively good solution. The limitation of this approach is that it guarantees the overload probability for each link only. In the future, we will develop an algorithm that can guarantee the overload probability for the end-to-end flows.

## References


1. Bienstock, D., Chopra, S., Gnlk, O., Tsai, C.Y.: Minimum cost capacity installation for multicommodity network flows. Mathematical Programming 81, 177–199 (1998)
2. Ibm website, `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`
3. Geant project, `http://www.geant.net`
4. Dantzig, G.B.: Linear programming under uncertainty. Management Science, 197–206 (1955)
5. Soyster, A.L.: Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research 21, 1154–1157 (1973)
6. Pavon-Marino, P., Garcia-Manrubia, B., Aparicio-Pardo, R.: Multihour network planning based on domination between sets of traffic matrices. Computer Networks 55(3), 665–675 (2011)
7. Aparicio-Pardo, R., Skorin-Kapov, N., Pavon-Marino, P., GarciaManrubia, B.: (Non)-reconfigurable virtual topology design under multihour traffic in optical networks. IEEE/ACM Transactions on Networking 20(5), 1567–1580 (2012)
8. Duffield, N.G., Goyal, P., Greenberg, A.G., Mishra, P.P., Ramakrishnan, K.K., van der Merive, J.E.: A flexible model for resource management in virtual private networks. In: ACM SIGCOMM, Cambridge, USA, pp. 95–108 (September 1999)
9. Chekuri, C., Oriolo, G., Scutella, M.G., Shepherd, F.B.: Hardness of robust network design. Networks 50(1), 50–54 (2007)
10. Altin, A., Yaman, H., Pinar, M.C.: The robust network loading problem under polyhedral demand uncertainty: Formulation, polyhedral analysis, and computations. INFORMS Journal on Computing (2010)
11. Mattia, S.: The robust network loading problem with dynamic routing. Universita di Roma la Sapienza, Tech. Rep (2010)
12. Bertsimas, D., Sim, M.: The price of robustness. Operations Research, pp. 35–53 (2004)
13. Koster, A., Kutschka, M., Raack, C.: On the robustness of optimal network design. In: IEEE International Conference on Communications (ICC 2011), Kyoto, Japan (June 2011)
14. Poss, M., Raack, C.: Affine recourse for the robust network design problem: between static and dynamic routing. Zuse Institute Berlin, Tech. Rep (2011)
15. Belotti, P., Capone, A., Carello, G., Malucelli, F.: Multi-layer mpls network design: The impact of statistical multiplexing. Computer Networks 52(6), 1291–1307 (2008)
16. Altin, A., Amaldi, E., Belotti, P., Pinar, M.C.: Provisioning virtual private networks under traffic uncertainty. Networks 49(1), 100–155 (2007)
17. Charnes, A., Cooper, W.: Chance-constrained programming. Management Science, 73–79 (1959)
18. Dfn network, `http://sndlib.zib.de`